SPOTIFY CLONE – WEB APPLICATION

‘SUMMER TRAINING REPORT’ SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

BY

YOGESH(04796302720)

TO

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

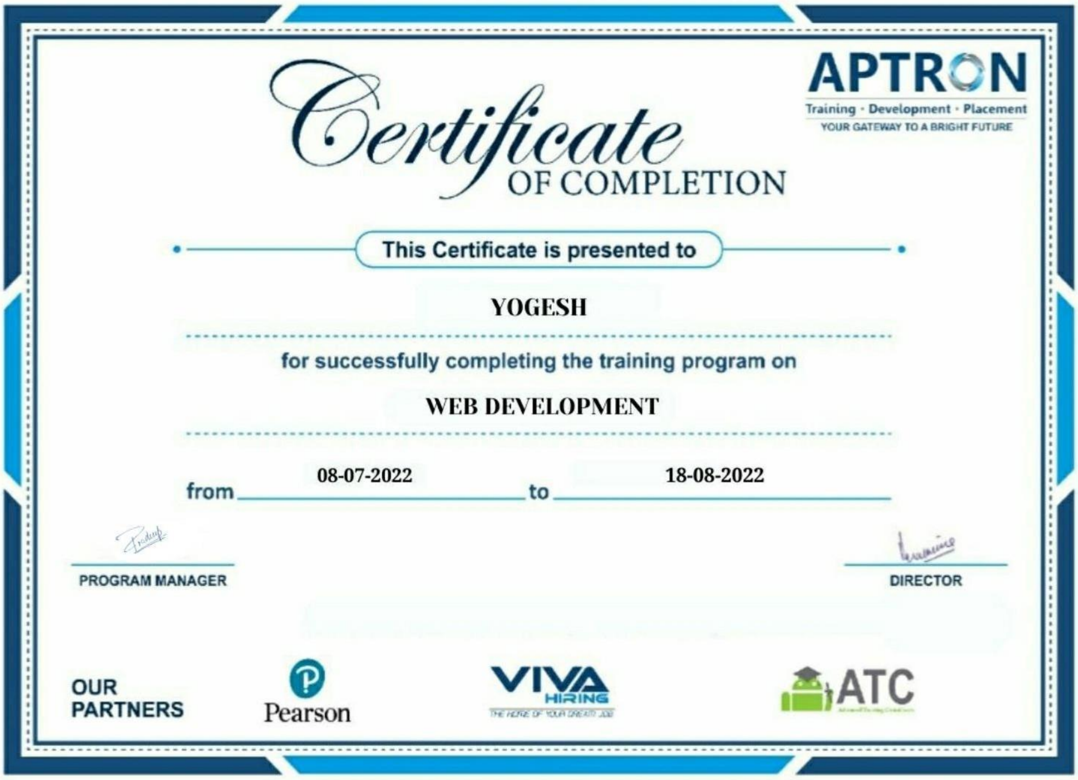
MAHARAJA SURAJMAL INSTITUTE OF TECHNOLOGY

AFFILIATED TO GURU GOBIND SINGH INDRAPRASTA UNIVERSITY

(C-4 MARKET, FIRE STATION RD, JANAKPURI, NEW DELHI, DELHI 110058)

Certificate of Completion of Web Development Course

(Web Development Course by ‘APTRON – TRAINING INSTITUTE IN NOIDA’)



APTRON

Training · Development · Placement

YOUR GATEWAY TO A BRIGHT FUTURE

Certificate

OF COMPLETION

This Certificate is presented to

YOGESH

for successfully completing the training program on

WEB DEVELOPMENT

from 08-07-2022 to 18-08-2022

PROGRAM MANAGER

DIRECTOR

OUR PARTNERS

Pearson

VIVA HIRING

ATC

Course Information

Course Syllabus

WEEK 1

8 hours to complete

Course Overview and Website Structure and Hosting

This first module provides an overview of how websites function, their structure, and the ins and outs of choosing a website name and selecting an online host to house your website. By the end of this module, you'll be able to: find and select a web hosting company; choose an effective domain name; use the host to manage your websites; and discuss how networks and the internet function at a high level.

9 videos , 3 readings, 1 quiz

WEEK 2

8 hours to complete

Designing Your Own Website: HTML Basics

In this module, we'll begin to explore how to design and create websites by exploring the base language used to power all websites: HTML. By the end of this lesson, you'll be able to: identify and use common HTML tags; add an image to a webpage; create HTML-formatted tables; use hyperlinks to connect a series of webpages; upload your finished HTML pages to a web host; and, learn some tips and tricks for styling pages and practicing your coding.

10 videos , 1 reading, 2 quizzes

WEEK 3

6 hours to complete

Introduction to Programming Using JavaScript

Now that you know some basic HTML, it's time to turn our attention to another common scripting language used to make websites dynamic - that is allowing users to interact with your webpages - JavaScript. While learning about JavaScript, you'll also gain some foundational knowledge common to all programming languages. By the end of this module, you'll be able to: discuss what is meant by dynamic content; perform essential programming language tasks; create simple JavaScript programs;

use JavaScript to set up alerts and respond to events, to read input, and to change HTML; and conduct basic JavaScript testing.

9 videos , 1 reading, 2 quizzes

WEEK 4

6 hours to complete

Websites with Style: CSS Properties, Colors and Fonts

While HTML and JavaScript are very useful for web development, they don't exactly make websites look attractive - that's where cascading style sheets, or CSS, comes into play. While HTML is used to build the structure of our pages and JavaScript is used to provide interactive functionality, CSS is used to graphically design and layout webpages. By the end of this module, you'll be able to: discuss common mistakes in designing a website; identify and apply CSS basics like purpose and syntax; use CSS properties to control fonts, colors, layouts, and other common properties; differentiate between in-line, internal, and external CSS; and practice and test your cascading style sheets.

10 videos , 1 reading, 2 quizzes

WEEK 5

6 hours to complete

Creating HTML Forms

In this lesson, we're going to learn to apply all of the skills we've acquired so far to make a very common, and very useful type of webpage - an HTML form. HTML forms are seen everywhere on the internet and are used to capture particular information from users in a variety of ways. By the end of this module you'll be able to: create basic HTML forms; identify and appropriately use different types of HTML input; validate user input; submit HTML forms; and test how your forms are working and troubleshoot errors.

7 videos , 1 reading, 2 quizzes

WEEK 6

10 hours to complete

Creating Web Applications

In our final module, we're going to explore the basics of designing web applications - programming useful and dynamic webpages that allow our users to interact with them. By the end of this lesson you'll be able to: distinguish between client and server-side development; apply advanced JavaScript programming skills to create logic with selection and iteration; create new features for applications; create more advanced HTML forms; and practice and learn additional troubleshooting and problem-solving techniques. The module wraps with a final peer review activity that will draw upon all of the skills you've learned in this course.

8 videos , 1 reading, 2 quizzes.

Acknowledgement

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to express my greatest appreciation to the all individuals who have helped and supported me throughout the project.

I am thankful to my Mentors for their ongoing support during the project, from initial advice, and encouragement, which led to the final report of this project. I would also like to thank Mr. Pradeep. S Sir who was always there for assistance as our Summer Training in charge.

A special acknowledgement goes to my classmates who helped me in completing the project by exchanging interesting ideas and sharing their experience.

I wish to thank my parents as well for their undivided support and interest who inspired me and encouraged me to go my own way, without whom I would be unable to complete my project.

At the end, I want to thank my friends who displayed appreciation to my work and motivated me to continue my work.

I am in debt of all these. Only because of them I was able to create my project and make it good and enjoyable experience.

Abstract

Making an interactive Web Application Spotify Clone using HTML, CSS and JavaScript and visual studio code IDE.

This Spotify Clone web application has been made entirely using HTML CSS and JavaScript with visual studio code as Integrated Development Environment. It is an interactive web application as the user can manipulate its functioning in multiple ways like:

1) Play and Pause the song.

2) Make playlist of their Liked songs

3) Download and listen songs offline.

This web application has been made responsive, means it can configure its appearance by aligning itself with the screen sizes and resolutions of different devices on which it is accessed like on desktop screen or on iPad.

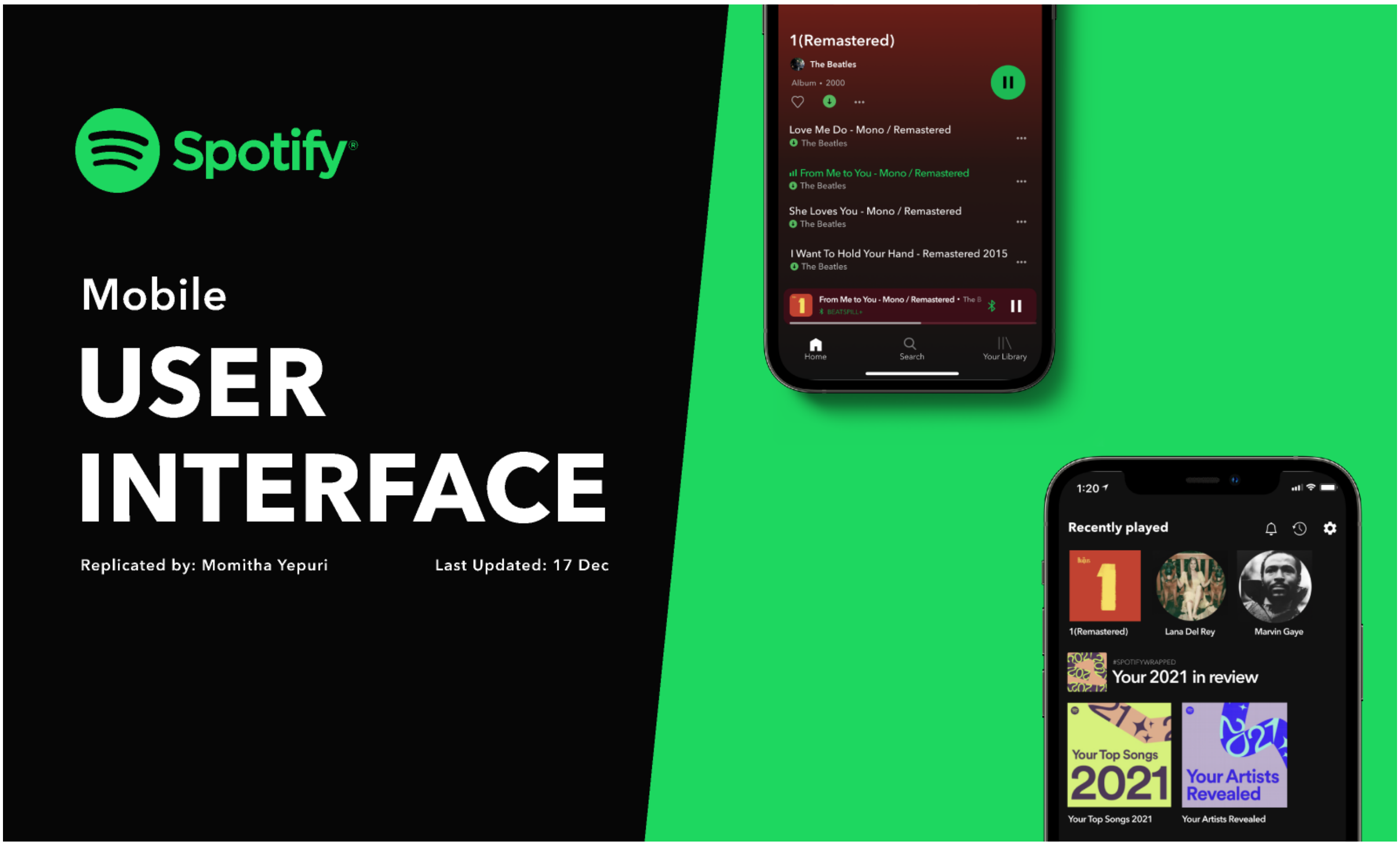
The goal of this project to develop Spotify clone web-application which is interactive and responsive and to practice the web development skills of HTML CSS AND JavaScript which were learned from the Web Development Course by APTRON – TRAINING INSTITUTE IN NOIDA. For this project, I used basic HTML, CSS and JavaScript learned in the Web Development Course.

Table of Contents

CHAPTERS PAGE

Chapter 1 - Introduction

Introduction – 1.1 Vision Statement Of Spotify Face the Music:



First section - Home Page

The Vision Statement of an interactive Web Application for **Spotify Clone** using HTML, CSS and JavaScript and visual studio code IDE.

This spotify clone web application is made entirely using HTML CSS and JavaScript with visual studio code as Integrated Development Environment. It is an interactive web application as the user can manipulate its functioning in multiple ways like:

a) Download and Listen Songs Offline.
b) Make Playlist Of their own music taste

1.2. Components of the vision statement

1.2.1. Concise

The vision statement of Spotify Face the Music is brief and to the point. This means that the company has not used long dialects and dialogues to delivers its opinion ad stance to the public and relevant stakeholders. The vision statement should be brief and comprehensive – it should communicate the essence of the business, and its future plans to help the stakeholders understand its business philosophy and business strategy.

1.2.2. Encompassing description

The vision statement of Spotify Face the Music should be brief but should be holistic in nature. This means that the visions statement should be complete in its description and information of what the company desires, and how it plans to achieve its long term goals strategically. The vision statement should be a comprehensive statement identifying the company's core strengths, which would enable it to achieve its futuristic goals.

1.3. How to develop a vision statement

The company should identify the following t be able to guide its business decisions towards future success and progress to be able to develop a successful vision statement:

1.3.1. Look at the organizational history

- When did Spotify Face the Music start the business?
- How many employees did the company have during the beginning?
- The significant milestones that Spotify Face the Music has achieved since being started, and when were these milestones achieved?
- Include all positive and negative milestones that Spotify Face the Music has faced, and how it overcame them?
- What does Spotify Face the Music enjoy the most about its business and why?

These questions will help the management and key decision makers at Spotify Face the Music to critically assess the history and the various decisions that were made for Spotify Face the Music and the business. These will be critically reviewed for the positive on the negative consequences they brought, and how that influenced the general business direction or Spotify Face the Music to have it stand in its present position.

1.3.2. Look at the present

- What is the business landscape for Spotify Face the Music currently?
- How many employees does the business have currently?
- What is the unique value proposition offered by Spotify Face the Music
- What are the strengths, weaknesses, opportunities and threats being faced by Spotify Face the Music in the present?
- What is the financial strength of the company in the present?

These questions will help strategic managers and decision-makers identify the need of the business to progress, as well as identify the resources needed for advancement. In addition, these questions will also help Spotify Face the Music in deciding the future direction it wants to take on, and how to achieve and realize that direction. Lastly, assessment of present standing for Spotify Face the Music will help the business identify how it can improve the business with potential changes.

1.3.3. Look at the future of the organization

- What are the long term goals for Spotify Face the Music?
- What are the short term goals for Spotify Face the Music?
- How can the company improve its offerings?
- Which processes and ideas can be implemented to help the business gain higher effectivity and efficiency?
- What is the potential for growth for Spotify Face the Music?
- What are the secondary goals of the business?

2.2. Components of a mission statement

2.2.1. Customer satisfaction

The mission statement of Spotify Face the Music focuses on addressing issues of customer satisfaction. The mission statement of Spotify Face the Music has identified its target customer groups, and also identified their needs and demands. The mission statement reflects on how its products and services work towards increasing customer satisfaction for its target customers.

2.2.2. Based on core competencies

The mission statement of Spotify Face the Music is based on its integral strengths and competencies. This is important for Spotify Face the Music as the mission statement will highlight the different systems and processes as well as strategic tactics that the company uses to achieve its organizational and strategic goals. The achievement of the goals will depend on how well Spotify Face the Music makes use of its core competencies.

2.2.3. Realistic and clear

The mission statement for Spotify Face the Music is also realistic and clear. This means that Spotify Face the Music has used simple, string, and easily understood words and phrases in the drafting of its mission statement. Clarity is important so that the mission statement is understood by all relevant stakeholders of Spotify Face the Music Company. Spotify Face the Music's mission statement is also realistic, which makes it able to achieve various set goals and targets.

2.2.4. Motivational and inspirational

The mission statement of Spotify Face the Music is motivational in that it works towards inspiring the employees and the workforce towards giving their optimal best performance towards the goal achievement of Spotify Face the Music. The mission statement of Spotify Face the Music is also inspirational in that it develops the need for growth and progress in individuals – for the betterment of not only the company but also for their own selves.

2.2.5. Specific and sharp

The mission statement of Spotify Face the Music is precise and to the point. It is easy to understand and delivers what the audience must know about Spotify Face the Music's offerings and operations. It is important to keep the missions statement short, sharp and precise to be able to successfully communicate the company's standing to stakeholders, instead of dragging it on into long pages with repetition and non-important aspects.

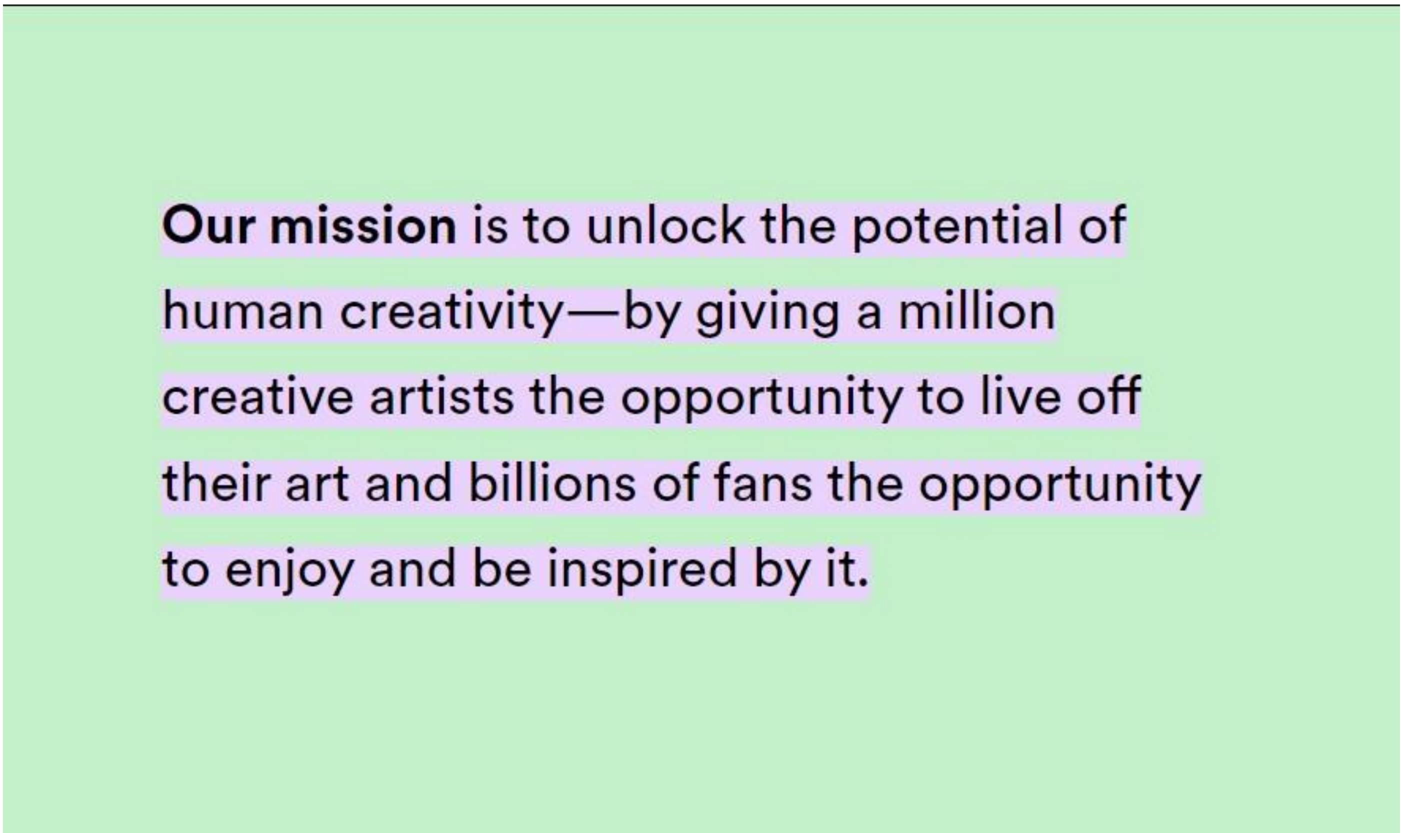
2.2.6. Reflects the company's offerings

The mission statement of a company should be based on what the company has to offer in terms of products and services. This means that the mission statement for Spotify Face the Music highlights its offerings, but ensures that this offering is in line with the values that the company stands for. The mission statement for Spotify Face the Music, therefore, identifies the ethical grounds through which the company systematically works to deliver its offering.

2.2.7. How does Spotify Face the Music manage to achieve its promised offering?

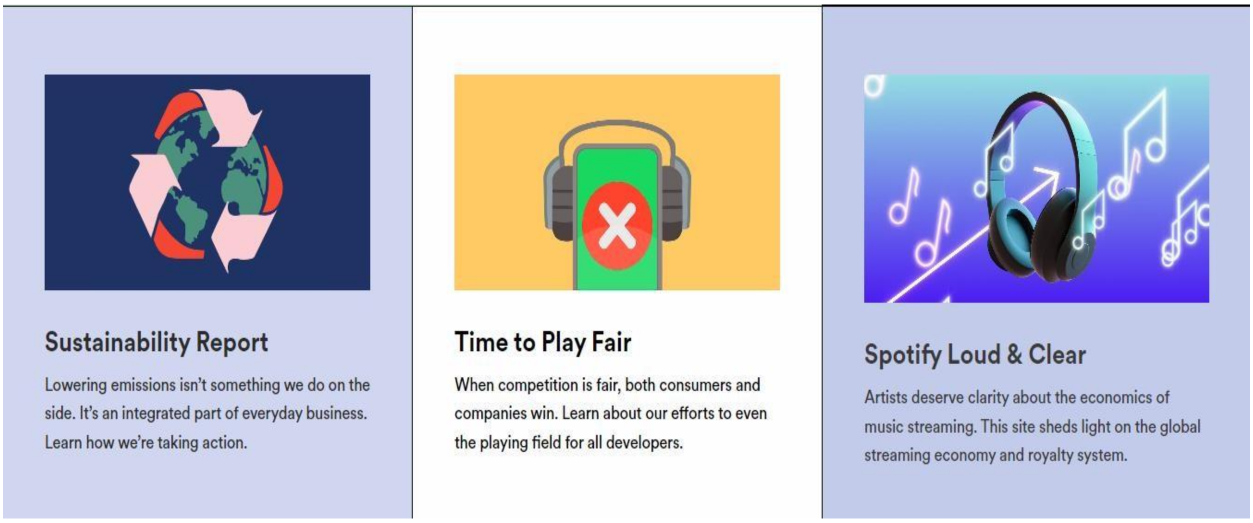
1) Explain the systems and operations employed at Spotify Face the Music
2) Identify relevant ethical policies in place at Spotify Face the Music
3) Highlight the use of transparency at Spotify Face the Music in all matters
4) Define the processes that Spotify Face the Music uses to deliver its promised offering to targetcustomer groups.

Mission Of Spotify Clone: -



Our mission is to unlock the potential of human creativity—by giving a million creative artists the opportunity to live off their art and billions of fans the opportunity to enjoy and be inspired by it.

Figure 1.1



Sustainability Report

Lowering emissions isn't something we do on the side. It's an integrated part of everyday business. Learn how we're taking action.

Time to Play Fair

When competition is fair, both consumers and companies win. Learn about our efforts to even the playing field for all developers.

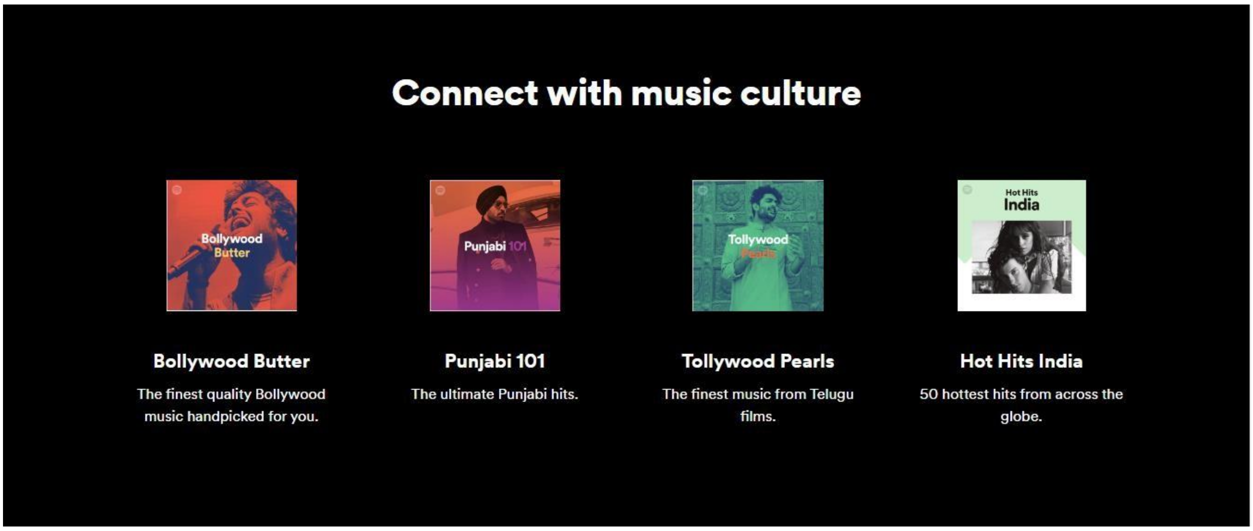
Spotify Loud & Clear

Artists deserve clarity about the economics of music streaming. This site sheds light on the global streaming economy and royalty system.

Figure 1.2

Benefits of Spotify Clone:

1) **Connect with the culture:-**



Connect with music culture

Bollywood Butter

The finest quality Bollywood music handpicked for you.

Punjabi 101

The ultimate Punjabi hits.

Tollywood Pearls

The finest music from Telugu films.

Hot Hits India

50 hottest hits from across the globe.

Figure 2.1

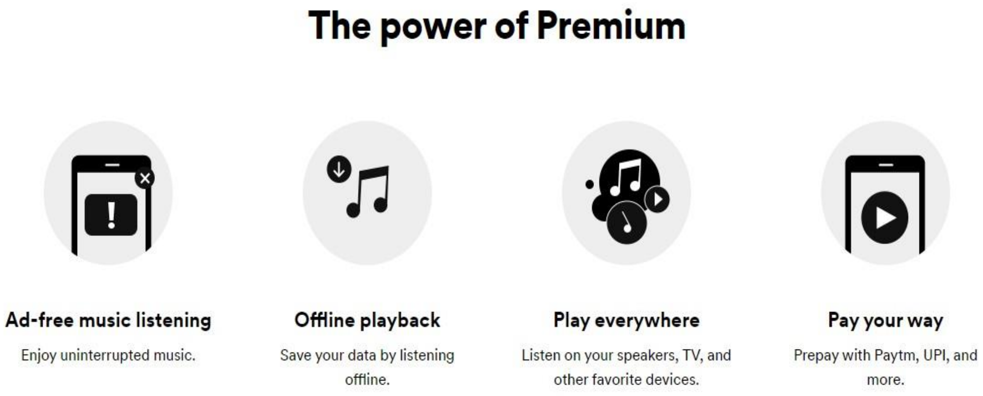
2) **If You Buy the Premium Version of Spotify**

a) **Ad-Free Music Listening –** Enjoy uninterrupted music.

b) **Offline playback -** Save your data by listening offline.

c) **Play everywhere -** Listen on your speakers, TV, and otherfavorite devices.

d) **Pay your Way -** Prepay with Paytm, UPI, and more.



The power of Premium

Ad-free music listening

Enjoy uninterrupted music.

Offline playback

Save your data by listening offline.

Play everywhere

Listen on your speakers, TV, and other favorite devices.

Pay your way

Prepay with Paytm, UPI, and more.

Figure 2.2

Software Used

- ‘Visual Studio Code’ I.D.E (Integrated Development Environment)
- Photo Editor to adjust the configuration of the cover images and background.
- HTML, CSS and JavaScript programming (skill) and React.js Framework.

About Organization:-

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

There are millions of tracks and episodes on Spotify. 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 Spotify surprise you. You can also browse through the collections of friends, artists, and celebrities, or create a radiostation and just sit back. Soundtrack your life with Spotify. Subscribe or listen for free

Major JavaScript Application areas

The working of various buttons like play, pause, previous and next is set using JavaScript.

The seek bar updates as the music plays and is used for reaching to specific destination in the song just by dragging it to the required place. Individual song panels have a play button on them which when clicked plays that particular song and turnsinto a pause button. When some other song play button clicked, the previous song pause button turns into Emmet comes built in visual studio code when ‘!’ is typed, the general structure of an html web pagecomes as dropdown on the screen.

CSS specification:-

The justify-content property used in the flex boxes, aligns the flexible container's items when the items do not use all available space on the main-axis (horizontally). “fontawesome.com” website with icon set and toolkit has been used for importing vector icons and sociallogos on the web application using link of personal took kit.

The cursor to the icons like the play, backwards and forwards has been made pointer by targeting the iconclass. Use of span tag in html to hold the song items: The <span> tag is an inline container used to mark up a partof a text, or a part of a document.

The <span> tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.
The <span> tag is much like the <div> element, but <div> is a block-level element and <span> is an inline element.

The bottom bar is made sticky. The ‘playing’ gif is executed only when a song is playing and it stops appearing when the song is stopped. The seek bar has been named as myProgressBar and the cursor becomes a pointer when hovered on it, this isdone to facilitate reaching a desired part of the lecture while playing.

Media Queries is an enhancement of the @media rules of CSS and the “media” attribute in HTML. It adds parameters such as size of display, color depth and aspect ratio. This is because within a class of media (such as TV sets) there can still be important variations. It is related to the work on CC/PP, but is a much more light-weight and limited solution.

Media Queries defines a syntax for short expressions that describe required features of media (or devices), e.g.: minimum or maximum screen size, color capabilities, resolution, aspect ratio, type of pointing device, viewing environment, scripting capabilities, etc. Media Queries is related to the work on CC/PP, but is a more light-weight and limited solution.

Such expressions can be attached as labels to style sheets or other resources, to indicate what media they are designed for. They are used, e.g., in HTML (in the media attribute). CSS uses them on '@import' and '@media' and they occur in similar ways in SVG and generic XML.

Media Queries level 4 is an extended version of the first Media Queries. It adds a handful of new media features, such as 'pointer' and 'hover' (for capabilities of the pointing device) and 'block-overflow' (for paged vs. scrolling media), which provide more precise information about the media than the old 'handheld' vs. 'screen' and 'projection' vs 'screen' distinctions.

Chapter 2- Project Design

Pure html, CSS and JavaScript is used without any readymade development framework so that the even the basic - ground level concepts can be applied for a deeper understanding of the website development work.

Audio API of JavaScript is used to add the songs in sequence using the loop.

The functioning of various buttons in the navigation like play, pause, previous and next is set using JavaScript. The seek bar updates as the song plays and is used for reaching to specific destination in the songs just by dragging it to the required place. Individual song panels have a play button on them which when clicked plays that particular song and turns into a pause button. When some other song's play button is clicked, the previous song's pause button turns into Emmet comes built in in visual studio code so when '!' is typed, the general structure of an html web page comes as dropdown on the screen.

This CSS file is linked to the HTML file:

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

This JavaScript file is linked to the HTML file:

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

These 2 files from 'fontawesome.com' are linked to the HTML file:

```
<script src="https://kit.fontawesome.com/26504e4a1f.js" cross origin="anonymous"></script>
<script src="https://kit.fontawesome.com/e479bf1d64.js" cross origin="anonymous"></script>
```

They include the icons for play, pause, forward and reverse. These buttons/icons are used in the HTML file as:

In the bottom bar:

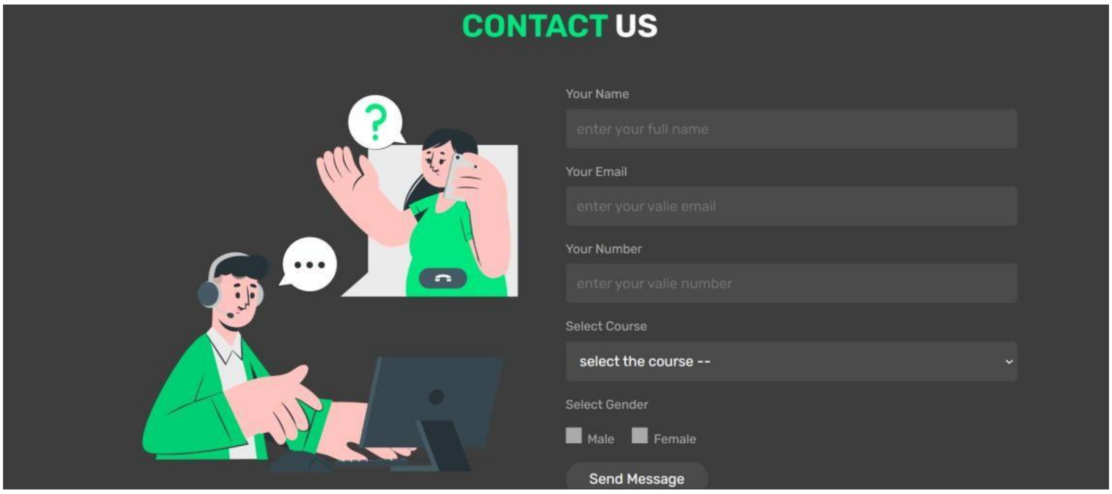
```
<div class="icons">
  <i class="fas fa-2x fa-step-backward" id="previous"></i>
        <i class="far fa-2x fa-play-circle" id="masterPlay"></i>
        <i class="fas fa-2x fa-step-forward" id="next"></i>
</div>
```

In the individual song icons:

```
<div class="songItem">
            <img alt="1">
            <span class="songName">Let me Love You</span>
            <span class="songlistplay"><span class="timestamp">03:44 </span><i id="29" class="far fa-2x songItemPlay fa-play-circle"></i></span>
            </div>
```

Google fonts are imported in the CSS file by writing their links at the top of the CSS file:

```
@import url('https://fonts.googleapis.com/css2?family=Ubuntu&display=swap');
@import url('https://fonts.googleapis.com/css2?family=Varela+Round&display=swap');
```



CONTACT US

Your Name

enter your full name

Your Email

enter your valie email

Your Number

enter your valie number

Select Course

select the course --

Select Gender

□ Male □ Female

Send Message

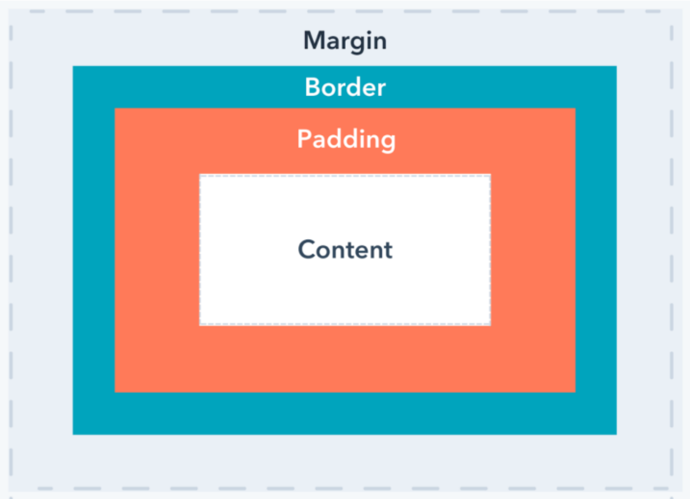
Figure 3.0

Contact us Page

Use of display flex in CSS :

It is used to make flex box/container. A flex container expands items to fill available free space or shrinks them to prevent overflow. Most importantly, the flex box layout is direction-agnostic as opposed to the regular layouts (block which is vertically-based and inline which is horizontally-based).

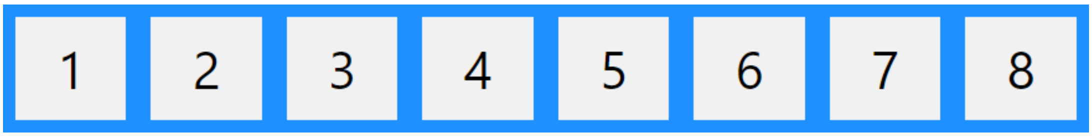
The flex property in CSS is the combination of flex-grow flex-shrink, and flex-basis property. It is used to set the length of flexible items. The flex property is much responsive and mobile friendly. It is easy to positioning child elements and the main container. For example: the individual songItem is included in flex boxes.

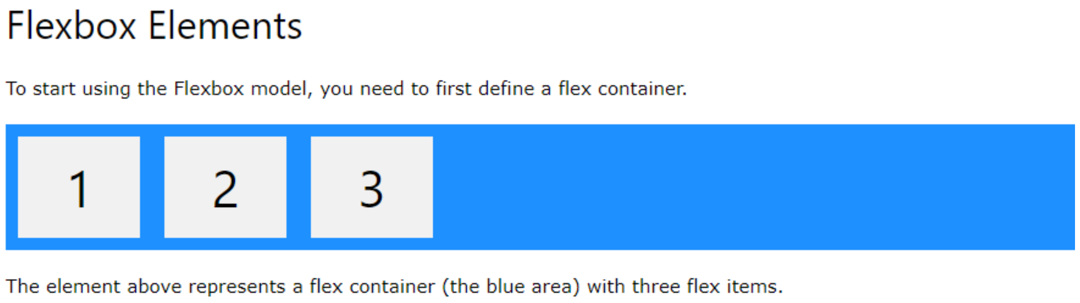


Difference between padding and margin in CSS

In CSS, a margin is the space around an element's border, while padding is the space between an element's border and the element's content.

CSS FLEX BOX PROPERTIES:





Flexbox Elements

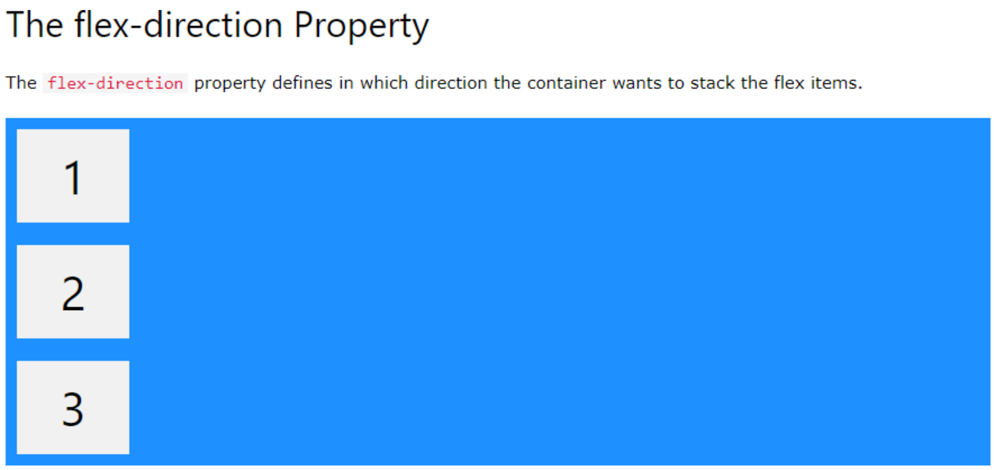
To start using the Flexbox model, you need to first define a flex container.

The element above represents a flex container (the blue area) with three flex items.

EXAMPLE:

A flex container with three flex items:

```
<div class="flex-container">
    <div>1</div>
    <div>2</div>
    <div>3</div>
</div>
```



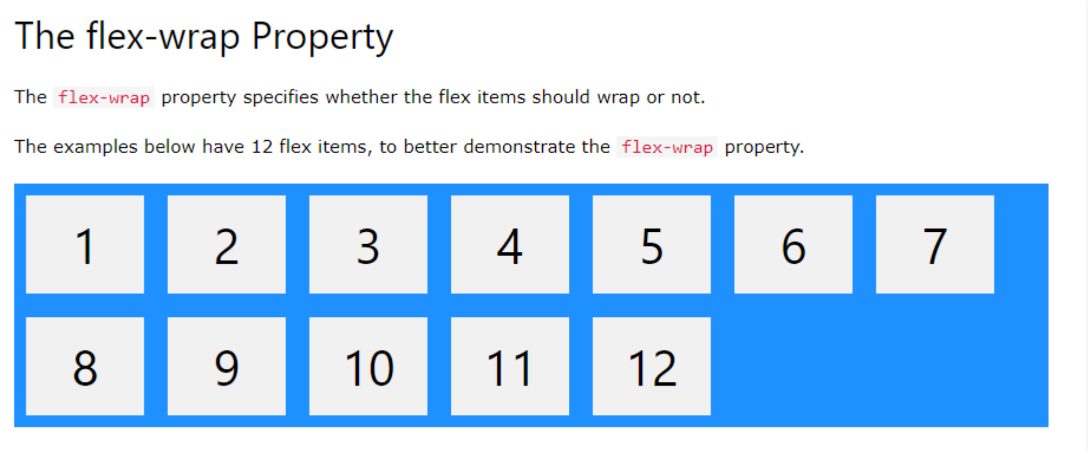
The flex-direction Property

The `flex-direction` property defines in which direction the container wants to stack the flex items.

Example

The column value stacks the flex items vertically (from top to bottom):

```
.flex-container {
   display: flex;
flex-direction:
   column;
   }
```



The flex-wrap Property

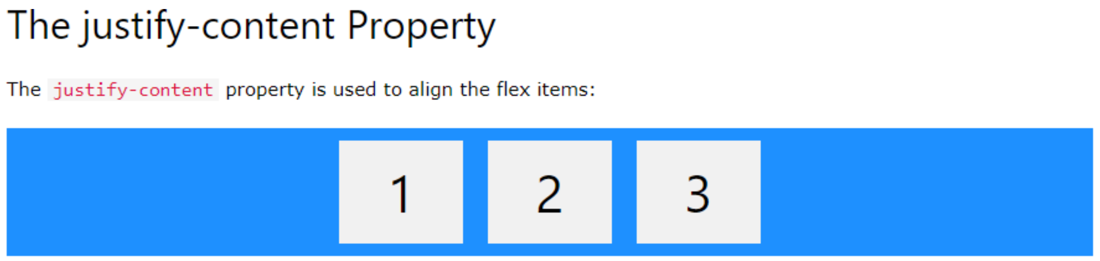
The `flex-wrap` property specifies whether the flex items should wrap or not.

The examples below have 12 flex items, to better demonstrate the `flex-wrap` property.

Example

The wrap value specifies that the flex items will wrap if necessary:

```
.flex-container { display: flex; flex-wrap: wrap;
}
```



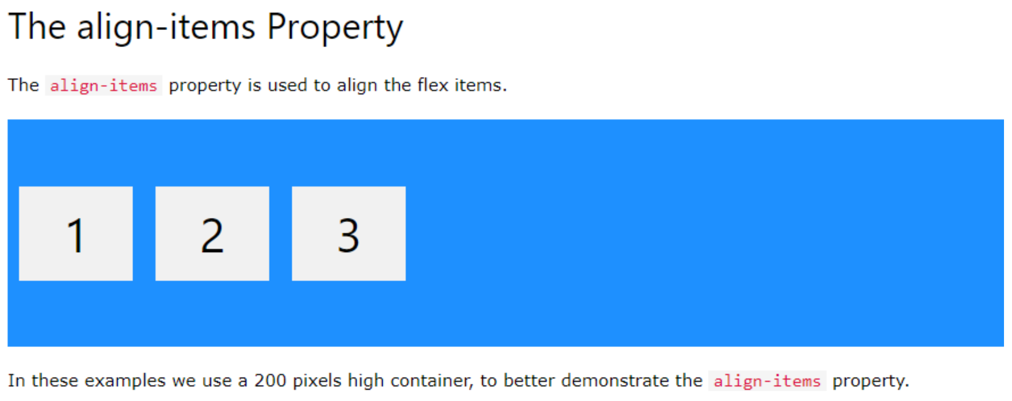
The justify-content Property

The `justify-content` property is used to align the flex items:

Example

The center value aligns the flex items at the center of the container:

```
.flex-container {
    display: flex;
    justify-content: center;
}
```



The align-items Property

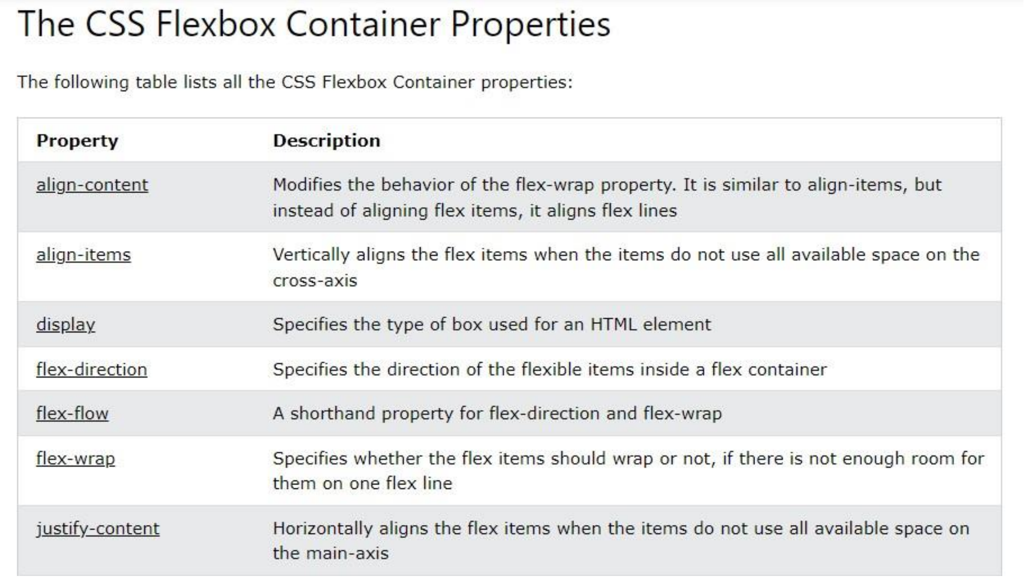
The `align-items` property is used to align the flex items.

In these examples we use a 200 pixels high container, to better demonstrate the `align-items` property.

Example

The center value aligns the flex items in the middle of the container:

```
.flex-container { display:
    flex; height: 200px; align-
    items: center;
}
```

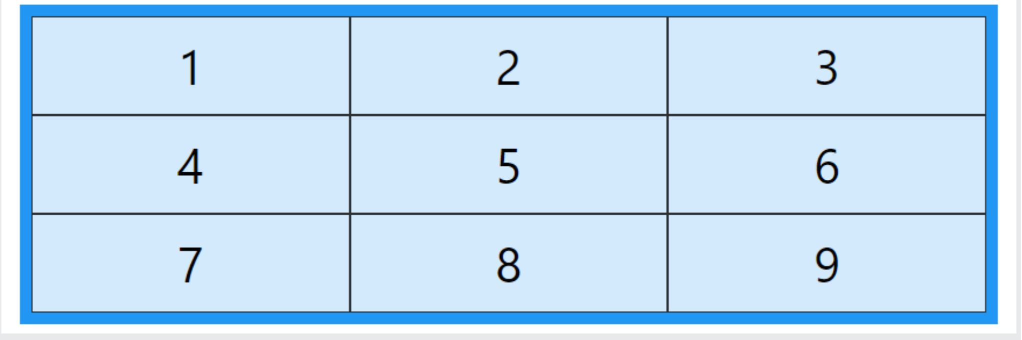


The CSS Flexbox Container Properties

The following table lists all the CSS Flexbox Container properties:

Property	Description
align-content	Modifies the behavior of the flex-wrap property. It is similar to align-items, but instead of aligning flex items, it aligns flex lines
align-items	Vertically aligns the flex items when the items do not use all available space on the cross-axis
display	Specifies the type of box used for an HTML element
flex-direction	Specifies the direction of the flexible items inside a flex container
flex-flow	A shorthand property for flex-direction and flex-wrap
flex-wrap	Specifies whether the flex items should wrap or not, if there is not enough room for them on one flex line
justify-content	Horizontally aligns the flex items when the items do not use all available space on the main-axis

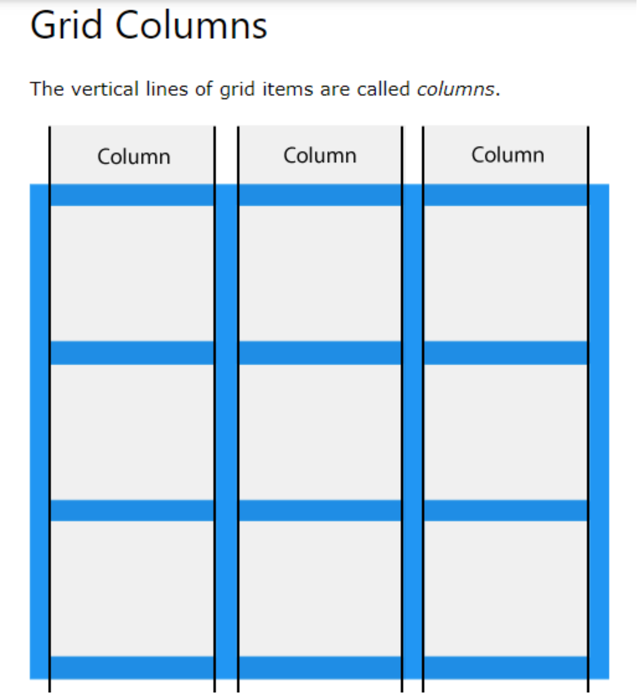
CSS GRID PROPERTIES :



1	2	3
4	5	6
7	8	9

Example

```
<div class="grid-container">
    <div class="grid-item">1</div>
    <div class="grid-item">2</div>
    <div class="grid-item">3</div>
    <div class="grid-item">4</div>
    <div class="grid-item">5</div>
    <div class="grid-item">6</div>
    <div class="grid-item">7</div>
    <div class="grid-item">8</div>
    <div class="grid-item">9</div></div>
```



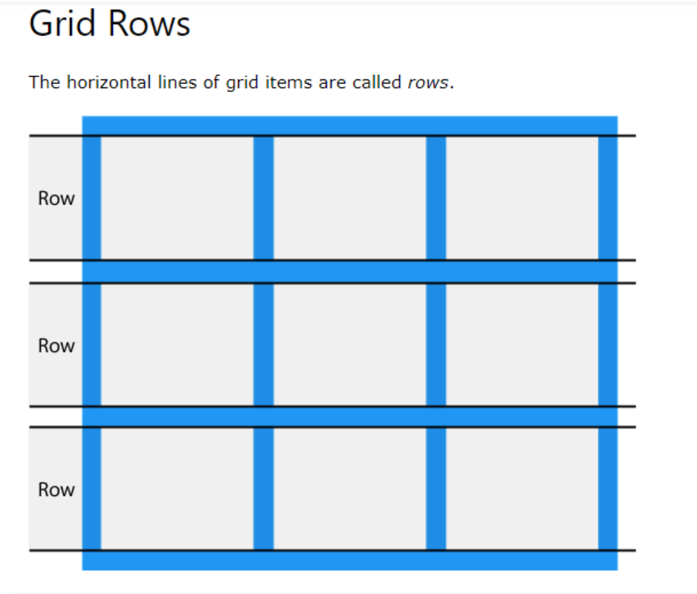
Grid Columns

The vertical lines of grid items are called *columns*.

Example

The column-gap property sets the gap between the columns:

```
.grid-container { display:
    grid; column-gap:
    50px;
}
```



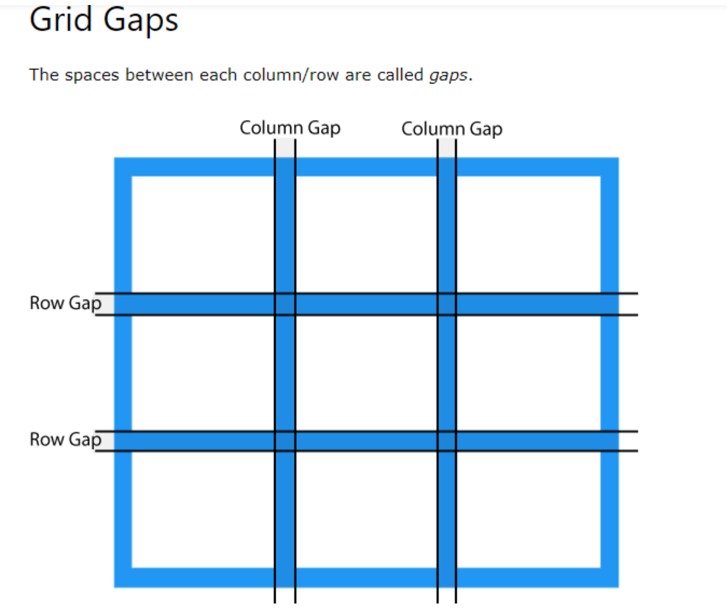
Grid Rows

The horizontal lines of grid items are called *rows*.

Example

The row-gap property sets the gap between the rows:

```
.grid-container {display:
    grid; row-gap: 50px;
}
```



Grid Gaps

The spaces between each column/row are called *gaps*.

Example

The column-gap property sets the gap between the columns:

```
.grid-container {
   display: grid;
   column-gap: 50px;
}
```

The row-gapproperty sets the gap between the rows:

```
.grid-container {
   display: grid; row-
   gap: 50px;
}
```

The gapproperty is a shorthand property for the row-gapand the column-gapproperties:

```
.grid-container {
   display: grid; gap:
   50px 100px;
}
```

Font Size REM:

.rem measure is used to define the text size in ‘brand’ class of CSS.

1 rem unit is equal to the computed value of font-size on the root element. This means that 1rem equals the font size of the html element (which for most browsers has a default value of 16px).viewport units in CSS are truly “responsive length units” in the sense that their value changes every time the browser resizes. Viewport Height (vh). This unit is based on the height of the viewport. A value of 1vh is equal to 1% of the viewport height.

Other relative units of measurement used in CSS are:

Relative Lengths

Relative length units specify a length relative to another length property. Relative length units scale better between different rendering medium.

Unit	Description
em	Relative to the font-size of the element
ex	Relative to the x-height of the current font
ch	Relative to the width of the "0"
rem	Relative to font-size of the root element
vw	Relative to 1% of the width of the viewport*
vh	Relative to 1% of the height of the viewport*
vmin	Relative to 1% of viewport's* smaller dimension
vmax	Relative to 1% of viewport's* larger dimension
%	Relative to the parent elemen

The song progress bar (seek bar) at the bottom of the page is induced by:

```
<input type="range" name="range" id="myProgressBar" min="0" value="0" max="100">
```

It is given minimum to maximum values of 0 to 100.

The position of the bottom bar and the navigation bar is made sticky in CSS i.e. they will stay at their original positions irrespective of the scrolling in the page.

```
position: sticky;
```

The justify-content property used in the flex boxes, aligns the flexible container's items when the items do not use all available space on the main-axis (horizontally).

fontawesome.com website with icon set and toolkit has been used for importing vector icons and social logos on the web application using link of personal took kit.

The cursor to the icons like the play, backwards and forwards has been made pointer by targeting the icon class.

Use of span tag in html to hold the song items: The <span> tag is an inline container used to mark up a part of a text, or a part of a document.

The <span> tag is easily styled by CSS or manipulated with JavaScript using the class or id attribute.

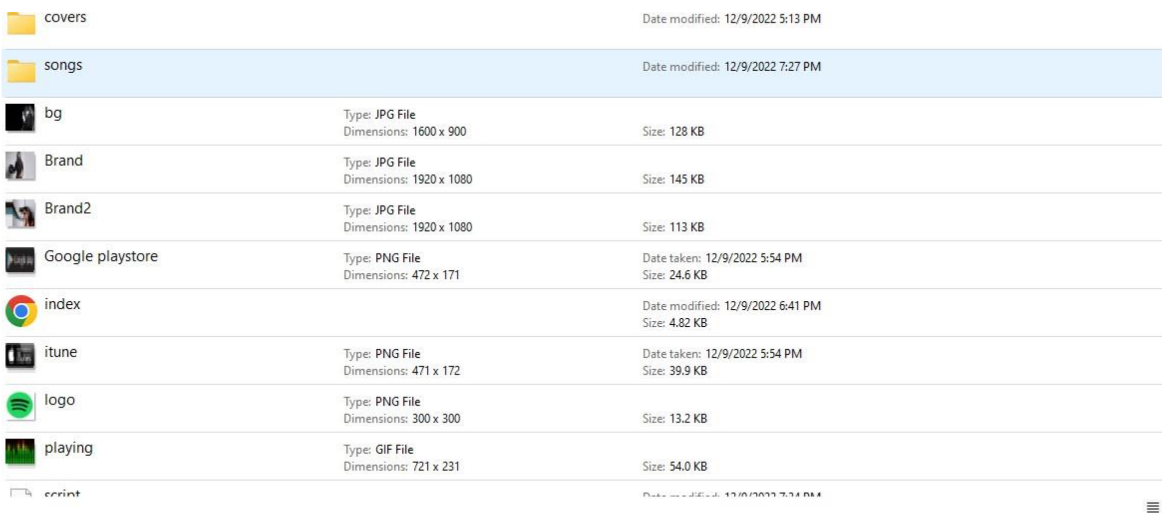
The <span> tag is much like the <div> element, but <div> is a block-level element and <span> is an inline element.

The bottom bar is made sticky. The ‘playing’ gif is executed only when a song is playing and it stops appearing when the song is stopped.

The seek bar has been named as myProgressBar and the cursor becomes a pointer when hovered on it, this is done to facilitate reaching a desired part of the song while playing.

Chapter 3- Implementation

Storage of the Web Application files:



covers		Date modified: 12/9/2022 5:13 PM
songs		Date modified: 12/9/2022 7:27 PM
bg	Type: JPG File Dimensions: 1600 x 900	Size: 128 KB
Brand	Type: JPG File Dimensions: 1920 x 1080	Size: 145 KB
Brand2	Type: JPG File Dimensions: 1920 x 1080	Size: 113 KB
Google playstore	Type: PNG File Dimensions: 472 x 171	Date taken: 12/9/2022 5:54 PM Size: 24.6 KB
index		Date modified: 12/9/2022 6:41 PM Size: 4.82 KB
itune	Type: PNG File Dimensions: 471 x 172	Date taken: 12/9/2022 5:54 PM Size: 39.9 KB
logo	Type: PNG File Dimensions: 300 x 300	Size: 13.2 KB
playing	Type: GIF File Dimensions: 721 x 231	Size: 54.0 KB
script		Date modified: 12/9/2022 7:34 PM

Fig 4. Folder for the storage of the Web Application files

- Covers contain the song item covers

- Songs folder contain the 30 .mp3 songs to e linked to the website

- index is the html file

- script is the JavaScript file

- style in the CSS file

- rest are background pictures and gif.

To play the songs on clicking the play button: use of JavaScript.

These variables are declared in the starting of JavaScript:

```
// Initialize the Variables

let songIndex = 0; // will set the initial song index to 0
let audioElement = new Audio('songs/1.mp3'); // will set the default initial song to be 1.mp3

let masterPlay = document.getElementById('masterPlay'); // masterplay is the id given to the play button, which when clicked will play the respective song

let myProgressBar = document.getElementById('myProgressBar');

let gif = document.getElementById('gif');

let masterSongName = document.getElementById('masterSongName');

let songItems = Array.from(document.getElementsByClassName('songItem')); // this araay is to store the 30 song items under 3 genres
```

This variable contains information about the songs like their name, location and cover:

```
let songs = [

   {songName: "Lahore - Guru Randhawa", filePath: "songs/1.mp3", coverPath: "covers/1.jpg"},

{songName: "Phulkari - Karan Randhawa", filePath: "songs/2.mp3", coverPath: "covers/2.jpg"},

   {songName: "Illegal Weapon - Garry Sandhu", filePath: "songs/3.mp3", coverPath: "covers/3.jpg"},

   {songName: "8 parche - Baani Sandhu", filePath: "songs/4.mp3", coverPath: "covers/4.jpg"},

   {songName: "Bapu Zimidar - Jassi Gill", filePath: "songs/5.mp3", coverPath: "covers/5.jpg"},

]
```

The 3 containers from the HTML file contain the song items which are uniquely identified by ‘id’ which ranges from 0 to 29 to exclusively identify the 30 song items.

song listing in the JavaScript includes their name, file path, cover image and the number of song items in the html file and the number of songs listed in the JavaScript ‘let songs’ list should be the same to avoid inconsistency.The song items in index.html file are given indexes from 0 to n in order to run the loop in the JavaScript file.

```
<div class="container">

      <div class="songList">

         <div class="songItemContainer">
```

```
        <p class="genre">Top 10 - Trending Punjabi Songs</p>
        <div class="songItem">
          <img alt="1">
          <span class="songName">Let me Love You</span>
          <span class="songlistplay"><span class="timestamp">03:17 </span><i id="0" class="far fa-2x songItemPlay fa-play-circle"></i></span>
        </div>
        <div class="songItem">
          <img alt="1">
          <span class="songName">Let me Love You</span>
          <span class="songlistplay"><span class="timestamp">04:25 </span><i id="1" class="far fa-2x songItemPlay fa-play-circle"></i></span>
        </div>
      </div>
    </div>
    <div class="songBanner"></div>
  </div>
```

This function includes the feature which plays the respective song

```
masterPlay.addEventListener('click', ()=>{

    if(audioElement.paused || audioElement.currentTime<=0){

        audioElement.play();

        masterPlay.classList.remove('fa-play-circle');

        masterPlay.classList.add('fa-pause-circle');

        gif.style.opacity = 1;

 }

    else{

        audioElement.pause();

        masterPlay.classList.remove('fa-pause-circle');

        masterPlay.classList.add('fa-play-circle');

        gif.style.opacity = 0;

    }
```

When the bigger play button is clicked and just after the song starts playing, the play button transitions into a pause button, which when clicked, pauses the song and the play button reappears at its previous position.

```
audioElement.play();
```

when the song starts playing, this happens:

```
masterPlay.classList.remove('fa-play-circle');

masterPlay.classList.add('fa-pause-circle');
```

the play button is removed and the pause button is added when the play button is clicked.

```
gif.style.opacity = 1;
```

it turns the opacity of the gif to 1 i.e. the gif appears when the song starts playing.

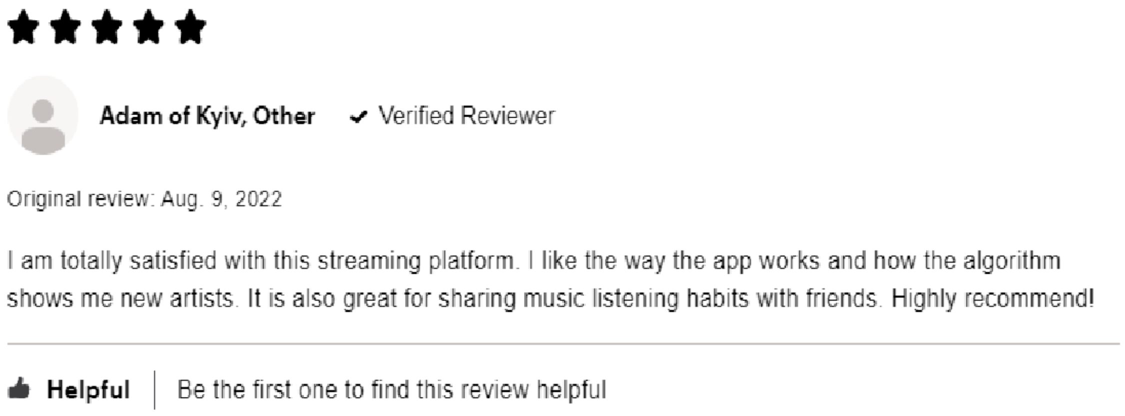
```
gif.style.opacity = 0;
```

it turns the opacity of the gif to 0 i.e. the gif disappears when the song stops playing.

The transition for the gif to appear on playing the song is lagged by adding transition in the CSS class:

```
.songInfo img{
    transition: opacity 0.4s ease-in;
    width: 145px;
    opacity: 0;
    transition: opacity 0.4s ease-in;
    height: 40px;
    margin-top: 20px;
}
```

CUSTOMER’S REVIEWS



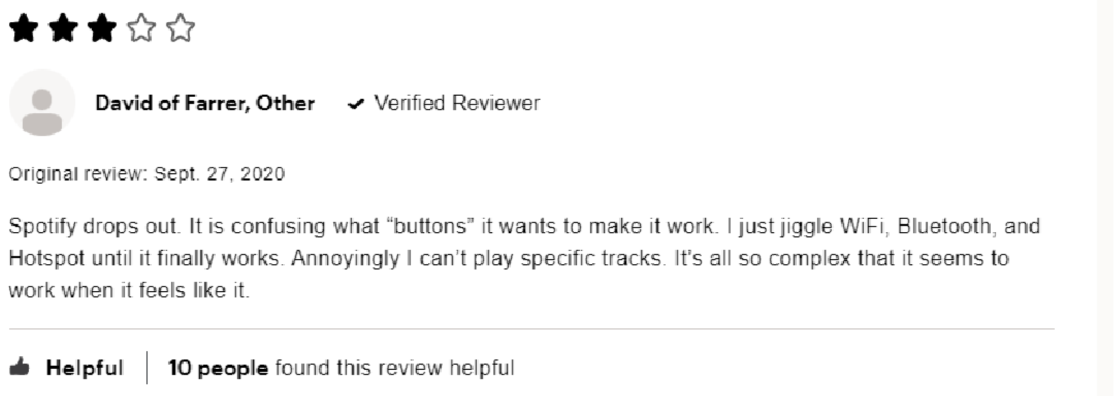
★★★★★

Adam of Kyiv, Other ✓ Verified Reviewer

Original review: Aug. 9, 2022

I am totally satisfied with this streaming platform. I like the way the app works and how the algorithm shows me new artists. It is also great for sharing music listening habits with friends. Highly recommend!

👍 Helpful | Be the first one to find this review helpful



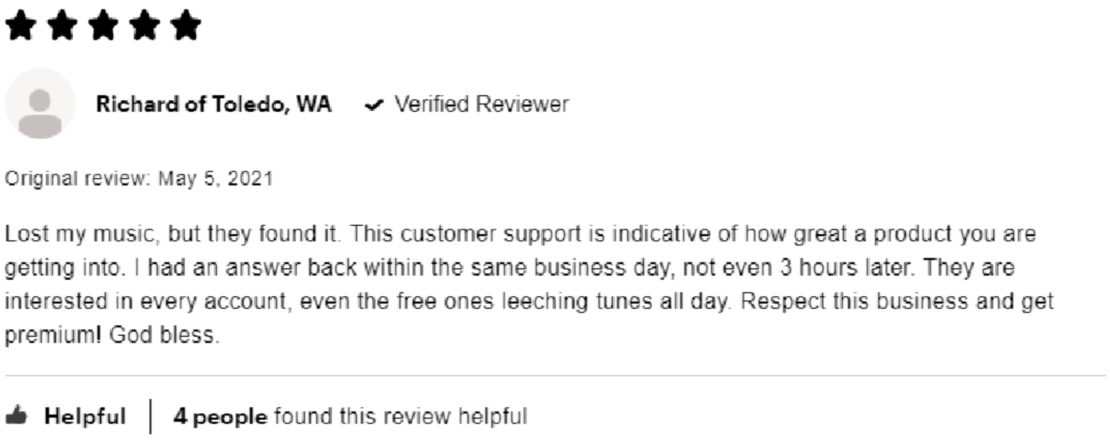
★★★☆☆

David of Farrer, Other ✓ Verified Reviewer

Original review: Sept. 27, 2020

Spotify drops out. It is confusing what “buttons” it wants to make it work. I just jiggle WiFi, Bluetooth, and Hotspot until it finally works. Annoyingly I can’t play specific tracks. It’s all so complex that it seems to work when it feels like it.

👍 Helpful | 10 people found this review helpful



★★★★★

Richard of Toledo, WA ✓ Verified Reviewer

Original review: May 5, 2021

Lost my music, but they found it. This customer support is indicative of how great a product you are getting into. I had an answer back within the same business day, not even 3 hours later. They are interested in every account, even the free ones leeching tunes all day. Respect this business and get premium! God bless.

👍 Helpful | 4 people found this review helpful

Chapter 4-Result & Discussion

Progress bar:

The initial value is set to 0 and the slider moves with the progress of the song and the slider can be dragged to the position in the progress bar where we have to reach/seek in the song. The progress of a song is interpreted as 100 percent and the slider moves in proportion to it.

```
// Listen to Events

audioElement.addEventListener('timeupdate', ()=>{

    // Update Seekbar

    progress = parseInt((audioElement.currentTime/audioElement.duration)* 100);
    myProgressBar.value = progress;

})

myProgressBar.addEventListener('change', ()=>{

    audioElement.currentTime = myProgressBar.value * audioElement.duration/100;

})
```

Here, a formula for current time in the seek bar:

```
audioElement.currentTime = myProgressBar.value * audioElement.duration/100;
```

Current time = progress value (in percent) * total duration of the song being

played/100

The gif will appear when the song is playing and will disappear as soon as the song stops playing.

```
// Handle play/pause click

masterPlay.addEventListener('click', ()=>{
    if(audioElement.paused ||

        audioElement.currentTime<=0){

        audioElement.play();

        masterPlay.classList.remove('fa-play-

        circle');masterPlay.classList.add('fa-

        pause-circle'); gif.style.opacity = 1;

    }

    else{

        audioElement.pause();

        masterPlay.classList.remove('fa-pause-

        circle');

        masterPlay.classList.add('fa-play-circle');
        gif.style.opacity = 0;

    }

})

gif.style.opacity = 1;

gif.style.opacity = 0;
```

The gif will appear only when the song is playing as its opacity is set to 1

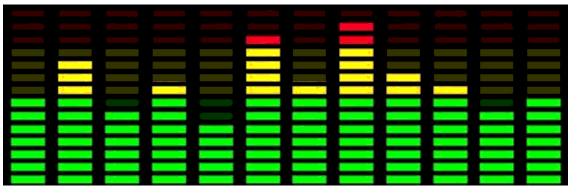


Fig. 4

The gif will disappear when the song is stopped as its opacity is set to 0

The functioning of the next and previous buttons is set by if-else statement in JavaScript and by settingconstraint on the ‘id’ of the songs.

The initial song index is set as 0. For next, the index is increased by 1 in every step.

```
document.getElementById('next').addEventListener('click', ()=>{
   if(songIndex>=30){
       songIndex = 0
   }
   else{
       songIndex += 1;
   }
   audioElement.src = `songs/${songIndex+1}.mp3`;
   masterSongName.innerText = songs[songIndex].songName;
   audioElement.currentTime = 0;
   audioElement.play();
   masterPlay.classList.remove('fa-play-circle');
   masterPlay.classList.add('fa-pause-circle')
})
```

The initial song index is set as 0. For previous, the index is decreased by 1 in every step.

```
document.getElementById('previous').addEventListener('click', ()=>{
   if(songIndex<=0){
       songIndex = 0
   }
```

```
  else{
    songIndex -= 1;
  }
  audioElement.src = `songs/${songIndex+1}.mp3`;
  masterSongName.innerText = songs[songIndex].songName;
  audioElement.currentTime = 0;
  audioElement.play();
  masterPlay.classList.remove('fa-play-circle');    masterPlay.classList.add('fa-pause-circle');
})
```

Here,

```
  masterSongName.innerText = songs[songIndex].songName;
```

, will change the name of the song in the bottom bar as it is navigated by using the previous and next button.The name corresponding to the respective song index will be displayed in the bottom bar, to the right of gif.

Chapter 5-Future Scope & Conclusion

Market Statistics of Music Streaming Platforms & Apps:

Spotify is the world’s biggest music streaming platform by number of subscribers. Users of service simply need to register to have access to one of the largest collections of music in the history, plus podcasts and other audio content. It operates on a freemium model. Free Spotify access comes with lower sound quality,advertisements and requires an internet connection. Those who pay for Spotify Premium can listen uninterrupted to high-quality recordings and are able to download songs for offline listening. Spotify was founded in 2006 in Stockholm, in Sweden, by Daniel Ek and Martin Lorentzon. The two wanted to create a legal digital music platform to respond togrowing challenge of online music piracy in the early 2000s.

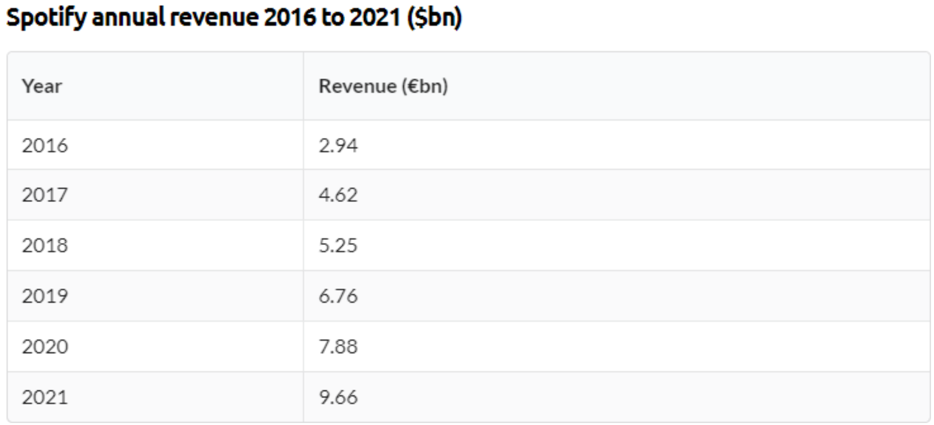
Eventually convincing record labels to agree to share content in return for an aggregate of 20 percent stake, Spotify was launched in 2008. It was an instant success, with the Facebook partnership helping it rise rapidly to prominence. Surviving the transition to mobile, Spotify went public in April 2018, with a market cap of $26.5 billion after the firstday of trading.

By the mid - 2000s, Pandora and Spotify had both launched, which both tried to reshape the cost of music. Instead of paying for the song, Pandora and Spotify would pay a much smaller amount per stream to the record company. This fundamentally changed how music is valued. Instead of sales, the emphasis was on repeated listens and getting a song in a popular playlist or radio station. The music industry shunned the new platforms at the start, with several artists blocking Spotify and Pandora from using their material. YouTube also got its start in 2005 and by 2010 it was the most popular video service in the world, overtaking MTV and other channels as the premier place to watch music videos. Major music labels launched VEVO in 2009 as a way to control the distribution and revenue share of music videos on YouTube.

Even with the poor reception from the music industry, YouTube, Pandora and Spotify continued to grow in users. But it wouldn't be until the mid-2010s that music streaming became the dominant revenue generator for music labels, surpassing physical and digital.

Spotify key statistics

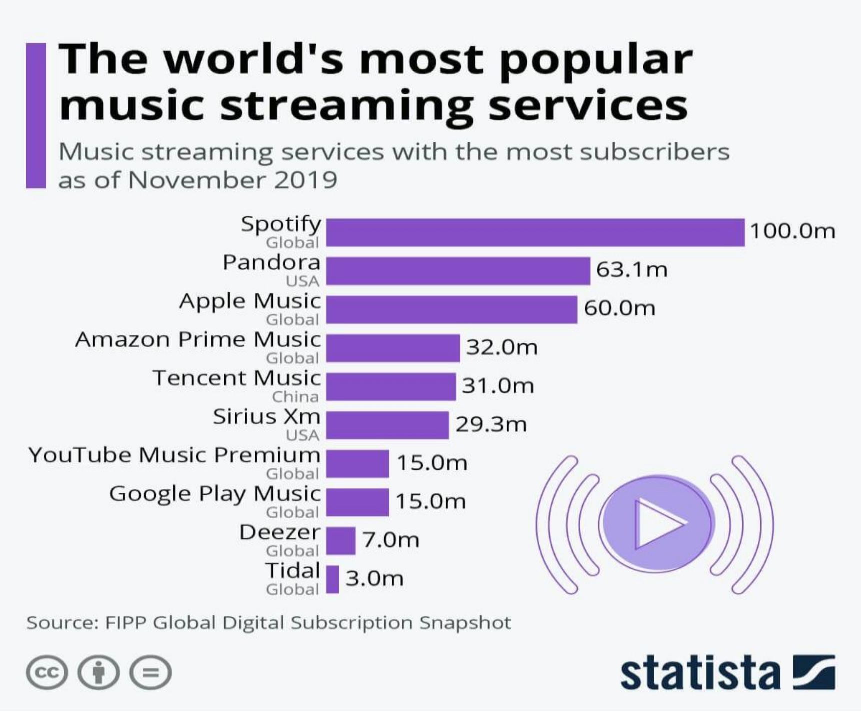
Spotify generated €9.66 billion revenue in 2021, a 22% increase year-on-year Spotify has never published an net profit. In 2021, it posted a €39 million loss 422 million people use Spotify once a month, 182 million are subscribers
70 million songs are available on Spotify and 2.9 million podcasts



Spotify annual revenue 2016 to 2021 ($bn)

Year	Revenue (€bn)
2016	2.94
2017	4.62
2018	5.25
2019	6.76
2020	7.88
2021	9.66

If the recent stats have anything to go by, from the year 2014-2020, the music apps revenue has crossed 500%. Not just this, but since the year 2019, the growth in terms of revenue is expected more than 100 billion USD on a year-to-year basis going up to the year 2023. This constant increase in the number of app users ensures recurring revenue for the music app business.



The world's most popular music streaming services

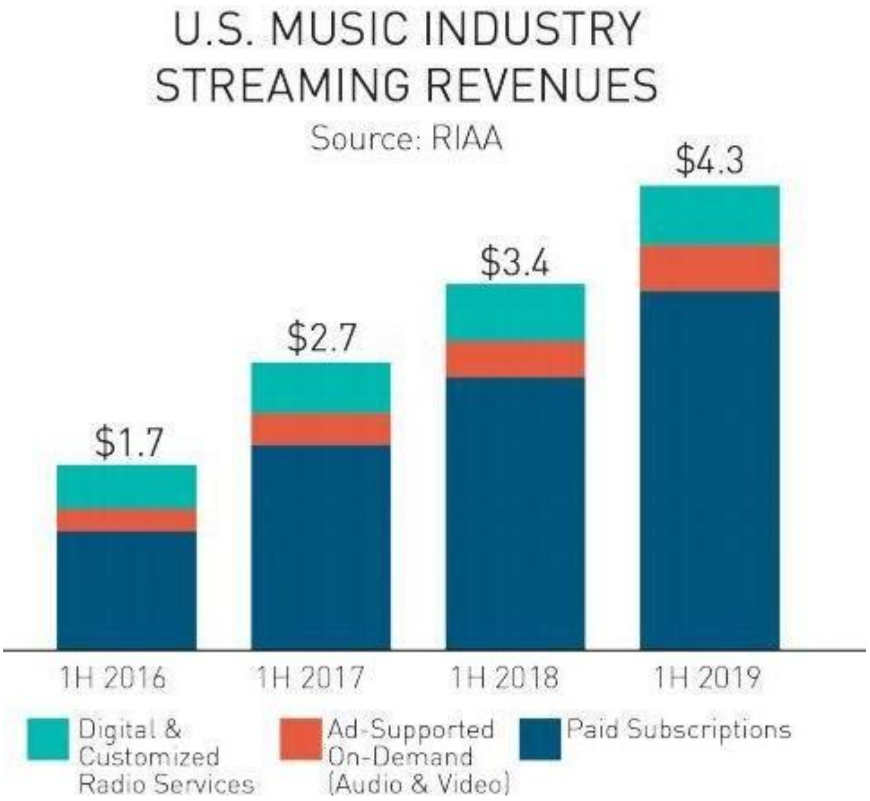
Music streaming services with the most subscribers as of November 2019

Source: FIPP Global Digital Subscription Snapshot

The world's most popular music streaming services

How You Can Benefit By Creating A Music Streaming application?

As technology is taking onto the world, the music lovers' charm for the music stream apps has evolved as well. These days we hardly see people collect cassettes, records of their favorite music artists as they prefer to rather listen to it on their phones which they are carrying with them 24×7.

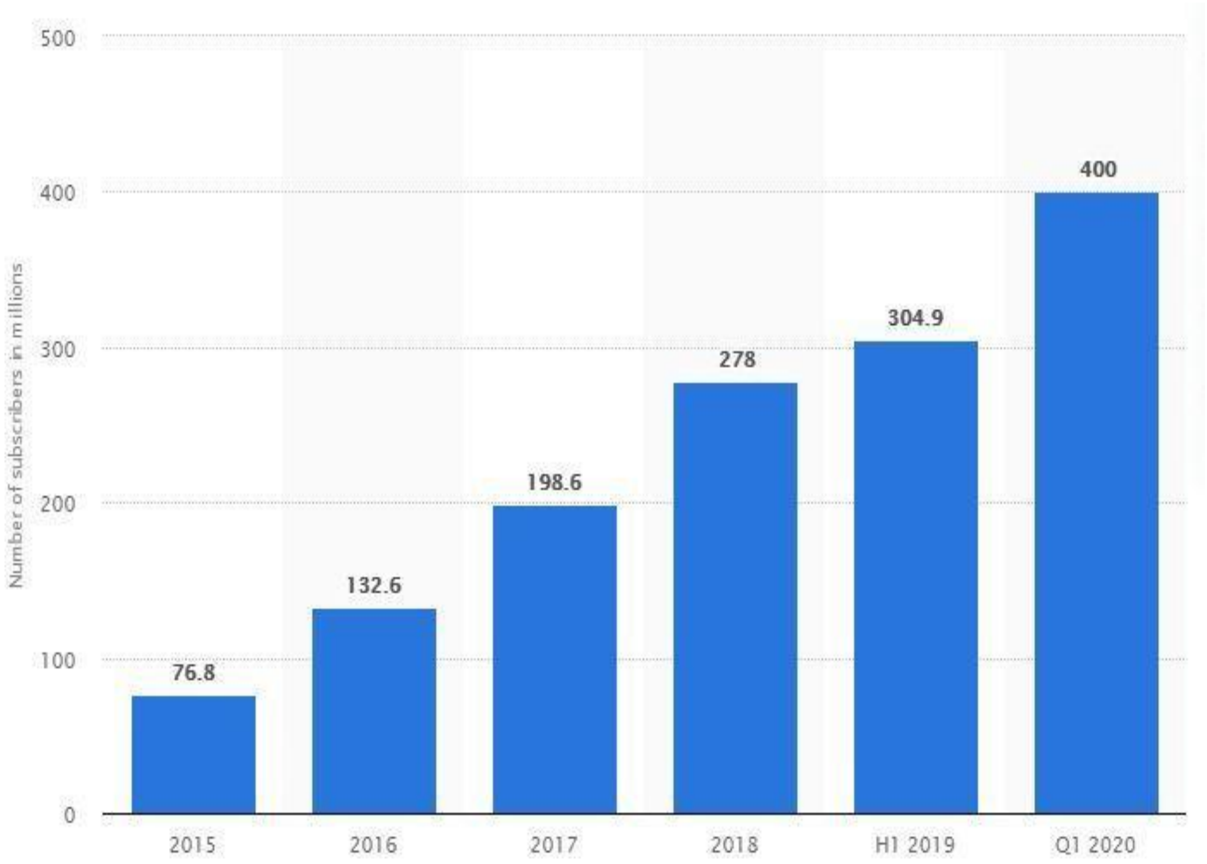


U.S. music industry streaming revenues

Not just the convenience, but these apps also offer abundant options to music lovers satisfying their tastes and preferences in music.

Also, if you are looking to enter into the music streaming app business, then know that the opportunity here is massive and you can easily start your own music streaming service. The demand for digital music is here to stay with more and more people showing interest in it every day.

Number of Music Streaming App Users Worldwide 2015 to 2020



Number of subscribers with year

The online music streaming market across the world is expected to rise to more than $10 billion by 2023. With the rise in technology in past few years, music streaming apps have been among the most sought after apps and here it not just offers on-demand music but allows to captivate the audience via music reviews, movie reviews, podcasts, audio interviews, motivational lectures, and a lot more, hence increasing customer retention. In the music apps, the users get a synchronized music library as per users' likes. With the use of technologies like Artificial Intelligence and Big Data, the app suggests favorite tracks and music to the users.

Also, the easy access to the internet across the world has further increased the consumption and demand for online music. Hence, the time is just right to enter the market and reap the profits.

Monetization Opportunities for Music Streaming Services & Apps

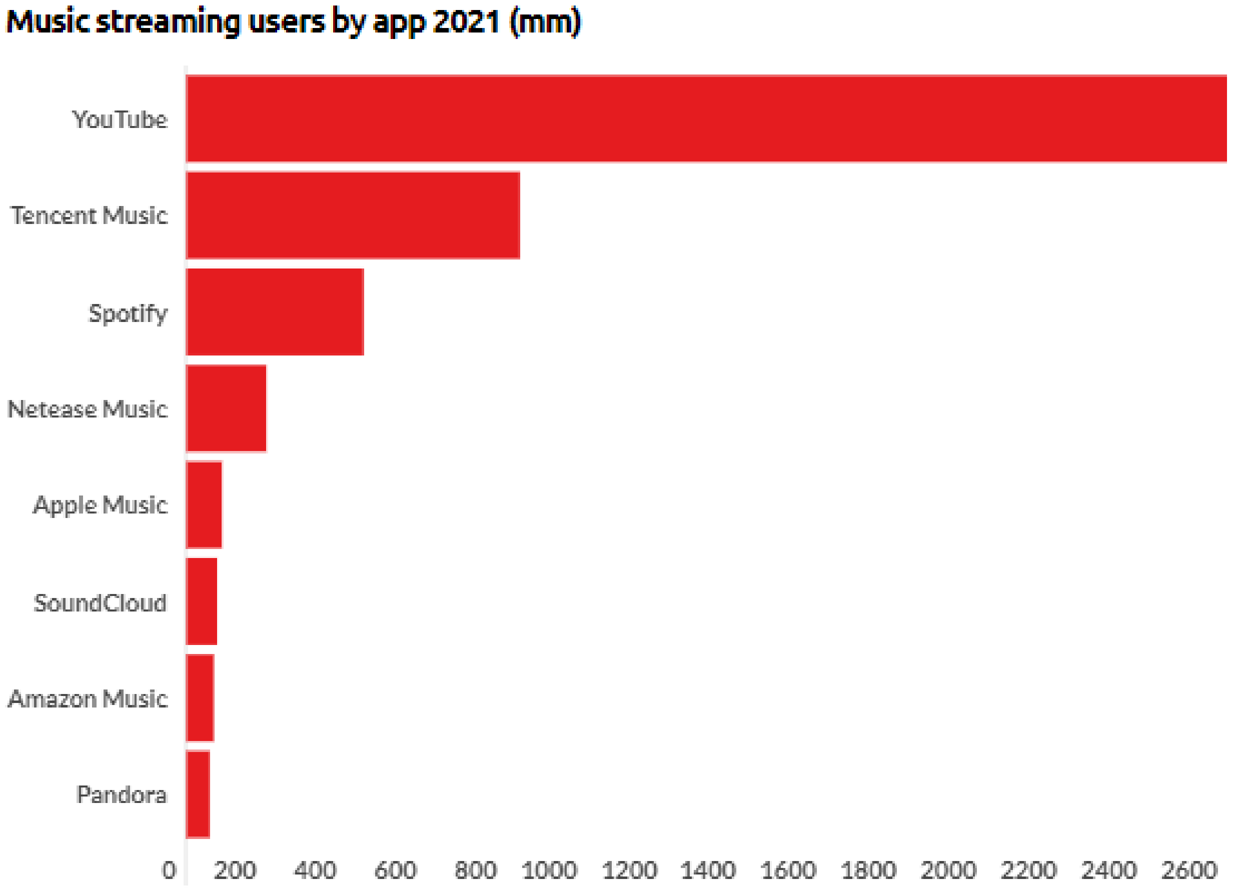
There are many monetization opportunities available even if the app developed by you is available for free. However, mainly there are two major strategies to monetize music streaming app, i.e.:

- Subscription
- Advertisements

These are also the two revenue models adopted by most of the music streaming apps. The app offers free & premium app versions to the users. According to their listening preferences, the users can choose the version that suits them the best. The music or audio streaming apps are one of the most prominent in the app stores, thus they provide higher user retention rates and this attracts a lot of advertisers.

Music streaming users by app:-

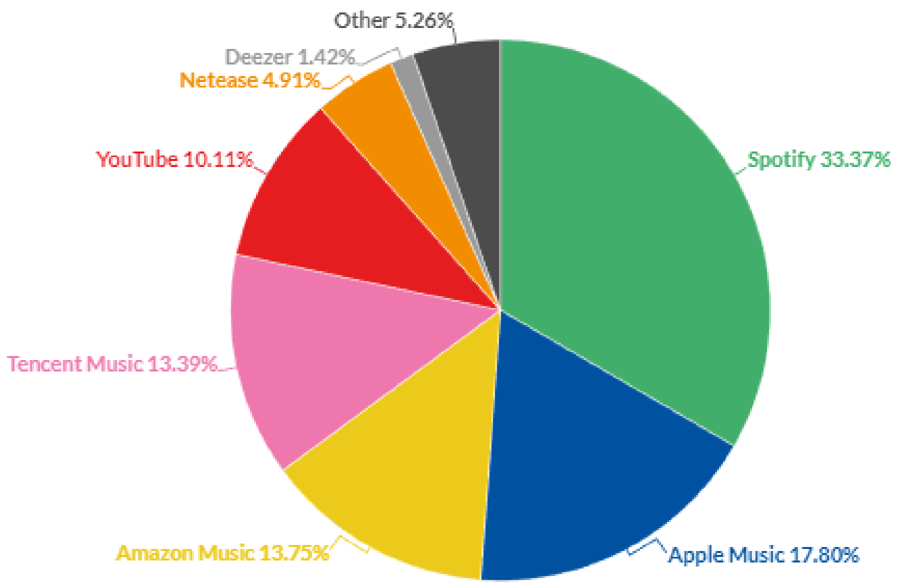
In terms of usage, YouTube is far ahead of other platforms, with two billion active users who listen to music. Tencent's various music platforms in China have over 800 million users, while Spotify free and paid users total 381 million.



Music streaming Market share:-

Spotify is the platform for about a third of all music streaming listeners worldwide, with Apple Music in second place. Tencent and Netease are the two key platforms in China, despite Spotify and Apple Music not being blocked by the Chinese government.

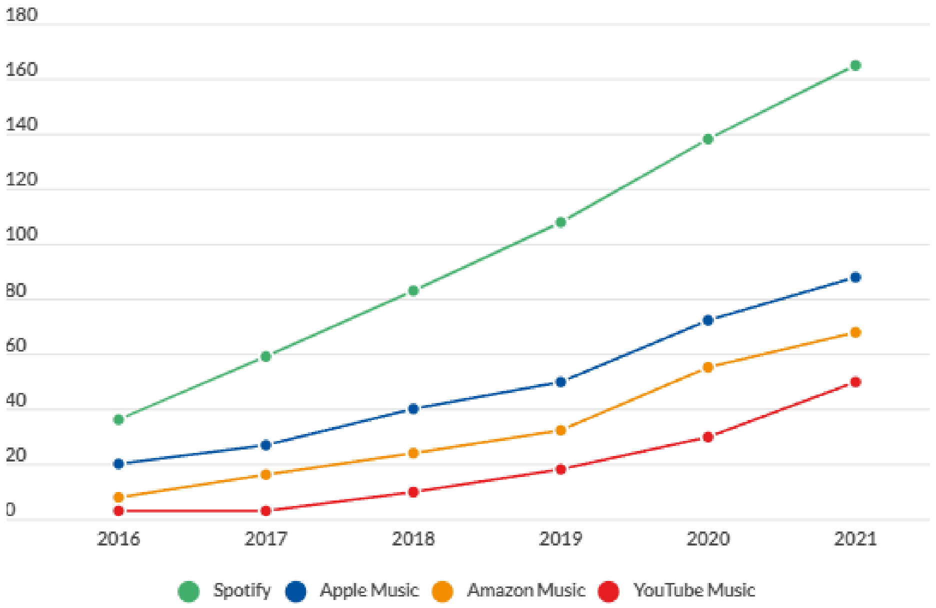
Global music streaming marketshare by subscribers 2021 (%)



Music streaming subscribers by app

As one of the first music streaming platforms, Spotify had over 50 million subscribers before Apple had launched its own platform. It has kept this lead, although market share has declined as more services have launched.

Music streaming subscribers by app 2016 to 2021 (mm)



References

- ‘fontawesome.com’ is used for the amazing buttons that are used for play, pause, forward and backward in the web application.
- Google font is used to modify the font used in the CSS section.
- Web Development Course by University of California, Davis is used to learn HTML, CSS AND JavaScript basics and applications.
- The songs used are downloaded from ganna.com
- Youtube: CodeWithHarry
- Youtube : Clever Programming
- W3School Website
- Youtube : Programming With Mosh