Index

Chapter 1

Introduction

1.1 Abstract:

In a world increasingly driven by digital experiences, our web music player project emerges as a dynamic and immersive platform aimed at revolutionizing the way users engage with their music. This project transcends traditional music playback, offering a sophisticated ecosystem for music enthusiasts. Through a seamless blend of cutting-edge technologies and user-centric design, our web music player promises to redefine the digital music landscape. This web music player caters to a diverse audience, providing a comprehensive set of features that extend beyond mere music playback. Users can not only explore and enjoy their favourite tunes but also actively participate in creating and sharing playlists. Social integration fosters a sense of community, allowing users to connect over shared musical interests. The platform's intelligent algorithms provide personalized recommendations, enhancing the discovery of new and exciting music

1.2. Scope of system:

Enjoy a seamless music playback experience for diverse musical tastes. Create and share playlists with friends for a communal music discovery. Receive tailored song suggestions based on individual preferences. Share favourite tracks and playlists on social media for a shared musical experience. Navigate effortlessly with an easy-to-use interface accessible to everyone. Use the platform on different devices and operating systems for flexibility. Connect with like-minded music enthusiasts, building a vibrant community. Actively contribute to playlist creation and music curation. Explore, organize, and enjoy music with intuitive and user-friendly features. Anticipate continuous improvements and integration of new features. Access the web music player on various devices, ensuring inclusivity. Expect ongoing updates, making it a dynamic hub for digital music engagement.

1.3. Operating Environment

- Hardware Requirement: □ Client Side
    - Desktop/Laptop
    - Mobile
    - Tablet
    - Internet connection
    - Ram: 2GB and more

 Server Side

- Web Server
- Ram: 4GB and more

○ Software Requirement: Client Side

- Operating System: Windows, Android, IOS, Linux

1.4. Brief Description of Technology Used Operating System used:

HTML:

HTML or Hyper Text Mark-up Language is the standard mark-up language used to create web pages.HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like<html>). HTML tags most commonly come in pairs like <h1>and </h1>, although some tags represent empty elements and so are unpaired, for example <img>.

The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags).

The purpose of a web browser is to read HTML documents and compose them into visible or audible web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page. HTML describes the structure of a website semantically along with cues for presentation, making it a mark-up language rather than a programming language.

CSS:

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the open web and is standardized across Web browsers according to W3C specifications. Previously, development of various parts of CSS specification was done synchronously, which allowed versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, and CSS3.

When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large websites, where fonts and color information were added to every single page, became a long and expensive process.
CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.

CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document. Most commonly, CSS is combined with the markup languages HTML or XHTML.

Cascading Style Sheet (CSS) is used to set the style in web pages that contain HTML elements. It sets the background color, font-size, font-family, and color … etc. property of elements on a web page. There are three types of CSS, which are given below Inline CSS. Internal or Embedded CSS

Inline CSS:

Inline CSS contains the CSS property in the body section attached with element is known as inline CSS. This kind of style is specified within an HTML tag using the style attribute.

Internal or Embedded CSS:

This can be used when a single HTML document must be styled uniquely. The CSS rule set should be within the HTML file in the head section i.e the CSS is embedded within the HTML file.

External CSS:

Contains separate CSS file which contains only style property with the help of tag attributes (For example class, id, heading … etc.). CSS property written in a separate file with .css extension and should be linked to the HTML document using link tag. This means that for each element, style can be set only once and that will be applied across web pages.

JAVASCRIPT:

The web music player harnesses the power of JavaScript to deliver a dynamic and interactive user experience. As a versatile and client-side scripting language, JavaScript is pivotal in shaping the real-time functionality of the platform directly within users' browsers. Its agility allows for responsive updates, creating a seamless environment for music playback, playlist management, and social interactions.

JavaScript empowers the frontend development, working synergistically with HTML5 and CSS3 to craft an intuitive and visually appealing interface. Its event-driven nature facilitates interactive features, ensuring users can navigate, discover, and engage with their music effortlessly.

In the realm of backend development, Node.js utilizes JavaScript to execute server-side operations efficiently. This choice enhances the platform's scalability and performance, contributing to a robust backend infrastructure that seamlessly supports the demands of a diverse user base.

In essence, JavaScript is not just a programming language for the web music player; it is the driving force behind the platform's responsiveness, interactivity, and cross-browser compatibility, making it an indispensable component in delivering an enriched digital music experience.

FIREBASE:

The web music player integrates Firebase as a fundamental component for robust and real-time database management. Firebase, a cloud-based platform, brings a suite of tools to enhance the performance, scalability, and responsiveness of the system.

Key Features of Firebase Integration:

- **Real-time Database**: Firebase Realtime Database is employed to store and synchronize music-related data instantaneously, ensuring that users experience up-to-the-moment changes in playlists, preferences, and interactions.
- **Authentication Services**: Firebase Authentication enhances user security by providing seamless and secure login options, ensuring a protected and personalized music exploration experience.
- **Cloud Functions**: Leveraging Firebase Cloud Functions allows for the execution of server-side logic, enabling dynamic and efficient handling of various backend operations.

- **Hosting Services:** Firebase Hosting ensures the deployment of the web music player, providing a reliable and scalable infrastructure for hosting the frontend and backend components.

The integration of Firebase not only optimizes data storage and retrieval but also enhances the overall reliability and responsiveness of the web music player. This cloud-based solution contributes to the platform's adaptability, enabling users to engage with their music seamlessly across different devices and locations.

GSAP (GreenSock Animation Platform):

GSAP is a robust JavaScript animation library renowned for its simplicity and exceptional performance. With a user-friendly syntax, it facilitates the creation of smooth, high-performance animations across various web browsers. GSAP's rich features include timeline control, allowing precise sequencing and timing of animations. Its plugin architecture extends functionality, and it seamlessly animates HTML elements, SVG, and custom JavaScript objects. GSAP is widely used for crafting dynamic user interfaces, from simple transitions to complex timeline-based animations. Responsive design capabilities ensure consistent animations across different devices. The community support is vibrant, contributing to continuous development and sharing of resources. Overall, GSAP stands as a go-to choice for developers seeking versatile, performant, and visually engaging web animations.

NODE.JS:

Node.js is a powerful, open-source, server-side JavaScript runtime that enables the execution of JavaScript code outside the browser. Leveraging the V8 JavaScript engine, Node.js excels in building scalable and high-performance applications. It follows an event-driven, non-blocking I/O model, enhancing efficiency and handling concurrent requests adeptly. Node.js is widely adopted for developing real-time applications, APIs, and server-side components, thanks to its asynchronous capabilities. Its expansive package ecosystem, managed by npm (Node Package Manager), provides a wealth of modules and libraries for diverse development needs. Node.js is cross-platform, allowing developers to write server-side applications that run seamlessly on various operating systems. Its versatility, speed, and thriving community make Node.js a preferred choice for modern web development.

Chapter 2

Proposed System

2.1. Study of Similar System:

In examining existing systems in the digital music domain, our proposed web music player distinguishes itself through innovative features such as collaborative playlist creation, personalized recommendations, and seamless social sharing. Unlike traditional players, our system prioritizes community engagement and actively involves users in the curation and sharing of their musical journeys.

2.2. User of System:

The target users of our web music player include a diverse audience ranging from casual listeners to avid music enthusiasts. These users will engage in various roles, from individual playlist curators to participants in collaborative playlist creation. The system is designed to accommodate different preferences and usage patterns.

Chapter3

Analysis and Design

3.1. System Requirements (Functional and Non-Functional)

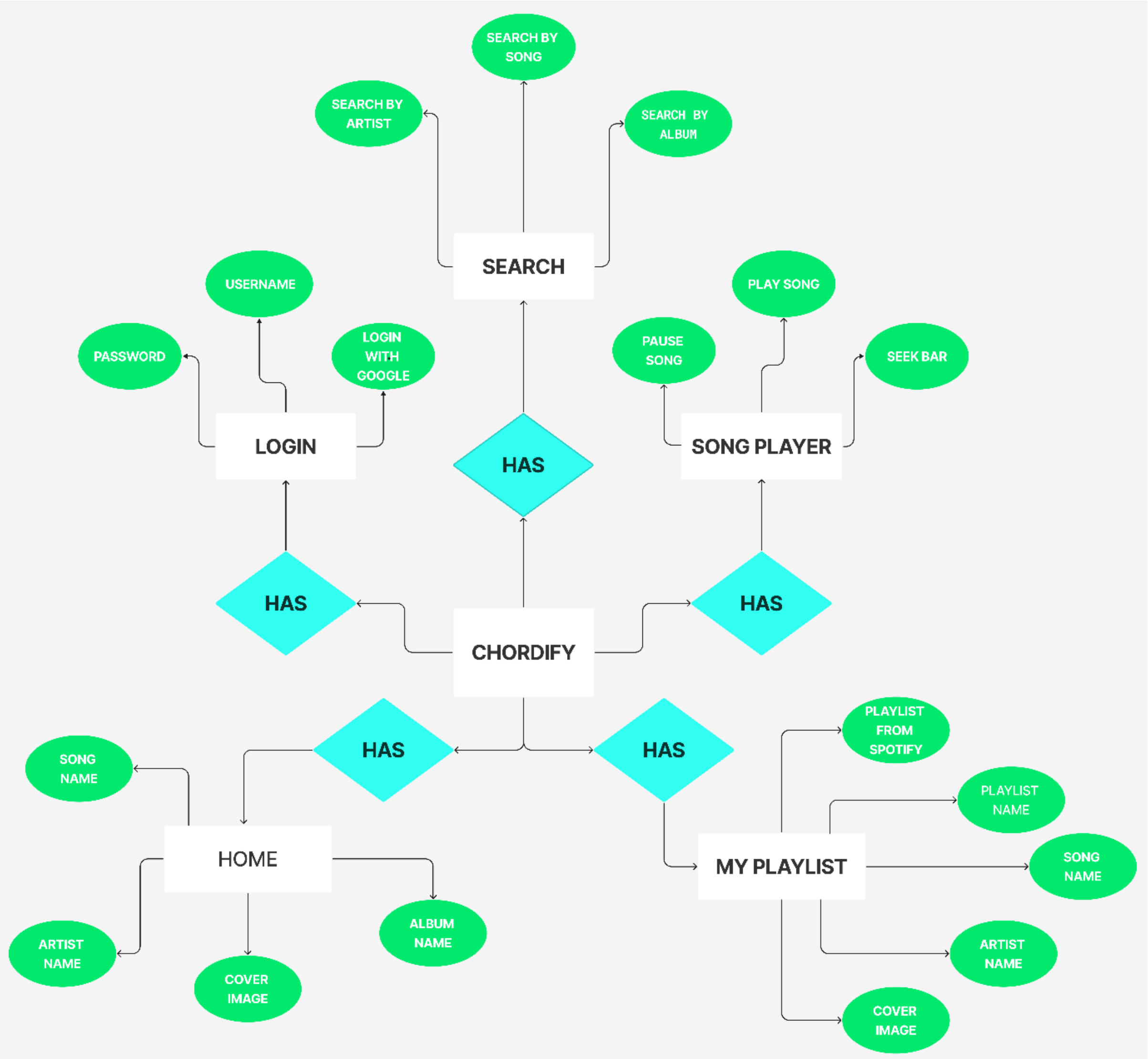
Functional Requirements:

Users must be able to securely register accounts, providing essential details such as username, email, and password. The system should allow users to play music seamlessly, supporting various file formats for an extensive music library. Users should be able to effortlessly share their favourite tracks, playlists, and collaborative creations on various social media platforms. The system should support user interactions such as liking, commenting, and following, promoting community engagement. A robust search functionality is essential, allowing users to easily discover new artists, albums, and tracks. Users must have the ability to personalize their profiles, including adding profile pictures, updating bio information, and adjusting privacy settings. The system should adhere to accessibility standards, ensuring a user-friendly experience for individuals with diverse needs, including those with disabilities. An administrative panel is required for system administrators to manage user accounts, monitor platform activity, and address any issues promptly.

Non-Functional Requirements:

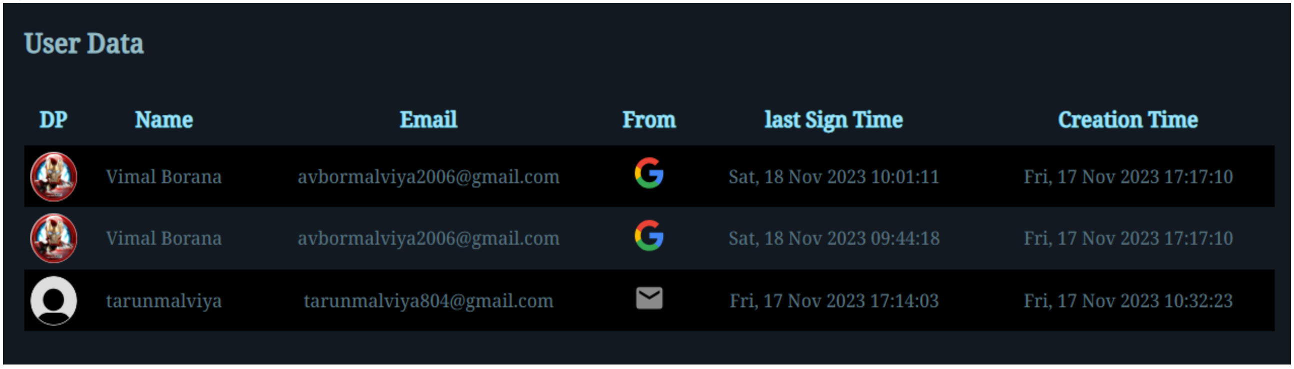
Ensure responsive and quick interactions, minimizing latency in music playback. Accommodate a growing user base with scalability measures for optimal performance. Implement robust encryption and secure authentication to safeguard user data and privacy. Provide an intuitive and user-friendly interface, minimizing the learning curve for users. Ensure consistent availability with minimal risk of service interruptions or downtime. Function seamlessly across different web browsers for a consistent user experience.

3.2.Entity Relationship Diagram (ERD):



3.3 Table Structure:

1) Users:



User Data

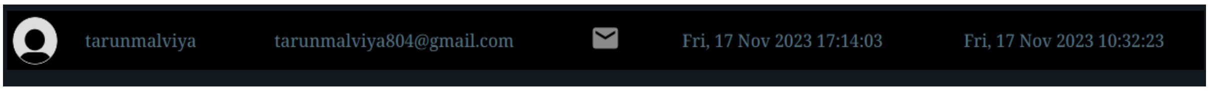
DP	Name	Email	From	last Sign Time	Creation Time
	Vimal Borana	avbormalviya2006@gmail.com	G	Sat, 18 Nov 2023 10:01:11	Fri, 17 Nov 2023 17:17:10
	Vimal Borana	avbormalviya2006@gmail.com	G	Sat, 18 Nov 2023 09:44:18	Fri, 17 Nov 2023 17:17:10
	tarunmalviya	tarunmalviya804@gmail.com	✉	Fri, 17 Nov 2023 17:14:03	Fri, 17 Nov 2023 10:32:23

2) Songs:

Column Name	Data Type	Primary Key	Foreign Key
Song Url	Varchar		
Song Name	char	yes	
Artist	char		
Album	char		
Duration	var		

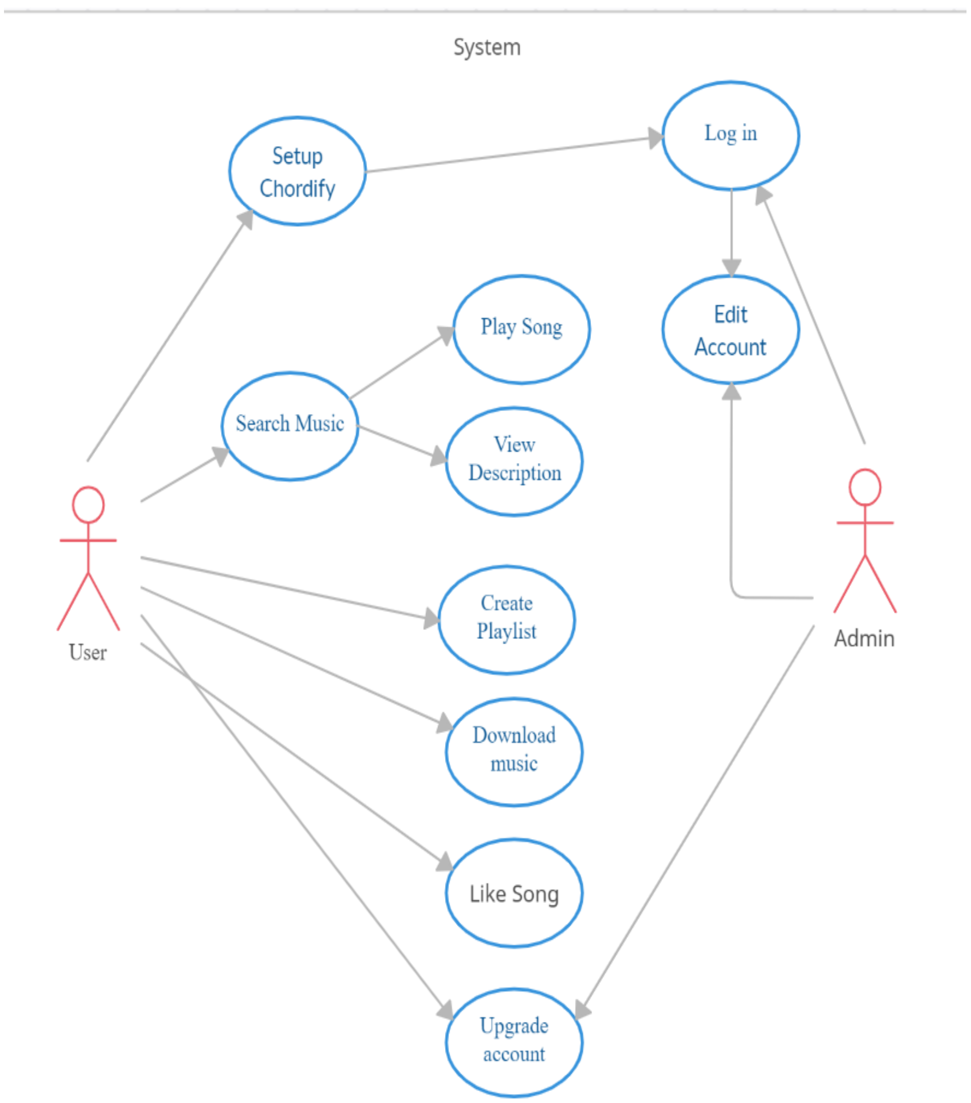
3)Admin:

Column Name	Data type	Primary key	Foreign Key
Email Id	Varchar	yes	
Password	Varchar		

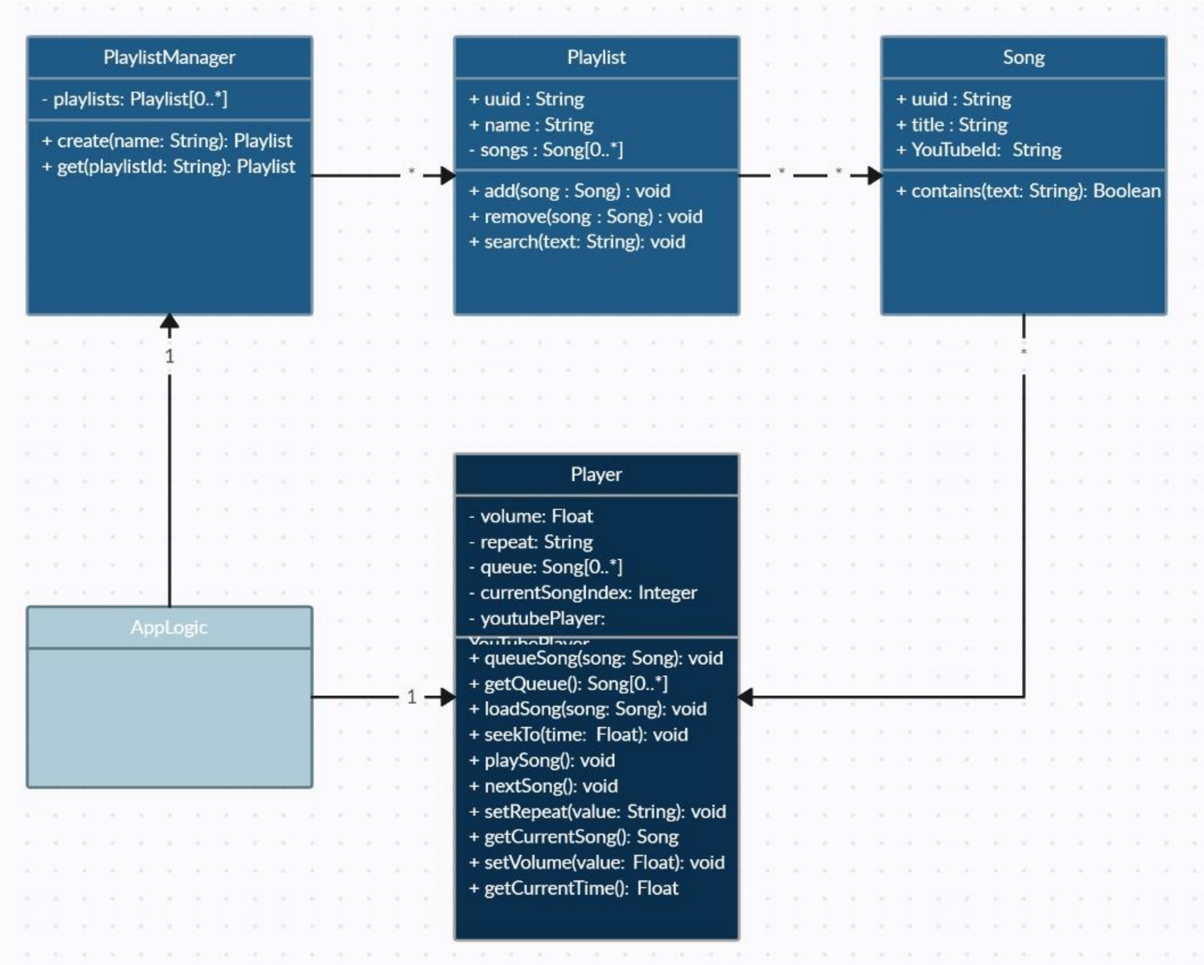


tarunmalviya	tarunmalviya804@gmail.com	✉	Fri, 17 Nov 2023 17:14:03	Fri, 17 Nov 2023 10:32:23

3.4. Use Case Diagram:

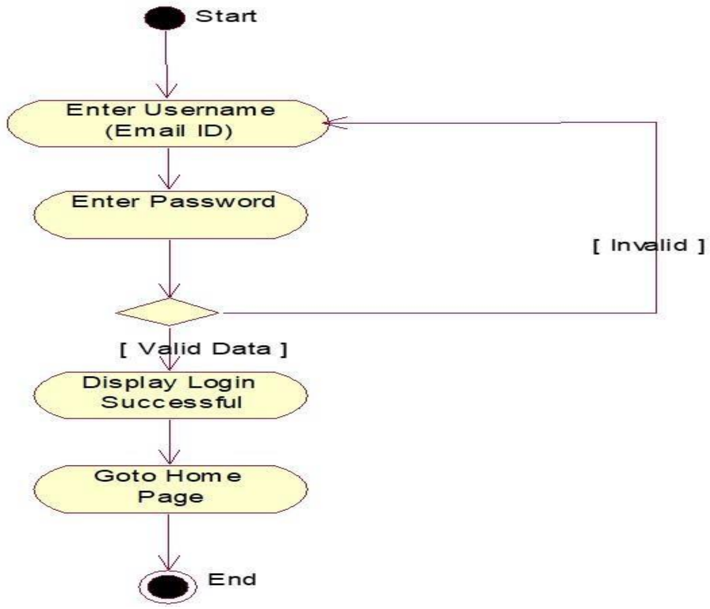


3.5. Class Diagram:

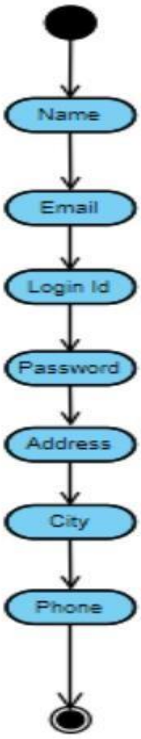

PlaylistManager
- playlists: Playlist[0..*]
+ create(name: String): Playlist
+ get(playlistId: String): Playlist
Playlist
+ uuid : String
+ name : String
- songs : Song[0..*]
+ add(song : Song) : void
+ remove(song : Song) : void
+ search(text: String): void
Song
+ uuid : String
+ title : String
+ YouTubeId: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

3.6. Activity Diagram:

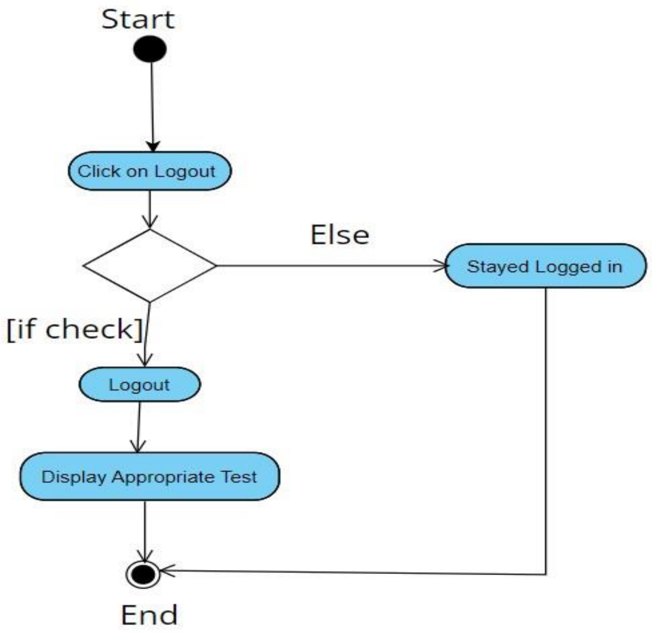
Login:



Registration:

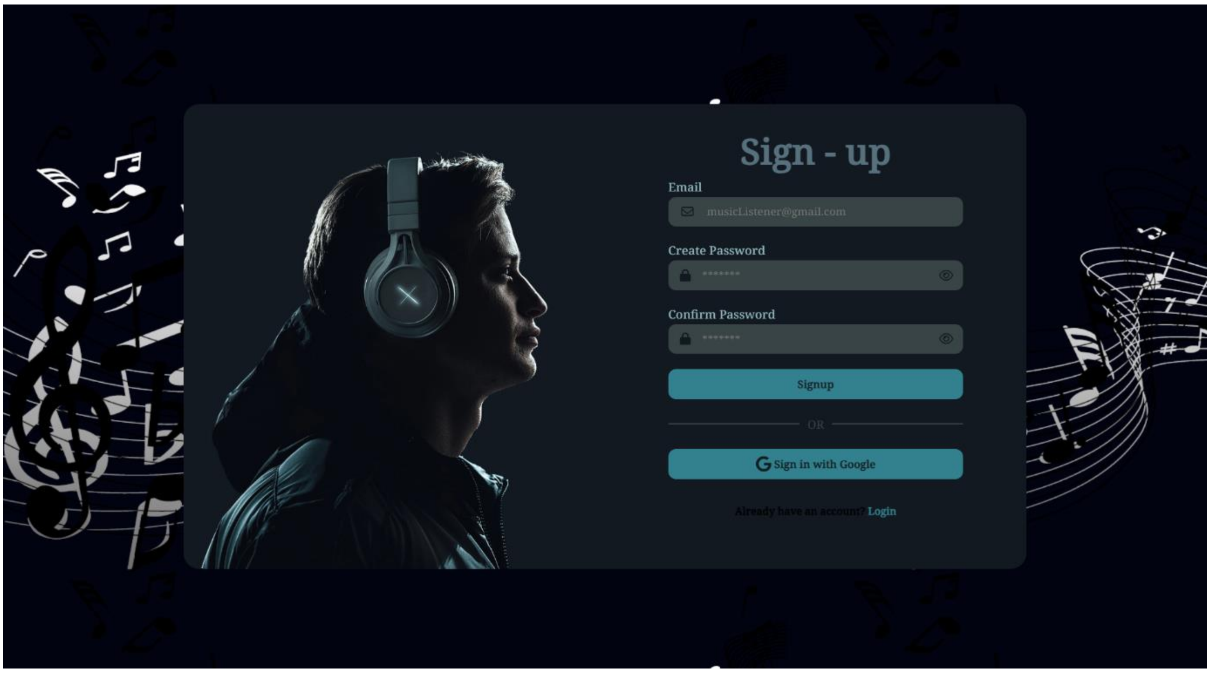


Logout:



3.7. Sample Input Output Screen:

Sign-Up Page:



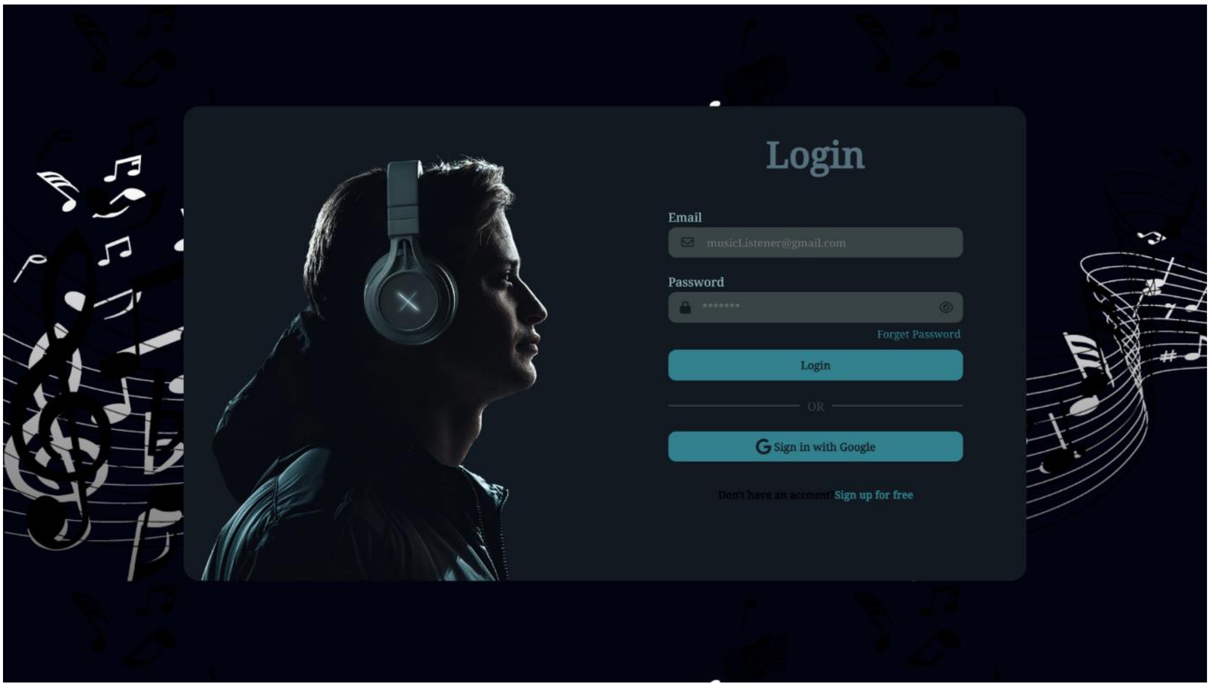
Sign - up

Email
musicListener@gmail.com

Create Password
****

Confirm Password
****

Log-In Page:



Login

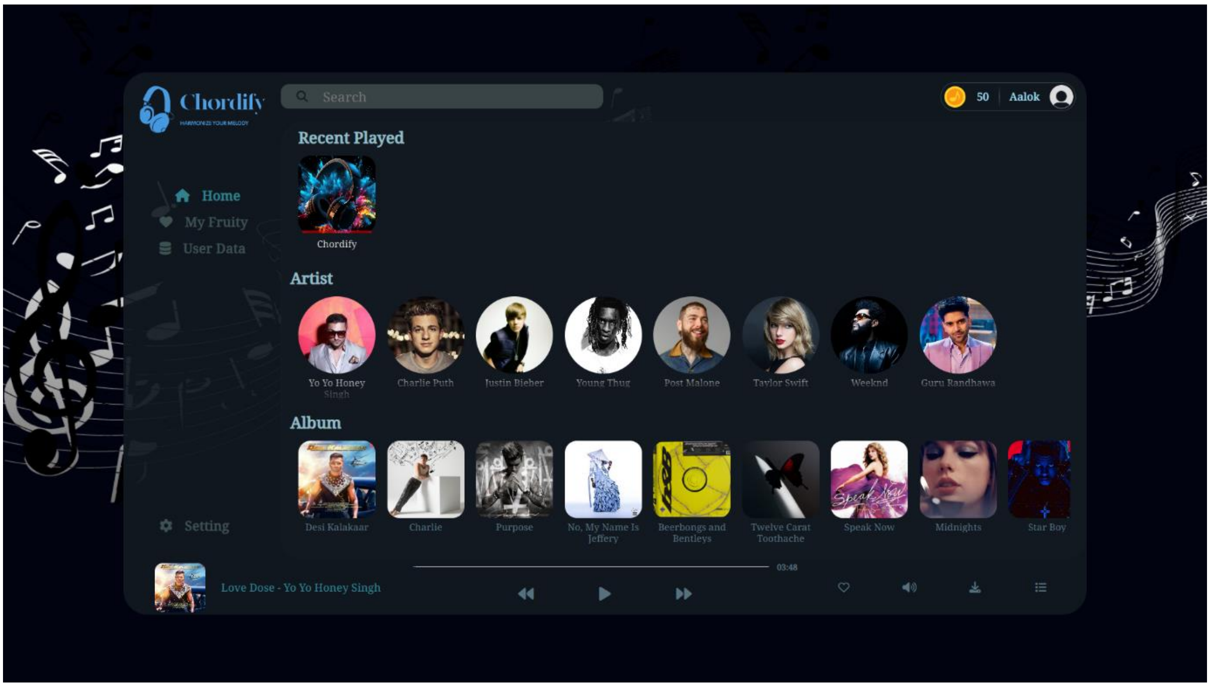
Email

musicListener@gmail.com

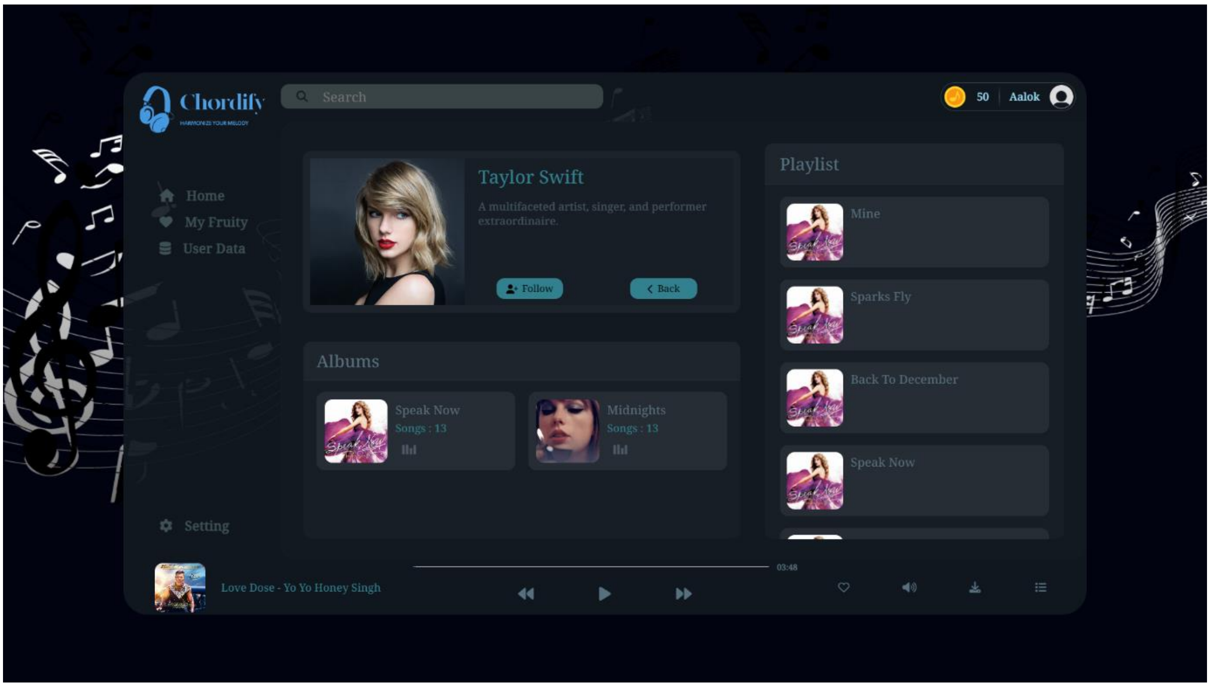
Password

****

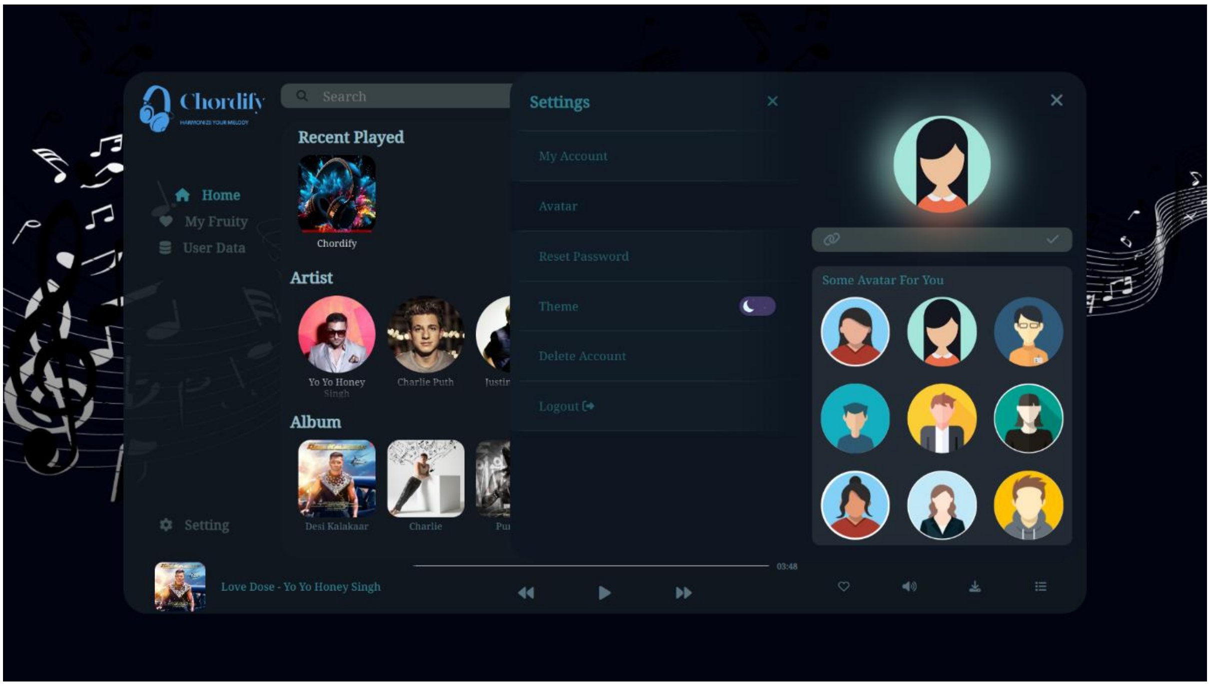
Home Page:



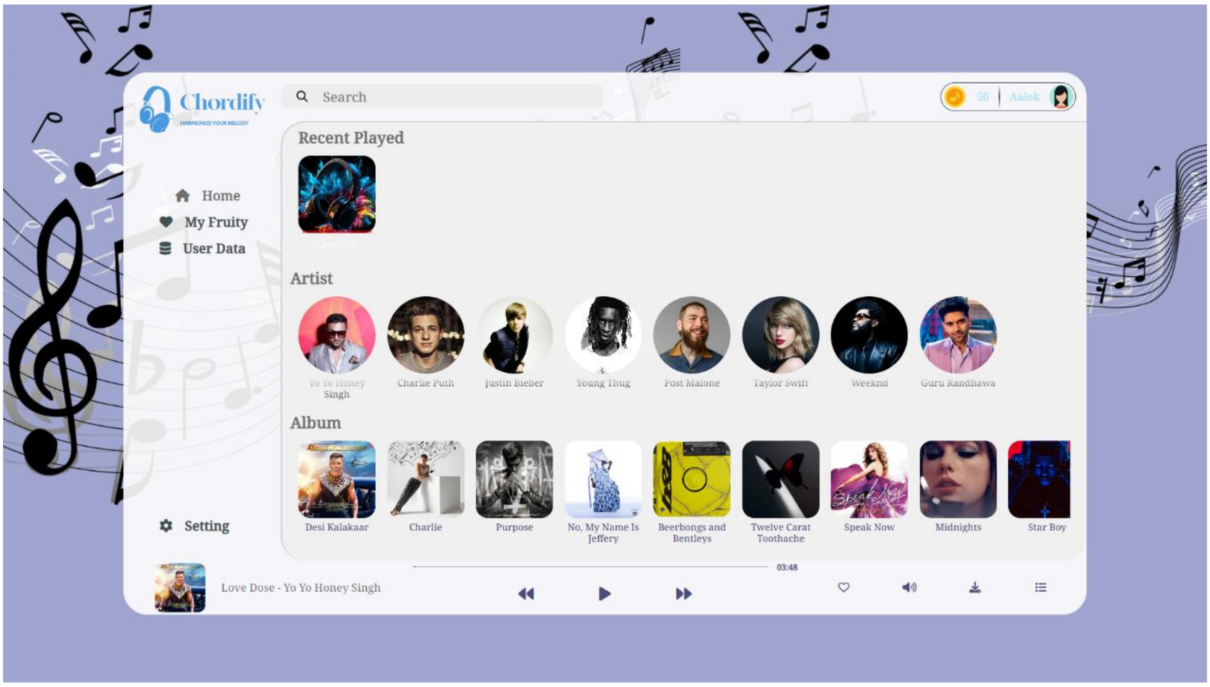
Artist & Playlist Page:



Profile & Setting Page:



Dark To Light Theme:


Chordify
HARMONIZE YOUR MELODY
Search
50
Aalok
Home
My Fruity
User Data
Recent Played
Artist
Yo Yo Honey Singh
Charlie Puth
Justin Bieber
Young Thug
Post Malone
Taylor Swift
Weeknd
Guru Randhawa
Album
Desi Kalakaar
Charlie
Purpose
No, My Name Is Jeffery
Beerbongs and Bentleys
Twelve Carat Toothache
Speak Now
Midnights
Star Boy
Setting
Love Dose - Yo Yo Honey Singh
03:48

Chapter 4

Coding

4.1. Code Snippets:

Index.html: -

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- link css and js file from other folder -->
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css"/>
    <!-- <link rel="stylesheet" href="styles.css"> -->
    <link rel="stylesheet" type="text/css" href="./css/styles.css">
    <link rel="stylesheet" type="text/css" href="./css/srcStyles.css">
    <script src="https://kit.fontawesome.com/a820812e42.js" crossorigin="anonymous"></script>
    <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=Noto+Serif:wght@500&display=swap"
rel="stylesheet">
    <script src="https://www.gstatic.com/firebasejs/ui/6.0.1/firebase-ui-auth.js"></script>
    <link type="text/css" rel="stylesheet"
href="https://www.gstatic.com/firebasejs/ui/6.0.1/firebase-ui-auth.css" />
    <title>Chordify</title>
    <link rel="icon" type="image/x-icon" href="./Image/favicon2.png">
</head>

<body>
    <!-- ========================= ==> Music wrapper <== ========================= -->
    <!-- Alert -->
    <div class="alertContainer flexDirection" style="position: absolute; top: 0; right: 0;
width: 17%; height: auto; display: flex; align-items: center; padding: 1%; z-index:
5;"></div>
    <div class="mainMusicContainer displayFlex">
        <div class="musicContainer">
            <div id="avatarMainSection" class="flexDirection">
                <div id="avatarXmark">
                    <i id="backBtn" class="fa-solid fa-xmark fa-xl"></i>
                </div>
                <div class="uploadedAvatar">
                    <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/2/2c/Default_pfp.svg/340px-
Default_pfp.svg.png" id="avatarImage">
                    <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/2/2c/Default_pfp.svg/340px-
Default_pfp.svg.png" id="bgAvatarImage">
                </div>
                <!-- <label class="displayFlex" for="uploadImg">Upload Avatar</label>
                <input type="file" accept="image/jpeg, image/png, image/jpg" id="uploadImg">
-->
                <div id="inputUrlWrapper">
                    <div id="avatarImageInput" class="displayFlex">
                        <i class="fa-solid fa-link fa-lg"></i>
                    </div>
                    <input type="url" accept="image/png, image/jpeg, image/jpg"
name="uploadImg" id="uploadImg">
                    <div id="saveBtn" class="displayFlex">
                        <i class="fa-solid fa-check fa-lg"></i>
                    </div>
                </div>
                <div class="defaultAvatarSection">
                    <div class="avatarHeading">Some Avatar For You</div>
                    <div id="avatarWrapper"></div>
                </div>
            </div>
            <div id="settingSection">
                <div id="settingHeading" class="settingInnerDiv">
                    <h3>Settings</h3>
                    <i class="fa-solid fa-xmark fa-lg"></i>
                </div>
                <hr>
                <div id="myAccountSetting" class="settingInnerDiv">
                    <div>
                        <div>My Account</div>
                    </div>
                    <div id="myAccountSettingInnerFoldDiv" class="SettingInnerFoldDiv">
                        <table style="width:max-content; text-wrap: nowrap;">
                            <tr>
                              <td>Name</td>
                              <td>&nbsp:&nbsp</td>
                              <td id="myAccountUserName">Maria Anders</td>
                            </tr>
                            <tr>
                              <td>Email</td>
                              <td>&nbsp:&nbsp</td>
                              <td id="myAccountEmail">Aalok@gmail.com</td>
                            </tr>
                            <tr>
                                <td>Password</td>
                                <td>&nbsp:&nbsp</td>
                                <td id="myAccountPassword"><i class="fa-solid fa-key"></i>
encrypted</td>
                              </tr>
                          </table>
                    </div>
                </div>
                <hr>
                <div id="avatarSetting" class="settingInnerDiv">
                    <div>
                        <div>Avatar</div>
                    </div>
                </div>
                <hr>
                <div id="resetPasswordSetting" class="settingInnerDiv">
                    <div>
                        <div>Reset Password</div>
                    </div>
                    <div id="resetPasswordSettingInnerFoldDiv" class="SettingInnerFoldDiv">

                        <!-- Password Input Section -->

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

                        <div class="resetPasswordInputWrapper">
                            <i class="fa-solid fa-lock fa-ml"></i>
                            <input type="password" id="resetPasswordOldPassword"
name="password" max="12" min="6" placeholder="***" maxlength="12">
                            <i class="fa-regular fa-eye fa-ml"></i>
                        </div>

                        <!-- Conform Password Input Section  -->

                        <label for="confirmPassword">New Password</label>

                        <div class="resetPasswordInputWrapper">
                            <i class="fa-solid fa-lock fa-ml"></i>
                            <input type="password" id="resetPasswordNewPassword"
name="password" max="12" min="6" placeholder="***" maxlength="12">
                            <i class="fa-regular fa-eye fa-ml"></i>
                        </div>

                        <label for="confirmPassword">Confirm Password</label>

                        <div class="resetPasswordInputWrapper">
                            <i class="fa-solid fa-lock fa-ml"></i>
                            <input type="password" id="resetPasswordConfirmPassword"
name="password" max="12" min="6" placeholder="***" maxlength="12">
                            <i class="fa-regular fa-eye fa-ml"></i>
                        </div>

                        <div id="resetPasswordChangeBtn" class="displayFlex">Change
Password</div>
                    </div>
                </div>

                <hr>

                <div id="themeSetting" class="settingInnerDiv">
                    <div>
                        <div>Theme</div>
                    </div>

                    <div class="tdnn">
                        <div class="moon"></div>
                   </div>
                </div>

                <hr>

                <div id="deleteAccountSetting" class="settingInnerDiv">
                    <div>
```

```
                        <div>Delete Account</div>
                    </div>

                    <div id="deleteAccountSettingInnerFoldDiv" class="SettingInnerFoldDiv">
                        <div>Deleting your account will permanently erase all your fruity
data. This action cannot be undone.</div>
                        <div class="displayFlex">
                            <div>Confirm</div>
                            <div>Cancel</div>
                        </div>
                    </div>
                </div>

                <hr>

                <div id="logoutAccountSetting" class="settingInnerDiv">
                    <div>
                        <div>
                            Logout  <i class="fa-solid fa-right-from-bracket"></i>
                        </div>
                    </div>
                </div>

                <hr>
            </div>


            <div class="loadingSection">
                <img src="./Image/Logo (2).png" alt="" style="width: 30vw; opacity: 0;"
id="loadingImg">
            </div>

            <div class="subMusicContainer">
                <div class="sideBar">
                    <div class="logoSection">
                        <img src="./Image/Logo (2).png" class="logoImg">
                    </div>
                    <div class="playerSection">
                        <div class="sideNavbar">
                            <nav>
                                <ul style="color: rgb(51, 129, 142); translate: none; rotate:
none; scale: none; transform: translate(20px, 0px);">
                                    <div class="sideNavbarIcon">
                                        <i class="fa-solid fa-house"></i>
                                    </div>
                                    <li>Home</li>
                                </ul>

                                <ul>
                                    <div class="sideNavbarIcon">
                                        <i class="fa-solid fa-heart"></i>
                                    </div>
                                    <li>My Fruity</li>
                                </ul>
                                <ul style="display: none;">
                                    <div class="sideNavbarIcon">
                                        <i class="fa-solid fa-database"></i>
                                    </div>
                                    <li>User Data</li>
                                </ul>
                            </nav>
                        </div>
                        <div class="sideNavbarSetting">
                            <nav>
                                <ul id="settingBtn" style="margin-top: 180%;">
                                    <div class="sideNavbarIcon">
                                        <i class="fa-solid fa-gear"></i>
                                    </div>
                                    <li>Setting</li>
                                </ul>
                            </nav>
                        </div>
                    </div>
                </div>
                <div class="mainSection">
                    <div class="navSection">
                        <div class="searchSection displayFlex">
                            <div class="searchIcon">
                                <i class="fa-solid fa-magnifying-glass fa-sm"></i>
                            </div>
                            <input type="text" class="searchBox" placeholder="Search">
                        </div>

                        <div class="avatarSection">
                            <div class="avatar displayFlex">

                                <svg height="100%" style="aspect-ratio: 1/1;">
                                    <circle cx="50%" cy="50%" r="12" stroke="#ffd424" stroke-
width="3" fill="#f89512" />
                                    <!-- <svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" fill="rgb(255, 212, 36)" version="1.1" id="Capa_1"
height="70%" viewBox="-46.89 -173.89 562.64 562.64" xml:space="preserve">

                                        <g id="SVGRepo_bgCarrier" stroke-width="0"/>

                                        <g id="SVGRepo_tracerCarrier" stroke-linecap="round"
stroke-linejoin="round"/>

                                        <g id="SVGRepo_iconCarrier"> <g> <path
d="M235.554,0C109.784,0,3.532,102.367,3.532,223.536c0,68.314,33.927,149.905,86.43,207.865l1.4
84,1.639l2.164-0.429 c0.553-0.1,12.578-2.637,17.603-
11.834c12.317,10.095,23.253,11.085,34.757,6.961c5.115,9.843,7.103,20.559-6.606,24.445 c-
12.906,3.675-30.152-1.094-42.886-3.025c-21.219-3.222-66.178-4.156-87.61-3.358c-10.092,0.381-
10.128,16.082,0,15.701 c25.874-
0.958,74.494,1.606,100.092,5.241c12.808,1.82,29.16,4.341,41.437-1.623c21.209-10.303,17.637-
27.875,9.105-44.146 c6.404-3.907,13.235-8.844,20.638-14.15c-20.528-9.017-47.406-49.355-
65.764-84.528c-12.99-24.902-38.401-77.676-35.536-104.35 c-15.867,10.896-28.212,20.386-
```

```
34.91,32.837c-1.854-11.745-2.852-23.271-2.852-34.43c0-96.018,100.557-160.516,194.469-160.516
c116.331,0,194.46,83,194.46,160.516c0,11.093-0.897,22.456-2.625,33.953c-6.744-12.219-18.987-
21.612-34.652-32.36 c2.869,26.674-22.537,79.447-35.533,104.35c-18.351,35.173-45.236,75.52-
65.762,84.528c28.204,20.238,47.103,34.46,72.189,11.085
c5.133,7.975,14.463,12.158,14.939,12.359l2.601,1.117l1.884-2.115c51.359-57.743,84.528-
138.493,84.528-205.742 C467.575,102.357,361.322,0,235.554,0z"/> <path d="M93.646,209.917c-
0.99,0-1.863,0.232-2.673,0.589c-0.287,0.126-0.601,0.218-0.87,0.391 c-
11.407,7.468,3.939,55.427,31.244,107.756c24.917,47.752,53.217,82.208,66.652,82.208c0.25,0,0.4
75-0.072,0.713-0.092 c0.988-0.108,1.923-0.353,2.725-0.878c0.878-0.581,1.569-1.479,2.166-
2.528c0.03-0.061,0.062-0.145,0.092-0.204 c6.885-12.788-5.973-57.491-31.038-
105.532C137.771,243.93,107.46,209.909,93.646,209.917z"/> <path d="M350.25,318.652c27.303-
52.329,42.647-100.296,31.242-107.756c-0.269-0.172-0.585-0.265-0.874-0.391 c-0.806-0.356-
1.679-0.589-2.669-0.589c-13.806,0-44.126,34.013-69.012,81.709c-25.062,48.049-37.926,92.756-
31.041,105.536
c0.031,0.063,0.067,0.14,0.1,0.2c0.589,1.058,1.286,1.964,2.16,2.528c0.821,0.537,1.771,0.798,2.
785,0.89
c0.213,0.024,0.421,0.093,0.657,0.093C297.023,400.873,325.328,366.404,350.25,318.652z"/> <path
d="M247.696,93.629c-8.111-1.381-14.741,4.59-14.741,12.818v84.997c-4.795-1.759-9.959-2.771-
15.371-2.771 c-24.697,0-44.715,20.019-
44.715,44.721c0,24.693,20.018,44.711,44.715,44.711c18.098,0,33.662-10.748,40.691-26.205
c3.414-7.481,3.903-20.99,3.936-29.212c0.092-26.299,0.092-78.205,0.092-
78.205c57.771,0,54.309,34.622,47.545,54.505 c-2.646,7.789-
1.07,9.091,4.496,3.032C384.842,125.226,286.681,100.264,247.696,93.629z"/> </g> </g>

                                    </svg> -->
                                    <svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" fill="rgb(255, 212, 36)" version="1.1" id="Capa_1"
height="100%" viewBox="0 -194 335 748" xml:space="preserve">
                                        <g>
                                            <path d="M97.173,322.156c35.754,0.874,67.271-
11.577,84.481-30.805c6.111-6.845,10.074-14.932,10.836-16.527   c0.448-0.949,0.825-
1.955,1.149-2.997l45.168-148.824c2.678-8.782,9.991-10.542,15.978-3.577
c4.629,5.392,9.606,11.507,14.659,18.304c20.823,27.968,22.502,64.76,11.397,94.439c-
11.112,29.667-32.111,38.046-25.375,47.436   c6.757,9.418,33.226-13.974,50.453-41.793c17.212-
27.824,19.136-74.354,3.603-112.445c-15.54-38.099-38.17-62.592-42.486-82.467   c-0.269-1.272-
0.545-2.523-0.821-3.737c-0.453-2.06-0.269-5.574,0.429-7.837l1.242-4.105c3.391-11.146-2.89-
22.922-14.058-26.307   c-11.141-3.384-22.915,2.914-26.297,14.052L172.77,195.409c-2.673,8.784-
10.884,11.481-19.142,7.494   c-15.156-7.325-33.448-11.817-53.236-12.303c-53.387-1.311-
97.377,27.086-98.267,63.426   C1.235,290.35,43.784,320.862,97.173,322.156z"/>
                                        </g>
                                    </svg>
                                </svg>



                                <div id="coinCount" class="displayFlex">
                                    <div>0</div>
                                </div>

                                <hr>

                                <div>Aalok</div>
                                <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/2/2c/Default_pfp.svg/340px-
Default_pfp.svg.png" alt="Avatar" id="avatarImg">
                            </div>
                        </div>
                    </div>
                    <div class="mainMusicSection">
                        <div class="recentPlayedSection" id="recentSection">
                            <div id="recentPlayedHeading">Recent Played</div>
                            <div class="recentPlayedBoxForFlex displayFlex">
                                <div class="recentPlayedMainBox">
                                    <img
src="https://static.wixstatic.com/media/483423_0d77597e08a84557aaac57d0e9daa99c~mv2.gif"
alt="" id="beatGif">
                                </div>
                            </div>
                        </div>

                        <div class="artistSection" id="artistSection">
                            <div class="artistHeading">Artist</div>
                            <div class="artistSubSection displayFlex">
                                <div class="artistWrapper">
                                    <!-- <div class="artistDataWrapper">
                                        <img src="https://shorturl.at/TV249" alt=""
class="artistImg">
                                        <div class="artistName">Yo Yo Honey Singh</div>
                                    </div> -->
                                </div>
                            </div>
                        </div>

                        <div class="albumSectionWrapper" id="albumSection">
                            <div class="albumHeadingDiv">Album</div>
                            <div class="albumSubSectionDiv displayFlex">
                                <div class="albumWrapperSection">
                                    <!-- <div class="albumDataWrapper">
                                        <img src="https://scontent.fnag1-
4.fna.fbcdn.net/v/t39.30808-1/308837930_427756579461593_1434386063803892016_n.png?stp=dst-
png_p120x120&_nc_cat=103&ccb=1-7&_nc_sid=5fac6f&_nc_ohc=GRuk-UB-
ItsAX9YZP6c&_nc_ht=scontent.fnag1-
4.fna&oh=00_AfCtgxFXsWgCKh7pyZOkzWPyHisSErw0SixD3rKg1Cx9tg&oe=652B44BC" alt=""
class="albumImg">
                                        <div class="albumName">Desi Kalakaar</div>
                                    </div> -->
                                </div>
                            </div>
                        </div>

                        <div class="randomSong displayFlex">
                            <div class="randomSongDiv">
                                <div class="randomSongHeading displayFlex">SONGS</div>
                                <div class="randomSongWrapper">
                                    <!-- <div class="randomSongBox displayFlex">
                                        <div class="randomSongSubMiniBox">
```

```
                                            <div class="randomSongImg displayFlex">
                                                <img src="https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcTp9REClTzT7YmgnXrXgrfgDw3xLV673aPq4Q&usqp=CAU" alt="">
                                            </div>
                                            <div class="randomSongName">Desi Kalakaar</div>
                                        </div>
                                    </div> -->

                                </div>
                            </div>
                        </div>

                    </div>

                    <div class="playListDataSection">
                        <div class="imageData">
                            <div class="dataImgWrapper displayFlex">
                                <img src="https://play-
lh.googleusercontent.com/xVOHM_0sCJplpk9qJ_KRygXYiE3JAiieCv-p_70d9KdoqDmjqRHJAWUF24csoja-
hL8=w240-h480-rw" alt="">
                                <div class="nameData">
                                    <div class="nameAboutWrapper">
                                        <div class="artistNameDiv">Yo Yo Honey Singh</div>
                                        <div class="artistAboutDiv">he is the great artist of
india Lorem ipsum dolor sit amet, consectetur adipisicing elit. Ullam, aut!</div>
                                    </div>
                                    <div class="backFollowBtnWrapper">
                                        <div class="followBtn displayFlex"><i class="fa-solid
fa-user-plus"></i><div>Follow</div></div>
                                        <div class="backBtn displayFlex"><i class="fa-solid
fa-chevron-left"></i><div>Back</div></div>
                                    </div>
                                </div>
                            </div>
                        </div>
                        <div class="playlistData">
                            <div class="playListSongSection">
                                <div class="playlistHeadingWrapper">
                                    <div>Playlist</div>
                                </div>
                                <img src="./Image/ezgif.com-crop.gif" style="height:
30%;margin: auto;opacity: 0;" class="playlistBeatGif">
                                <div class="playlistSongWrapper">
                                    <!-- <div class="playlistSongData">
                                        <div class="playlistSongImageDiv displayFlex">
                                            <img src="https://akm-img-a-
in.tosshub.com/businesstoday/images/story/201906/bmw_660x450_062519044727.jpg?size=948:533"
alt="" class="playlistSongImage">
                                        </div>
                                        <div class="playlistSongNameDiv">
                                            <div class="playlistSongName">Lorem ipsum dolor
sit amet.</div>
                                        </div>
                                    </div> -->
                                </div>
                            </div>
                        </div>
                        <div class="albumData">
                            <div class="albumSection">
                                <div class="albumHeading">
                                    <div>Albums</div>
                                </div>
                                <div class="albumWrapper"></div>
                            </div>
                        </div>
                    </div>

                    <div class="searchDataSection">
                        <div class="recentSearchHeading">Recent Search <i class="fa-solid fa-
angle-up"></i></div>

                        <div class="div" style="width: 100%;height: max-content;">
                            <div class="recentSearchesWrapper">
                                <!-- <div class="recentSearchBox">
                                    <div class="recentSearchText">Look Who Has Coming</div>
                                    <div class="recentSearchXicon">
                                        <i class="fa-solid fa-xmark"></i>
                                    </div>
                                </div> -->
                            </div>
                        </div>

                        <div class="recentSearchHeading">Search Results</div>
                        <div class="searchDataWrapper">

                            <!-- <div class="searchDataBox displayFlex">
                                <div class="searchDataSubBox">
                                    <div class="searchDataImg displayFlex">
                                        <img src="./Image/coverImage.jpg" alt="">
                                    </div>
                                    <div class="searchDataName">Desi Kalakaar akjls
jlflsdjfljefgjg erjgejrgne ejfgkejnwg</div>
                                </div>
                            </div> -->

                        </div>
                    </div>

                    <div class="myFruitySection">
                        <div class="myFruityHeading">My Fruity</div>
                        <div class="myFruityWrapper">
                            <!-- <div class="myFruityBox displayFlex">
                                <div class="myFruitySubBox">
                                    <div class="myFruityImg displayFlex">
                                        <img src="./Image/coverImage.jpg" alt="">
                                    </div>
                                    <div class="myFruityName">Desi Kalakaar akjls
jlflsdjfljefgjg erjgejrgne ejfgkejnwg</div>
```

```
                                </div>
                            </div> -->
                        </div>
                    </div>

                    <div id="userDataSection">
                        <div id="userDataSectionHeading">User Data</div>

                        <table>
                            <tr>
                              <th>DP</th>
                              <th>Name</th>
                              <th>Email</th>
                              <th>From</th>
                              <th>last Sign Time</th>
                              <th>Creation Time</th>
                            </tr>
                        </table>
                    </div>

                    <!-- <div class="addMusicSection displayFlex">

                    </div> -->

                </div>
                <div class="musicBox displayFlex">
                    <div class="rangeWrapper displayFlex">
                        <div class="footerBar displayFlex">
                            <div class="songImg displayFlex">
                                <img src="./Image/Untitled.png" alt="Chordify">
                            </div>
                            <div class="songName">
                                <span id="songName">Chordify</span>
                            </div>
                        </div>
                        <div class="btnWrapper displayFlex flexDirection">
                            <div class="songDuration displayFlex">
                                <input type="range" value="0" min="0" max="100"
class="songRangeBar">
                                <p4 id="songDurationNum">3:16</p4>
                            </div>
                            <div class="songPlayPauseBtn">
                                <div class="backwardBtn">
                                    <i class="fa-solid fa-backward fa-lg"></i>
                                </div>
                                <div class="playBtn displayFlex">
                                    <i class="fa-solid fa-play fa-lg"></i>
                                </div>
                                <div class="forwardBtn">
                                    <i class="fa-solid fa-forward fa-lg"></i>
                                </div>
                            </div>
                        </div>
                        <div class="likeBtn">
```

```
                            <div class="iconWrapper">
                                <i class="fa-regular fa-heart fa-sm"></i>
                            </div>
                            <div class="iconWrapper">
                                <div class="rangeWrapperDiv">
                                    <input type="range" min="0" max="100" value="100"
class="volumeRange">
                                </div>
                                <i class="fa-solid fa-volume-high fa-sm"></i>
                                <!-- <i class="fa-solid fa-volume-off fa-xl"></i> -->
                            </div>
                            <div class="iconWrapper">
                                <i class="fa-solid fa-download fa-sm"></i>
                            </div>
                            <div class="iconWrapper">
                                <i class="fa-solid fa-list fa-sm"></i>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <!-- Main Container -->
    <div class="mainContainer">

        <div class="formSectionWrapper displayFlex">

            <!-- Background Image Container -->
            <div class="imgContainer">
                <!-- <img class="coverImg" src="./Image/pexels-nardo-3574678 (1).jpg"> -->
                <img class="coverImg" src="./Image/pexels-nardo-godot.png">
            </div>


            <!-- Container -->
            <div class="container">

    <!-- =================== ==> Flip Container One <== ===================  -->

                <div class="subContainer displayFlex flexDirection">

                    <h1 class="headingText">Login</h1>

                    <!-- Login Wrapper -->
                    <div class="loginContainer displayFlex flexDirection">

                        <!-- Email Input Section -->
                        <label class="width100Per" for="email">Email</label>
                        <div class="input slideanimation1 displayFlex margin10 width100Per
height10Per">
                            <div class="inputContainer">
                                <i class="fa-regular fa-envelope fa-ml"></i>
                            </div>
                            <input class="width100Per height100Per" type="email" id="email"
name="email" autocomplete="off" placeholder="musicListener@gmail.com">
                        </div>

                        <!-- Password Input Section -->
                        <label class="width100Per" for="password">Password</label>

                        <div class="input slideanimation2 displayFlex width100Per
height10Per" style="margin-bottom: 2%;">
                            <div class="inputContainer">
                                <i class="fa-solid fa-lock fa-ml"></i>
                            </div>
                            <input class="width100Per height100Per" type="password"
id="password" name="password" placeholder="***" maxlength="12">
                            <div class="inputEyeContainer" id="passwordEye">
                                <i class="fa-regular fa-eye fa-ml"></i>
                            </div>
                        </div>

                        <!-- Forget Password Button -->
                        <input id="forgetPasswordBtn" type="button" value="Forget Password">

                        <!-- Login Button -->
                        <button id="buttonanimation1" class="margin10 width100Per height10Per
radius10">Login</button>

                        <div class="displayFlex margin10 width100Per">
                            <hr>
                            <div id="orText">OR</div>
                            <hr>
                        </div>

                        <!-- Google Button -->
                        <button id="buttonanimation1" class="loginWithGoogle margin10
width100Per height10Per radius10"><i class="fa-brands fa-google fa-xl"></i>&nbsp;Sign in with
Google</button>

                        <!-- Sign Up Button -->
                        <div class="displayFlex margin10 width100Per height10Per">
                            <h5 class="link">Don't have an account?</h5><h5><a class="link"
id="linkColor1">Sign up for free</a></h5>
                        </div>

                    </div>
                </div>

            <!-- ==================== ==> Flip Container Two <== ===================  -->

                <div class="subContainer2 displayFlex flexDirection">

                    <h1 class="headingText">Sign - up</h1>

                    <!-- Login Wrapper -->
                    <div class="loginContainer displayFlex flexDirection">

                        <!-- Email Input Section -->
                        <label class="width100Per" for="email">Email</label>

                        <div class="input displayFlex margin10 width100Per height10Per">
                            <div class="inputContainer">
                                <i class="fa-regular fa-envelope fa-ml"></i>
                            </div>
                            <input class="width100Per height100Per" type="email"
id="newEmail" name="email" autocomplete="off" placeholder="musicListener@gmail.com">
                        </div>

                        <!-- Password Input Section -->

                        <label class="width100Per" for="password">Create Password</label>

                        <div class="input displayFlex margin10 width100Per height10Per">
                            <div class="inputContainer">
                                <i class="fa-solid fa-lock fa-ml"></i>
                            </div>
                            <input class="width100Per height100Per" type="password"
id="createPassword" name="password" placeholder="***" maxlength="12">
                            <div class="inputEyeContainer" id="createPasswordEye">
                                <i class="fa-regular fa-eye fa-ml"></i>
                            </div>
                        </div>

                        <!-- Conform Password Input Section  -->

                        <label class="width100Per" for="confirmPassword">Confirm
Password</label>

                        <div class="input displayFlex margin10 width100Per height10Per">
                            <div class="inputContainer">
                                <i class="fa-solid fa-lock fa-ml"></i>
                            </div>
                            <input class="width100Per height100Per" type="password"
id="confirmPassword" name="confirmPassword" placeholder="***" maxlength="12">
                            <div class="inputEyeContainer" id="confirmPasswordEye">
                                <i class="fa-regular fa-eye fa-ml"></i>
                            </div>
                        </div>

                        <!-- Login Button -->
                        <button id="loginButton" class="displayFlex margin10 width100Per
height10Per radius10">Signup

                            <svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink" style="margin-left: 2%; background: rgba(255, 255,
255, 0); shape-rendering: auto; aspect-ratio: 1/1;" height="80%" viewBox="0 0 100 100"
preserveAspectRatio="xMidYMid">
                                <circle cx="28" cy="75" r="11" fill="#131E24">
```

```
                                  <animate attributeName="fill-opacity"
repeatCount="indefinite" dur="1s" values="0;1;1" keyTimes="0;0.2;1" begin="0s"/>
                                </circle>

                                <path d="M28 47A28 28 0 0 1 56 75" fill="none"
stroke="#00dbff" stroke-width="10">
                                  <animate attributeName="stroke-opacity"
repeatCount="indefinite" dur="1s" values="0;1;1" keyTimes="0;0.2;1" begin="0.1s"/>
                                </path>
                                <path d="M28 25A50 50 0 0 1 78 75" fill="none"
stroke="#ffffff" stroke-width="10">
                                  <animate attributeName="stroke-opacity"
repeatCount="indefinite" dur="1s" values="0;1;1" keyTimes="0;0.2;1" begin="0.2s"/>
                                </path>
                            </svg>

                        </button>

                        <div class="displayFlex margin10 width100Per">
                            <hr>
                            <div id="orText">OR</div>
                            <hr>
                        </div>

                        <!-- Google Button -->
                        <button id="signWithGoogle" class="margin10 width100Per height10Per
radius10"><i class="fa-brands fa-google fa-xl"></i>&nbsp;Sign in with Google</button>
                        <!-- Sign Up Button -->
                        <div class="displayFlex margin10 width100Per height10Per">
                            <h5 class="link">Already have an account?</h5><h5><a class="link"
id="linkColor2">&nbsp;Login</a></h5>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
    <!-- All javaScript file links here -->
    <!-- <script type="text/javascript" src="https://code.jquery.com/jquery-
3.1.0.min.js"></script> -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.7.0/jquery.min.js"
integrity="sha512-
3gJwYpMe3QewGELv8k/BX9vcqhryRdzRMxVfq6ngyWXwo03GFEzjsUm8Q7RZcHPHksttq7/GFoxjCVUjkjvPdw=="
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/lettering.js/0.6.1/jquery.lettering.min.js"
integrity="sha512-
VJ/iYbiu1eJ6yLimfTi65t2R9TFcG5D9X8ZCfbbEFhTfPnKJh8byoKXEawi5ScJZBYL1eiirL1+MczZDx0Tz9Q=="
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/textile-js/2.1.1/textile.min.js"
integrity="sha512-
tXqnJLtLXWw7dsznazHu0eOB2d/oQQ1b6BMvIvwI5gUG5f3e8MxaNTkCBdeQmpwlemlXwNRas39dDQi0BGd2Bg=="
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
```

```
    <script
src="https://cdn.jsdelivr.net/npm/textillate@0.4.1/jquery.textillate.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.12.1/gsap.min.js"
integrity="sha512-
qF6akR/fsZAB4Co1QDDnUXWnaQseLGXoniuSuSlPQK6+aWhlMZcHzkasCSlnWoe+TJuudlka1/IQ01Dnhgq95g=="
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.12.1/ScrollTrigger.min.js"
integrity="sha512-
IHDCHrefnBT3vOCsvdkMvJF/MCPz/nBauQLzJkupa4Gn4tYg5a6VGyzIrjo6QAUy3We5HFOZUlkUpP0dkgE60A=="
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/gsap/3.11.4/ScrollToPlugin.min.js"></script>
    <script type="module" src="fireBase/firebaseInit.js"></script>
    <script type="module" src="fireBase/firebaseObserver.js"></script>
    <script type="module" src="fireBase/googleMethod.js"></script>
    <script type="module" src="fireBase/loginMethod.js"></script>
    <script type="module" src="fireBase/logoutMethod.js"></script>
    <script type="module" src="fireBase/SignInMethod.js"></script>
    <!-- <script type="module" src="fireBase/googleLogin.js"></script> -->
    <!-- <script type="module" src="fireBase/storage.js"></script> -->
    <script type="module" src="./js/loadSongs.js"></script>
    <script type="module" src="./js/artistAbout.js"></script>
    <script type="module" src="./js/artistPage.js"></script>
    <script type="module" src="./js/albumPage.js"></script>
    <script type="module" src="./js/profilePic.js"></script>
    <script type="module" src="./js/setting.js"></script>
    <!-- <script type="module" src="./js/loginFlip.js"></script> -->
    <!-- <script type="module" src="./js/loading.js"></script> -->
    <script type="module" src="./js/SideNavbar.js"></script>
    <script type="module" src="./js/songSearching.js"></script>
    <script type="module" src="./js/function.js"></script>
</body>
</html>
```

Chapter 5

Testing

5.1 Test Strategy

Test strategy describes the overall approach to testing. The goal of the testing process is to determine all faults in our project. The program was subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not.

Strategies used are:

Functional Testing: This ensures that the functionalities of a web application are properly functioning or not.

Interface Testing: This testing method ensures that the three main components of a web application which are web server, web browser and database are running harmoniously. This testing type checks whether there is any interruption while the data is being transferred. Upon that, the communication taking place between various interfaces is also thoroughly checked.

Compatibility Testing: This testing methodology ensures that a particular web application is compatible with all browsers. Compatibility testing takes place at three levels which are browser compatibility, operating system compatibility and device compatibility.

5.2 Unit Test Plan:

Unit testing is commenced when a unit has been created and effectively reviewed. In order to test a single module, we need to provide a complete environment i.e. besides the section we would require.

I. The procedures belonging to other units that the unit under test calls.

II. Non local data structures that modules accesses.

III. A procedure to call the functions of the unit under test with appropriate parameters.

Types of Unit Testing:

White Box Testing

White box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Test cases can be derived that

I. Guarantee that all independent paths within a module have been exercised at least once.

II. Exercise all logical decisions on their true and false sides,

III. Execute all loops at their boundaries and within their operational bounds, and Exercise internal data structures to ensure their validity.

Black Box Testing

Black box testing also called behavior testing, focuses on the functional requirement of the software. That is black box testing enables the software engineers to derive test of input condition. That will fully exercise all functional requirements for a program. Black box testing is not an alternative to a white box testing technique. Rather it is a complementary approach that is likely to in cover a different class of error's then white box method. Black Box Testing attempts to find errors in the following category.

I. Incorrect or Missing function

II. Interface errors

III. Errors in a data structure or external database accesses

IV. Behavior or performance error's and

V. Initialization and Termination error's

3) Gray-Box Testing

It's referred to as semi-transparent testing. It is a combination of a Black Box and White Box testing. It is the type of testing in which tester aware with internal functionality of a method or unit but not in a deeper level like white box testing. In this, the user partially aware of the internal functionality of a system. Different type of testing covered under a Gray box testing is mentioned as following

I. Matrix testing.

II. Pattern Testing.

III. Orthogonal Pattern testing.

IV. Regression Testing.

Test case for login:

Test Case No	Test field	Test Description	Test Data	Expected Result	Actual Result	Member

TS01	Username, Password	Valid data for login: i)Enter valid username and password ii)Click on submit button	i)Username = “divya” ii)Password= ” *****”	System should display successful login page and display user profile page	Display login successful message and display option page.	Success

TS02		Invalid data for login: i)Enter invalid username and password lick on submit button	i)Username = “saurabh@123” ii)Password= ****”	System should display error message “Invalid Login”	Display error message “Invalid Login”	Success
TS03	Username	Empty Validation: keep username field empty	username= “ ”	System should display error message “Please enter username	Display error message “Please enter username”	PASS
	Password	Empty Validation: keep password field empty	Password=” ”	system should display error message “Please enter password”	Display error message “Please enter password”	PASS

Test case for Sign-Up:

Test Case No	Test field	Test Description	Test Data	Expected Result	Actual Result	Member
TS01	Username, Password	Valid data for login: i)Enter valid username and	i)Username = “saurabh mule” ii)Password= ” ***”	System should display successful login page and display	Display login successful message and display option page.	Success

		password ii)Click on submit button		user profile page		
TS02		Invalid data for login: i)Enter invalid username and password lick on submit button	i)Username = “tejas@123” ii)Password= ****”	System should display error message “Invalid Login”	Display error message “Invalid Login”	Success
TS03	Username	Empty Validation: keep username field empty	username= “ ”	System should display error message “Please enter username	Display error message “Please enter username”	PASS

	Password	Empty Validation: keep password field empty	Password=” ”	system should display error message “Please enter password”	Display error message “Please enter password”	PASS

Chapter 6

Limitation of proposed system

- **Dependency on Internet Connectivity:**
  The web music player relies on a stable internet connection for seamless music streaming. Offline functionality may be limited.

- **Limited Browser Compatibility:**
  While efforts are made for cross-browser compatibility, certain features may perform differently across various browsers, impacting the user experience.

- **Data Privacy Concerns:**
  Despite robust security measures, data privacy concerns may arise, and users should be cautious about sharing sensitive information.

- **Music Library Size:**
  The system's performance may be affected when managing extensive music libraries, impacting playlist creation and search functionalities.

- **Device Compatibility:**
  Certain older devices or those with limited processing power may experience performance issues, affecting the overall user experience.

- **Collaborative Playlist Limitations:**
  Collaborative playlist features may encounter challenges in real-time synchronization, potentially leading to delays in updating shared playlists.

- **Limited AI Personalization:**
  While the system provides personalized recommendations, the depth of artificial intelligence (AI) personalization may be limited compared to specialized music recommendation systems.

- **Platform-Specific Features:**
  Some features, especially those reliant on third-party APIs or specific platform functionalities, may not be uniformly accessible across different devices and operating systems.

- **Continuous Development Needs:**
  Ongoing updates and maintenance are crucial to address emerging limitations and adapt to evolving technologies, requiring continuous development efforts.

- **External API Dependencies:**
  The integration of external APIs for music content may introduce dependencies and potential disruptions if these APIs undergo changes or become unavailable.

Understanding these limitations is essential for users to make informed decisions and for the development team to plan future enhancements and optimizations. Regular user feedback will be valuable in addressing and mitigating these limitations over time.

Chapter7

Proposed Enhancement

1. Future enhancement of the project is to make it more accurate.

2. Wanting to add online batches facility, according to user's convenient time and favourite batch.

3.To make this system more attractive.

4.Highly secure.

5.Generation of final bill for customer.

6.Making our system more responsive. want to increase the communication between customer and admin.

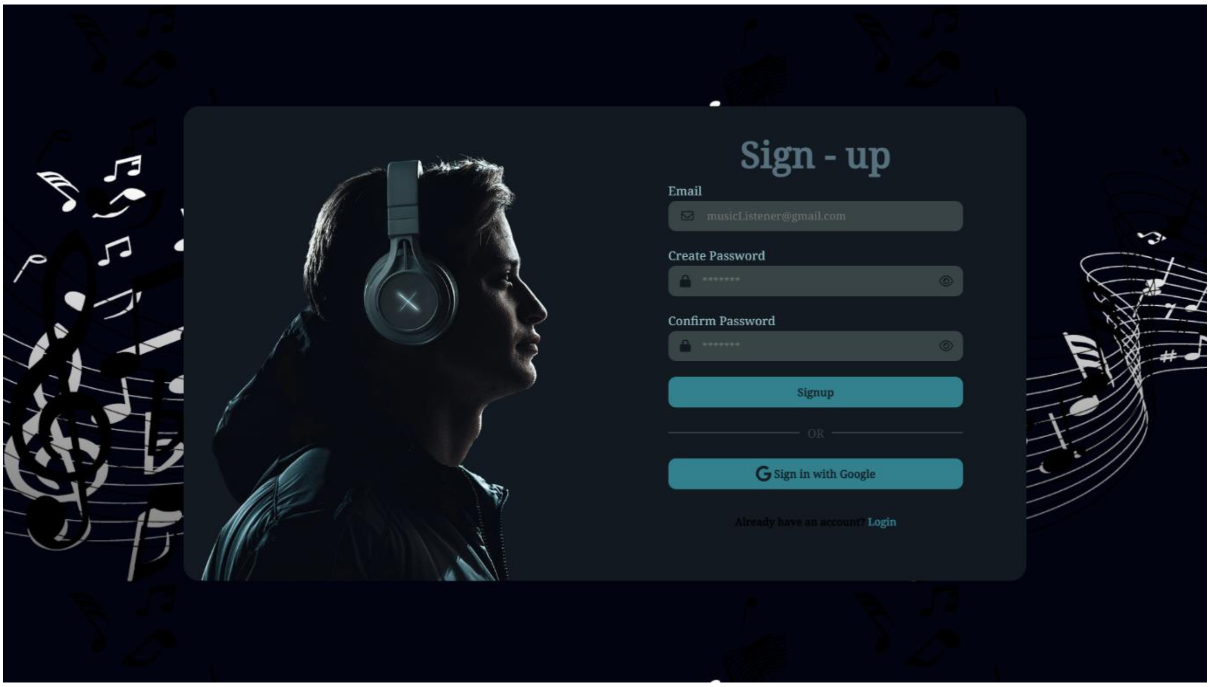
7.Making this system global is our dream.

8.Give the diet schedule for each member online.

Chapter 8

User Manual

Sign-Up:



Sign - up

Email

musicListener@gmail.com

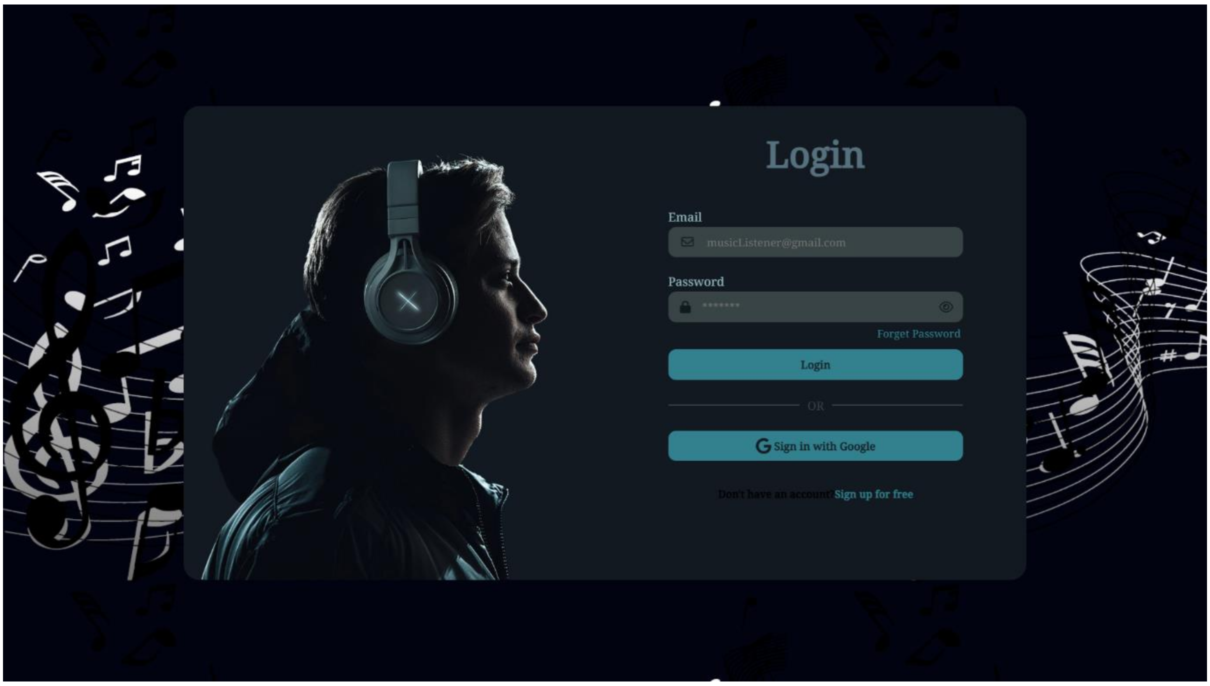
Create Password

****

Confirm Password

****

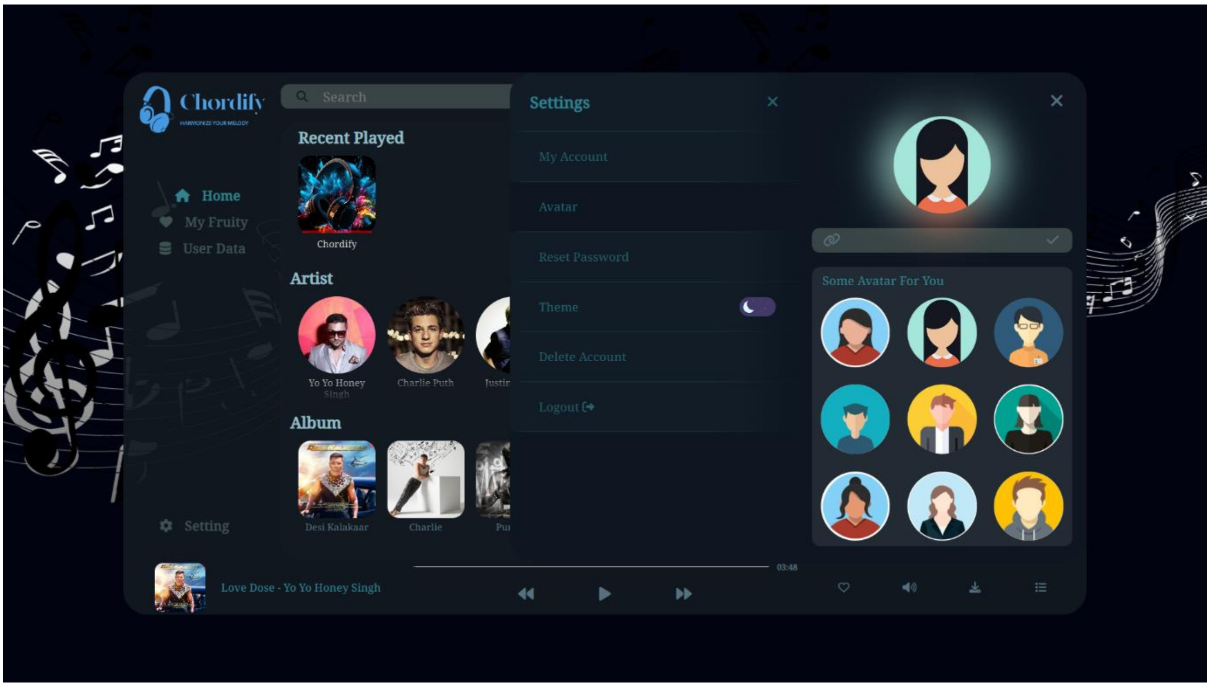
Login:



Email: musicListener@gmail.com

Password: *******

Setting Section:



Chordify
HARMONIZE YOUR MELODY
Search
Home
My Fruity
User Data
Setting
Recent Played
Chordify
Artist
Yo Yo Honey Singh
Charlie Puth
Justin
Album
Desi Kalakaar
Charlie
Pu
Settings
My Account
Avatar
Reset Password
Theme
Delete Account
Logout
Some Avatar For You
03:48
Love Dose - Yo Yo Honey Singh