

## SESSION 2023-2024

### unnamed (1).png SUBJECT:-

Major Project

SUBMITTED BY:- SUBMITTED TO:-

Anukool Sharma HPU

RollNo. 6210010013

# DEPARTMENT OF BCA



**CERTIFICATE FROM PROJECT GUIDE**

**To Whom It May Concern**

This is to certify that Mr**. Anukool Sharma**, a student of BCA (6th Semester), has developed a project under my guidance and supervision in the partial fulfillment for the award of “BACHELOR’S DEGREE IN COMPUTER APPLICATIONS” for the academic session 2023-2024. The submitted project has been found satisfactory. I wish him all the best in his future endeavors.

Mr. Ashutosh Chaudhary

Project Guide

MPGC Amb

### ACKNOWLEDGMENT

I **Anukool Sharma (318**) student of Maharana Pratap Govt. College Amb would like to express my sincere gratitude towards Bachelor of Computer Applications Department.

I would like to thank you Prof. Ajay Kumar Co-Ordinator of BCA/PGDCA for granting me the opportunity to build project for the college. I thank Mr. Ashutosh Chaudhary for their constant support during the project. The project would not have been completed without dedication, creativity and energy which my all fellows provided.

Yours faithfully,

Anukool Sharma

(Final Year BCA )

### DECLARATION

I hereby declare that the project entitled, “**Music Site (Spotify Clone)**” done at place **MPGC College, Amb** has not been in any case duplicated to submit to any other university for the award of any degree.To the best of my Knowledge other than me, no one has submitted to any other university. The project is done in partial fulfilment of the requirements for the award of degree of **BACHELOR OF COMPUTER APPLICATION** to be submitted as final semester project as part of our curriculum.

**Anukool Sharma,** Signature of the Student

**Index**

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Topic** | **Page No.** |
| **1** | **Chapter 1** | **1-5** |
| **2** | **Chapter 2** | **6-9** |
| **3** | **Chapter 3** | **10-18** |
| **4** | **Chapter 4** | **19-27** |
| **5** | **Chapter 5** | **28** |
| **6** | **Chapter 6** | **29** |

**Chapter-1**

# INTRODUCTION

Welcome to “Music Site”. This is the first module in the series we will see “What is Music website and how does it work”. Music site is a digital music services that gives you access to millions of songs from artists all over the world, like other music streaming platform for e.g. Youtube Music, Jio Savaan, Music Mania, Retro music, etc.

Music Mania is immediately appealing because you can access content for free by simply signing up using an Email address or by connecting with Facebook, Gmail Account. If you’re not keen on monthly subscription fees for Music Mania Premium, or just want to dip your toe in and test it out, it’s out, it’s easy to get started and there’s no commitment.

You can find out the main differences between Music Mania Free and Premium in our separate feature but as a quick summary, the free version is ad- supported, much like radio stations. The free version of Music Mania can be accessed on PC, laptop and mobile phone, but the full service needs a Music Mania Premium subscription.

# BACKGROUND

Sounds are all around us, from birds chirping and waves lapping against a coastline to cars honking in traffic. But sometimes sounds are put together in purposefulwaystocreateaspecificatmosphereortoexpressideasoremotions. Such organized sounds are called music.

Music is a collection of coordinated sound or sounds. Making music is the process of putting sounds and tones in an order, often combining them to create a unified composition. People who make music creatively organize sounds for a desired result, like a Beethoven symphony or one of Duke Ellington's jazz songs. Music is made of sounds, vibrations, and silent moments, and it doesn't always have to be pleasant or pretty. It can be used to convey a whole range of experiences, environments, and emotions.

Almost every human culture has a tradition of making music. Examples of early instruments like flutes and drums have been found dating back thousands of years. Ancient Egyptians used music in religious ceremonies. Many other African cultures have traditions related to drumming for important rituals. Today, rock and pop musician ns tour and perform around the world, singing the songs that made them famous. All of these are examples of music.

# PURPOSE

The purpose of this document is to inform user Music Mania is a digital streaming services that gives you access to millions of songs from artists all over the world.

Our mission to unlock the potential of human creativity by giving a million creative artists the opportunity to live off heirartand billions off ans the opportunity to enjoy and be inspired by it.

Music Mania manage and share tracks, including titles, for free, or upgrade to Music Mania free to access exclusive features for music including improved sound quality and an on-demand, offline listening experience.

# 1.4. SCOPE

With Music Mania, it’s easy to find the right music for every moment – on your phone, your computer, your tablet and more.

There are millions of tracks on music mania. So whether you’re behind the wheel, working out, partying or relaxing, the right music or podcast is always at your fingertips. Choose what you want to listen to, or let Music Mania surprise you.

You can also browse through the collections of friends, artists, and celebrities, or create a radio station and just sit back.

# 1.5 APPLICABILITY

The functions of playing music and multimedia have come essential in one device as a smart phone since the smart phone appeared.

It is very convenient, but it contains controversial arguments about sound quality, so many smart phone users use the music player application. By using these music applications, people start to think about the relationship between music playing and sound quality. However, those applications are not perfect, so it is hard to choose a good application.

This thesis is about the advantages of the sound quality of music player applications that are currently sold in Android Market through Right Mark Audio Analyzer program, and plans to suggest android music player application system design by analyzing applications by covering disadvantages of these applications.

## Chapter-2

1. **System Planning**

2.1. Survey of technologies

**1. Front End-:**

**HTML**

Hypertext Markup Language (HTML) is the standar[d markup languag](https://en.wikipedia.org/wiki/Markup_language)e for documents designed to be displayed in a [web browser.](https://en.wikipedia.org/wiki/Web_browser) It can be assisted by technologies such

As [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) and [scripting languages](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScrip](https://en.wikipedia.org/wiki/JavaScript)t.

[Web browser](https://en.wikipedia.org/wiki/Web_browser)s receive HTML documents from a [web server](https://en.wikipedia.org/wiki/Web_server) or from local storage

And [render](https://en.wikipedia.org/wiki/Browser_engine) the documents in to multimedia web pages. HTML describes the structure of a web page [semanticall](https://en.wikipedia.org/wiki/Semantic_Web)y and originally included cues for the appearance of the document.

[HTML elements](https://en.wikipedia.org/wiki/HTML_element) are the building blocks of HTML pages. With HTML constructs,[images](https://en.wikipedia.org/wiki/HTML_element)andotherobjectssuchas[interactiveforms](https://en.wikipedia.org/wiki/Fieldset)maybeembeddedintothe rendered page. HTML provides a means to create [structured documents](https://en.wikipedia.org/wiki/Structured_document) by denoting

Structural [semantics](https://en.wikipedia.org/wiki/Semantics) for text such as headings, paragraphs, lists, [links](https://en.wikipedia.org/wiki/Hyperlink), quotes and other items. HTML elements are delineated by *tags*, written using [angle brackets.](https://en.wikipedia.org/wiki/Bracket) Tags such

as and directly introduce content into the page. Other tags such

<img/>

<input/>

as surroundandprovideinformationaboutdocumenttextandmayincludeothertagsas sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

<p>

HTML can embed programs written in a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript](https://en.wikipedia.org/wiki/JavaScript), which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

**CSS**

Cascading Style Sheets (CSS) is a [stylesheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing

The [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup languag](https://en.wikipedia.org/wiki/Markup_language)e such as [HTML.](https://en.wikipedia.org/wiki/HTML) CSS is a cornerstone technology of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), alongside HTML and [JavaScript](https://en.wikipedia.org/wiki/JavaScript).

CSS is designed to enable these paration of presentation and content,

Including [layout](https://en.wikipedia.org/wiki/Page_layout), [colors](https://en.wikipedia.org/wiki/Color), and [fonts](https://en.wikipedia.org/wiki/Typeface). This separation can improve content [accessibilit](https://en.wikipedia.org/wiki/Accessibility)y, provide more flexibility and control in the specification of presentation characteristics, enable multiple [web page](https://en.wikipedia.org/wiki/Web_page)s to share formatting by specifying the relevant CSS in a separate .css file which reduces complexity and repetition in the structural content as well as enabling the .css file to be [cached](https://en.wikipedia.org/wiki/Cache_(computing)) to improve the page load speed between the pages that share the file and its formatting.

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such a son-screen, in print, by voice (via speech-based browser or [screen reader](https://en.wikipedia.org/wiki/Screen_reader)), and on [Braille-base](https://en.wikipedia.org/wiki/Braille_display)d tactile devices. CSS also has rules for alternate formatting if the content is accessed on a [mobile device](https://en.wikipedia.org/wiki/Mobile_device).

The name *cascading* comes from the specified priority scheme to determine which styleruleappliesifmorethanonerulematchesaparticularelement.Thiscascadingpriority scheme is predictable.

The CSS specifications are maintained by the [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C).

Internet media type ([MIME type](https://en.wikipedia.org/wiki/MIME_media_type)) text/css is registered for use with CSS by RFC

2318(March1998).TheW3C operates a free [CSS validation service](https://en.wikipedia.org/wiki/W3C_Markup_Validation_Service) for CSS documents

###### Language used JavaScript-

JavaScript **(**JS**)** is a lightweight, interpreted, or [just-in-time](https://en.wikipedia.org/wiki/Just-in-time_compilation) compiled programming language with [first-class functions](https://developer.mozilla.org/en-US/docs/Glossary/First-class_Function). While it is most well-known as the scripting language for Web pages, [many non-browser environments](https://en.wikipedia.org/wiki/JavaScript) also use it, such as [Node.js](https://developer.mozilla.org/en-US/docs/Glossary/Node.js), [Apache Couch DB](https://couchdb.apache.org/) and [Adobe Acrobat](http://www.adobe.com/devnet/acrobat/javascript.html). JavaScript is a [prototype-based](https://developer.mozilla.org/en-US/docs/Glossary/Prototype-based_programming), multi-paradigm, single-threaded, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) style.

JavaScript runs on the client side of the web, which can be used to design / program how the web pages behave on the occurrence of an event. JavaScript is an easy to learn and also powerful scripting language, widely used for controlling web page behavior.

**JavaScript is** *not* **"Interpreted Java"**. In a nutshell, JavaScript is a dynamic scripting language supporting [prototype based](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Details_of_the_Object_Model) object construction. The basic syntax is intentionally similar to both Java and C++ to reduce the number of new concepts required to learn the language. Language constructs, such

As if statements, for and while loops, and switch and try...catch blocks function the same as in these languages (or nearly so).

JavaScript can function as both a [procedural](https://en.wikipedia.org/wiki/Procedural_programming) and an [object oriented language](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Introduction_to_Object-Oriented_JavaScript).

Objects are created programmatically in JavaScript, by attaching methods and properties to otherwise empty objects at run time, as opposed to the syntactic class definitions common in compiled languages like C++ and Java. Once an object has been constructed it can be used as a blueprint (or prototype) for creating similar objects.

## Feasibility Study

###### Technical Feasibility

A technical feasibility study assesses the details of how you in tend to deliver a product or service to customers. Think materials, labor, transportation, where your business will be located, and the technology that will be necessary to bring all this together.

###### Economical Feasibility

The degree to which the economic advantages of something to be made, done, or achieved are greater than the economic costs.

###### Financial Feasibility

Financial feasibility focuses specifically on the financial aspect soft he study. It assesses the economic viability of a proposed venture by evaluating the startup costs, operating expenses, cash flow and making a forecast of future performance.

###### Operational Feasibility

Operational feasibility is the measure of how well a proposed systems olves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development.

# CHAPTER-3

###### Requirement and analysis Problem Definition-

The biggest drawback is the low audio quality, MP3 uses the lossy algorithm which deletes the lesser audible music content to reduce the file size, thus compromising on the music quality, Music piracy increased to a greater extent,

Cheaper or free duplicate versions of the original music files are available on the [Internet](https://www.online-sciences.com/technology/what-are-the-advantages-and-disadvantages-of-using-the-internet/) for download .

**The rear some disadvantages of the existing system.**

* The sound quality of the MP3 format is not as good as that of the CD , So , CD players provide clearer audio than do MP3 players , Although MP3s can be compressed at a higher bit rate , Most are encoded at 128 kilobits per second , compared with CDs , on which the listener receives sound at 196 kilobits per second , about 50 per cent higher .
* The data is susceptible to losses due to the malware or virus attacks , The people who used the file-sharing service , They had their computers accessed by the hackers , MP3 players are generally more expensive than CD players.
* MP3 compression may discard as much as 90 percent of the data from the original recording without a significant drop in sound quality , Nevertheless , The listeners with the exceptional hearing or high-end earphones may detect slight differences between the MP3 file & the original uncompressed CD recording .
* Unlike CDs , The albums on MP3s cannot be resold , When the people purchase the song from iTunes or another online MP3 store , They are not so much buying the song as indefinitely leasing it , This may limit the ability of the owners of MP3 players to refresh their libraries frequently , unlike owners of CD players , they cannot legally trade their songs for new ones.

## Requirement Specification-

Music Mania is immediately appealing because you can access content for free by simply signing up using an Email address or by connecting with Facebook, Gmail Account. If you’re not keen on monthly subscription fees for Music Mania Premium, or just want to dip your toe in and test it out, it’s out, it’s easy to get started and there’s no commitment.

You can find out the main differences between Music Mania Free and Premium in our separate feature but as a quick summary, the free version is ad- supported, much like radio stations. The free version of Music Mania can be accessed on PC, laptop and mobile phone, but the full service needs a Music Mania Premium subscription.

##### MODULES OF PROPOSE SYSTEM-

1. **Music-Streaming**

The streaming method doesn’t require the downloading of the entire file. The audio that the user requests is delivered to him in small packets to play the music instantly.

1. **Search**

The entire idea of a music streaming application is to give the listeners the opportunity search for the type of music they want to listen as per their mood.

1. **Playlists**

What could be a better option that giving you user platform where they can create a list of all their preferred tracks in a single spot, classified according to their mood.

1. **Offline-Mode**

This feature permits users to listen to their favorite music even without the internet connection. It utilizes the local storage of the device to cache the audio information.

# SOFTWARE AND HARDWARE REQUIREMENT-

### Hardware Requirement

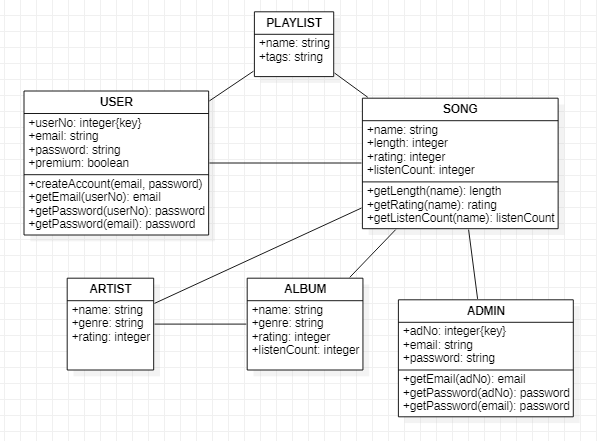
Hardware requirement for this system area’s Follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Processor | RAM | Disk Space |
| Client side | IntelP4or equivalent | 512MB | 2GB |
| IntelP4or equivalent | 512MB | 1GB |
| Server side | Server Environment  Capable H/w | 2GB | As per the size of requirements DBMS |

### Software Requirement

Software Requirement for this system area’s follows:

|  |  |
| --- | --- |
| FRONTEND | Html5,css, JS |
| OPERATING SYSTEM | Windows10 |

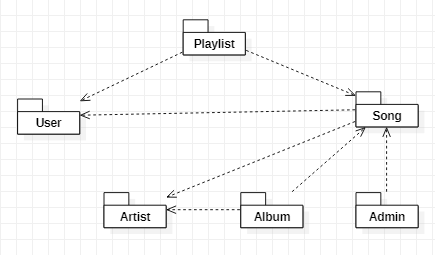


(CLASSDIAGRAM)

###### Package Diagram-:

Package diagram is used to simplify complex class diagrams, you can group classes into packages. A package is a collection of logically

Related UML elements. The diagram below is a business model in which the classes are grouped into packages: Packages appear as rectangles with small tabs at the top.

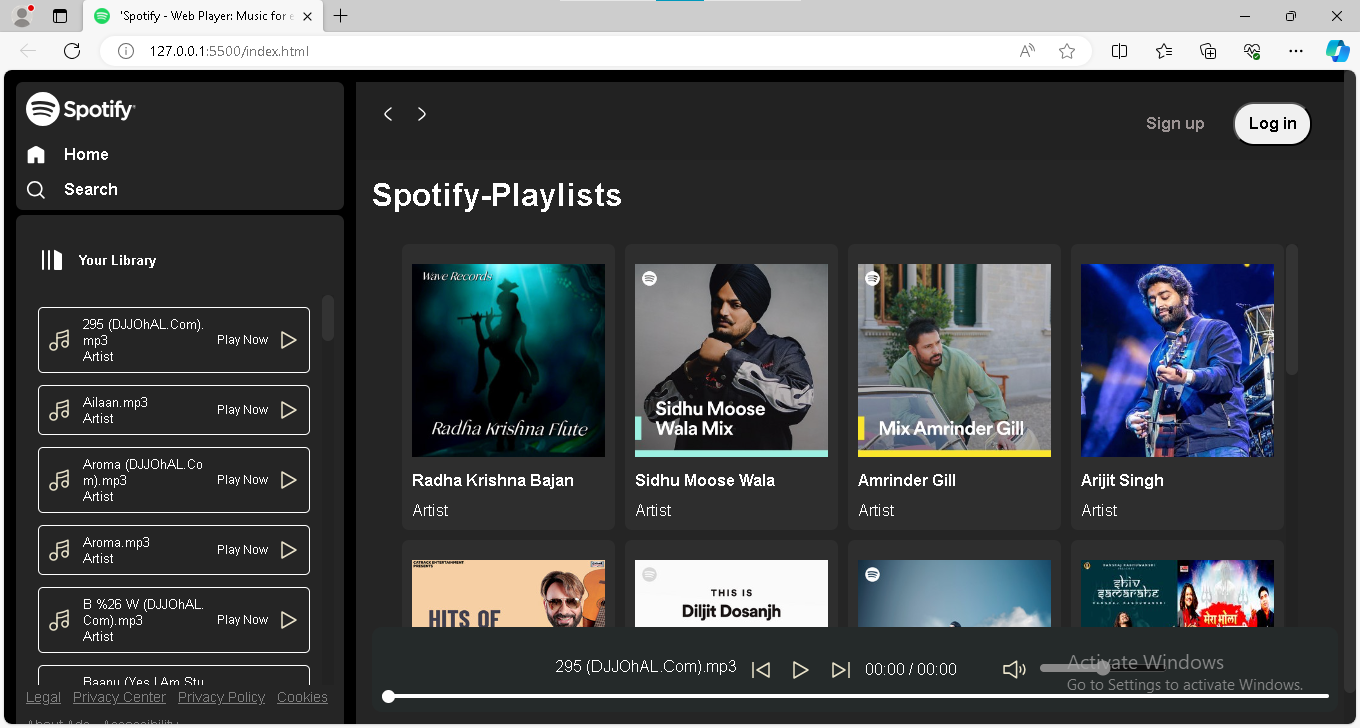


(PACKAGE DIAGRAM)

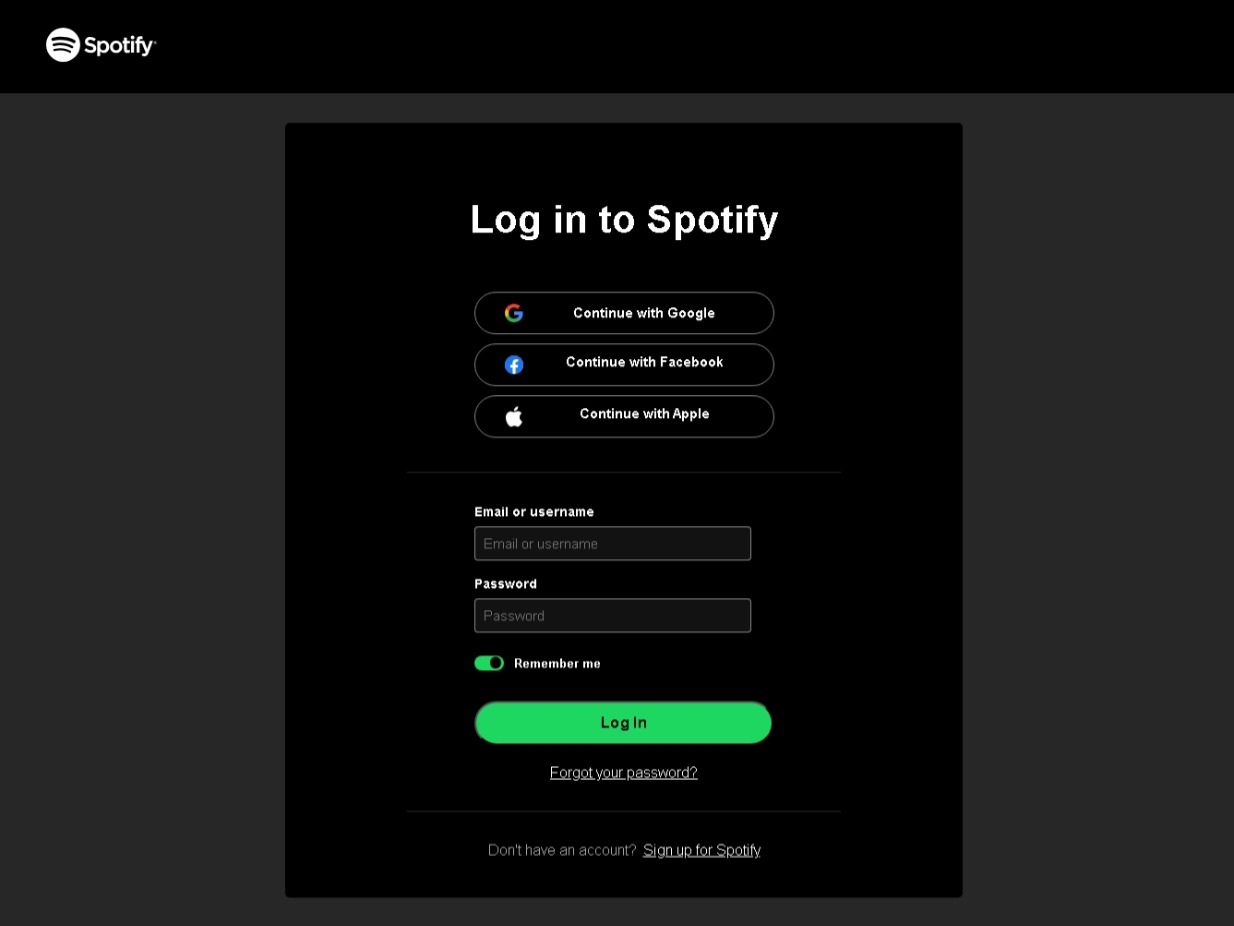
### User interface and Design:

User interface (UI) design is the process of making interfaces in software or computerized devices with a focus on looks or style. Designers aim to create designs userswillfindeasytouseandpleasurable.UIdesigntypicallyreferstographicaluser interfaces but also includes others, such as voice-controlled ones.

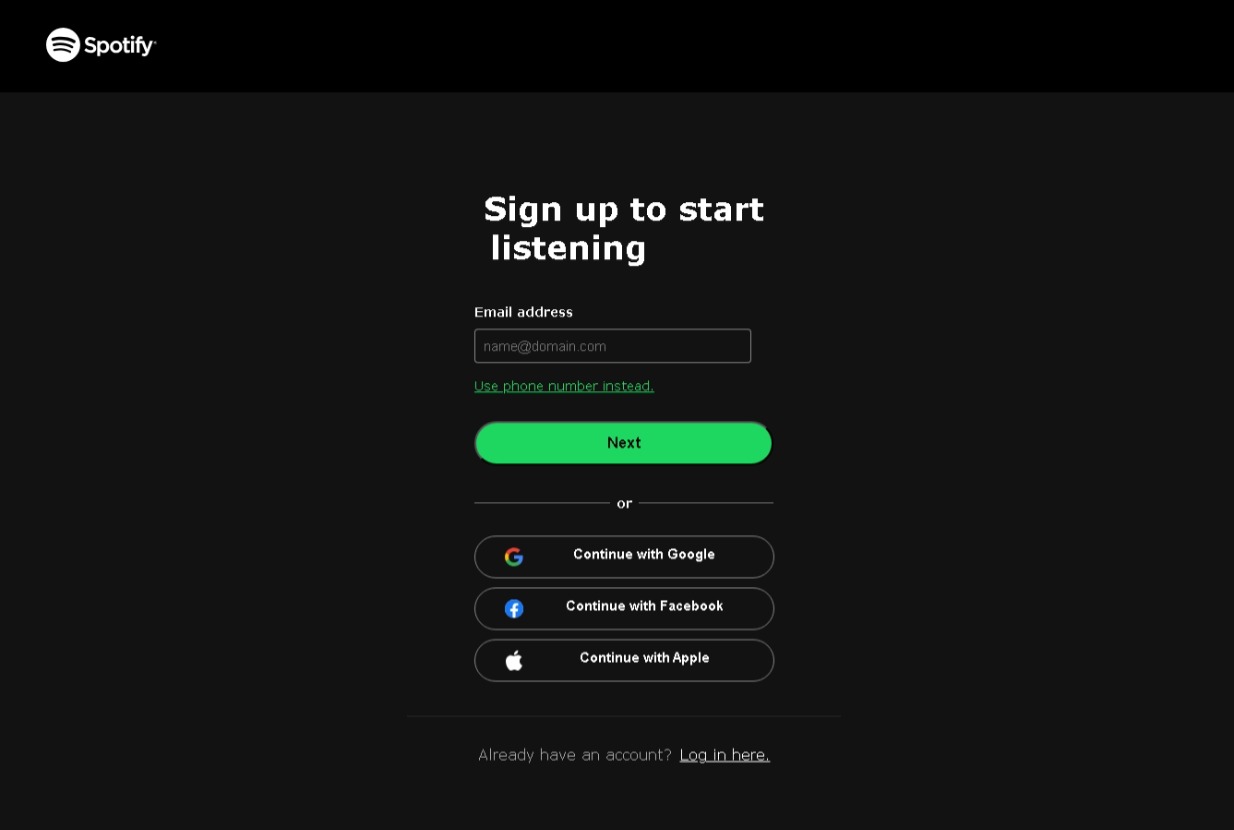
**Homepage-:**



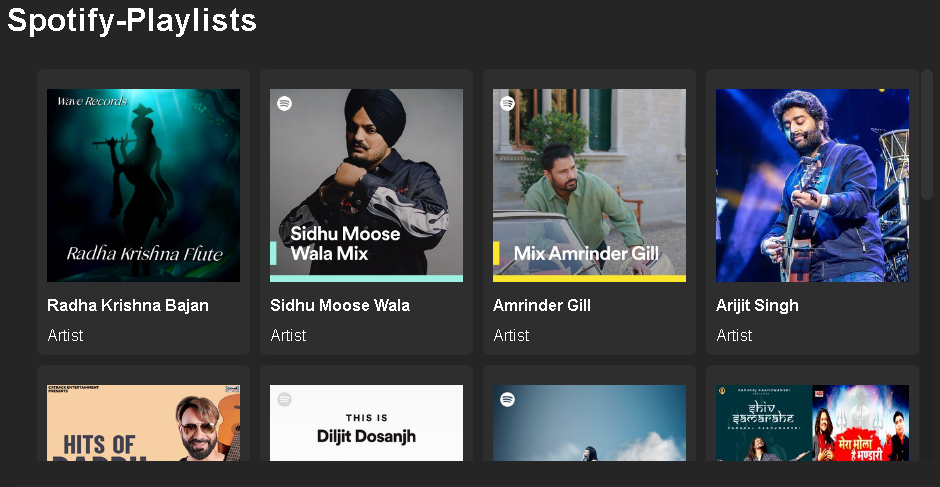
**LOGIN FORM-:**

****

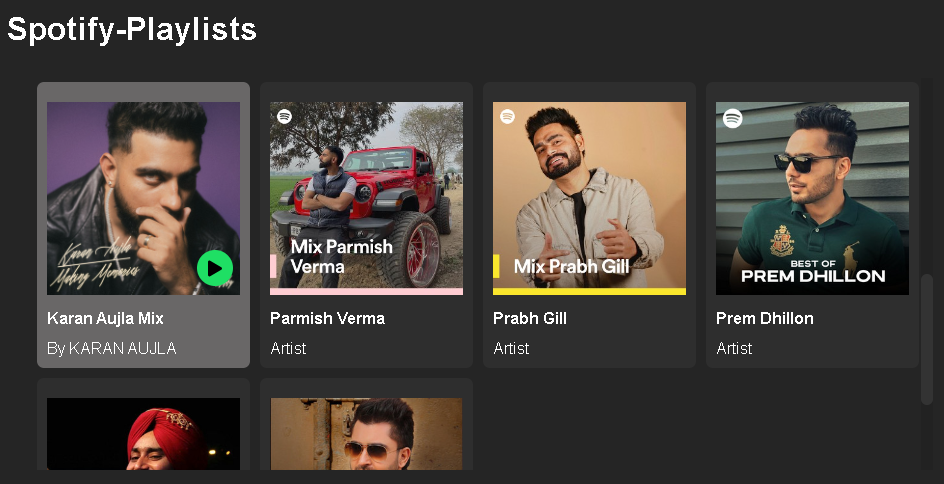
###### Registration Page-:



**Featured-:**

****

**Albums-:**

****

**Chapter 4**

**Code (Index.html)**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>'Spotify - Web Player: Music for everyone'</title>

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

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

</head>

<body>

    <div class="container flex bg-black">

        <div class="left">

            <div class="close">

                <img class="invert" src="./img/close.svg" alt="Close Icon" width="30">

            </div>

            <div class="home bg-grey rounded m-1 p-1">

                <div class="logo"><img class="invert" src="./img/logo.svg" width="110px"

alt="Logo Icon"></div>

                <ul>

                    <li><img class="invert" src="./img/home.svg" alt="Home Icon">Home</li>

                    <li><img class="invert" src="./img/search.svg" alt="Search

Icon">Search</li>

                </ul>

            </div>

            <div class="library bg-grey rounded m-1 p-1">

                <div class="heading">

                    <img class="invert" src="./img/playlist.svg" alt="playlist Icon">

                    <h4>Your Library</h4>

                </div>

                <div class="songList">

                    <ul>

                    </ul>

                </div>

                <div class="footer">

                  <div>

<a href="https://www.spotify.com/jp/legal/"><span>Legal</span></a></div>

                    <div><a href="https://www.spotify.com/jp/privacy/"><span>Privacy

Center</span></a></div>

                    <div><a href="https://www.spotify.com/jp/legal/privacy-

policy/"><span>Privacy Policy</span></a></div>

                    <div><a href="https://www.spotify.com/jp/legal/cookies-

policy/"><span>Cookies</span></a></div>

                    <div><a href="https://www.spotify.com/jp/legal/privacy-

policy/#s3"><span>About Ads</span></a></div>

                    <div>

<a href="https://www.spotify.com/jp/accessibility/"><span>Accessibility</span></a></div>

                </div>

            </div>

        </div>

        <div class="right bg-grey rounded">

            <div class="header">

                <div class="nav">

                    <div class="hamburgerContainer">

<img class="invert hamburger" src="./img/hamburger.svg" alt="Hamburger Icon" width="40px">

                    </div>

                </div>

                <div class="buttons">

                    <button class="signupbtn"><a href="./Sign up pages/sign.html">Sign

-up</a></button>

                    <button class="loginbtn"><a href="Login Pages/Login.html">Log

-in</a></button>

                </div>

            </div>

            <div class="spotifyPlaylists">

                <h1>Spotify-Playlists</h1>

                <div class="cardContainer">

                       <!-- card 1 start from here  -->

                    <div data-folder="RadhaKrishna" class="card">

                        <div class="play"></div>

                        <img src="./img/RadhaKrishna.jpg" alt="Happy Image">

                        <h4>Radha Krishna Bajan</h4>

                        <p>Artist</p>

                    </div>

                       <!-- card 2 start from here  -->

                       <div data-folder="SidhuMooseWala" class="card">

                        <div class="play"></div>

                        <img src="./img/Sidhu Bhai.jpg" alt="Happy Image">

                        <h4>Sidhu Moose Wala</h4>

                        <p>Artist</p>

                    </div>

                <div class="playbar">

                    <div class="seekbar">

                        <div class="circle">

                        </div>

                    </div>

                 <div class="abovebar">

                        <div class="songinfo"></div>

                    <div class="songbuttons">

                        <img id="previous" src="./img/prevsong.svg" alt="Previous button">

                        <img id="play" src="./img/play.svg" alt="Play Button">

                        <img id="next" src="./img/nextsong.svg" alt="Next Button">

                    </div>

                    <div class="timevol">

                        <div class="songtime"></div>

                        <div class="volume">

                            <img src="./img/volume.svg" alt="Volume Icon" width="25">

                            <div class="range">

                                <input type="range" name="volume" id="">

                            </div>

                        </div>

                    </div>

                 </div>

                </div>

            </div>

        </div>

    </div>

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

</body>

</html>

**JAVASCRIPT CODE**

**(script.js)**

console.log('Lets write JavaScript');

let currentSong = new Audio();

let songs;

let currFolder;

function secondsToMinutesSeconds(seconds) {

    if (isNaN(seconds) || seconds < 0) {

        return "00:00";

    }

    const minutes = Math.floor(seconds / 60);

    const remainingSeconds = Math.floor(seconds % 60);

    const formattedMinutes = String(minutes).padStart(2, '0');

    const formattedSeconds = String(remainingSeconds).padStart(2, '0');

    return `${formattedMinutes}:${formattedSeconds}`;

}

async function getSongs(folder){

    currFolder = folder;

    let a = await fetch(`http://127.0.0.1:5500/${folder}/`)

    // let a = await fetch(`http://127.0.0.1:5500/songs/${folder}/`)

    let response = await a.text()

    // console.log(response)

    let div = document.createElement("div")

    div.innerHTML = response;

   let as = div.getElementsByTagName("a")

    songs = []

   for (let index = 0; index < as.length; index++) {

    const element = as[index];

    if(element.href.endsWith(".mp3")){

    songs.push(element.href.split(`/${folder}/`)[1])

    }

   }

     // Show all the songs in the playlist

     let songUL =

document.querySelector(".songList").getElementsByTagName("ul")[0]

        songUL.innerHTML = ""

     for (const song of songs) {

       songUL.innerHTML = songUL.innerHTML  + `<li><img class="invert"

width="34" src="img/music.svg" alt="">

<div class="info">

<div> ${song.replaceAll("%20", " ")}</div>

<div>Artist</div>

</div>

<div class="playnow">

<span>Play Now</span>

  <img class="invert" src="img/play.svg" alt=""></div> </li>`;

     }

       // Attach an event listener to each song

  Array.from(document.querySelector(".songList").getElementsByTagName("li")).forEach(e => {

       e.addEventListener("click", element => {

           playMusic(e.querySelector(".info").firstElementChild.innerHTML.trim())

       })

   })

   return songs;

}

 const playMusic = (track, pause=false) =>{

    // let audio = new Audio("/songs/" + track)

    currentSong.src = `/${currFolder}/` + track

    if(!pause){

    currentSong.play()

    play.src = "img/pause.svg"

document.querySelector(".volume>img").src = document.querySelector(".volume>img").src.replace("mute.svg", "volume.svg")

        }

    })

// Load the playlist whenever card is clicked

Array.from(document.getElementsByClassName("card")).forEach(e => {

        e.addEventListener("click", async item => {

            console.log("Fetching Songs")

            songs = await getSongs(`songs/${item.currentTarget.dataset.folder}`)

            playMusic(songs[0])

        })

    })

// Add event listener to mute the track

document.querySelector(".volume>img").addEventListener("click", e=>{

        if(e.target.src.includes("volume.svg")){

            e.target.src = e.target.src.replace("volume.svg", "mute.svg")

            currentSong.volume = 0;

            document.querySelector(".range").getElementsByTagName("input")[0].value = 0;

        }

        else{

            e.target.src = e.target.src.replace("mute.svg", "volume.svg")

            currentSong.volume = .10;

            document.querySelector(".range").getElementsByTagName("input")[0].value = 10;

        }

    })

}

main();

**cascading style sheet**

**(style.css)**

@import url('https://fonts.googleapis.com/css2?family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap');

:root{

    --a:0;

}

.close {

    display: none;

    cursor: pointer;

}

\*{

    margin: 0;

    padding: 0;

    font-family: "Roboto", sans-serif;

}

body{

    background-color: black;

    color: white;

}

.left{

    width: 25vw;

    padding: 7px;

}

.right{

    width: 75vw;

}

.home ul li{

    display: flex;

    gap: 18px;

    width: 20px;

    list-style-type: none;

    padding-top: 15px;

    font-weight: 700;

    font-style: normal;

}

.heading{

    display: flex;

    gap: 15px;

    width: 100%;

    padding-top: 14px;

    font-weight: bold;

    align-items: center;

    padding: 23px 14px;

    font-size: 13px;

}

.heading img{

    width: 24px;

}

.library{

    min-height: 75vh;

    position: relative;

}

.footer{

    display: flex;

    flex-wrap: wrap;

    gap: 12px;

    position: absolute;

    bottom: 0;

    padding: 12px 0;

    font-size: 14px;

}

.footer a{

    color: grey;

}

.right {

   margin: 12px 0;

   position: relative;

}

.header{

    display: flex;

    justify-content: space-between;

    background-color: rgb(34, 34, 34);

    height: 78px;

}

.header > \*{

    padding: 20px;

}

.spotifyPlaylists{

    padding: 16px;

}

.cardContainer{

    display: flex;

    gap: 10px;

    flex-wrap: wrap;

    margin: 30px;

    max-height: 60vh;

    overflow-y:auto;

}

.card{

    /\* width: 200px; \*/

    width: 193px;

    padding: 10px;

    border-radius: 7px;

    background-color: #2e2e2e;

    position: relative;

## Chapter 5

#### Conclusion And Future Work

The Music Mania website future work is that it will be updated directly proportional with time. The user will have to constantly their work in order to match with trendy designs.

Well talking about future, the website will have a feature where the user add their favorite song in playlist without contacting the web developer for it the each time. The users might get a user panel through which they can easily update their playlist and wherever they want which playlist share on cross platforms.

The website will be updated with many other payment options and new wallet and methods will be added for buying the packages instead of email contact

From this I would like to conclude this topic here by thanking all my professors who have always been there for me as my mentor and helped me out in developing of this beautiful music website.

# CHAPTER 6

REFERENCES

I would like to thank all my supporters who provided me the stuff for the project.

**References that helped me building this website:**

* www.w3schools.com(For HTML, CSS and JS)
* [www.tutorialspoint.com](http://www.tutorialspoint.com)
* www.spotify.com (For website UI inspiration)