WEB TECHNOLOGY

MUSIC APP (Apple Music)

V.GOKUL SAI TEJA(AP19110010156)
M.HARITHA(AP19110010270)
P.CHARAN SAI(AP19110010038)

Acknowledgment:

Our interest in Music has always played a major role in this project.

We sincerely thanking Prof. Swamy Ayyappa (web technology professor) who taught us how to prepare a perfect web page using HTML, CSS, and JAVASCRIPT.

I pay my deep sense of gratitude to T. Ragunathan (HOD) of the CSE Department for giving this opportunity to present our skills gained from courses in this semester.

We feel to acknowledge our indebtedness and a deep sense of gratitude to our Mentor Sanjeev kumar who cleared our doubts (By mentoring sessions) and by providing the updates of the project accordingly. Lastly, we, the team members, have made an amazing effort by interacting with each other and solving the problems collectively.

Abstract:

The project that we've taken is iTunes Web App and this was one of the Music clone application which was developed by using iTunes API.

API (Application programming interface) is the set of programming code that enables data transmission between one software product and another.

Nowadays API's used widely across the internet to maintain and design many web applications in a very simpler way and using API's makes the website faster and the data must be requested to the website pathway(Database) and the responses will come as JSON(JavaScript Object Notation), Now by using this JSON response we need to render the whole HTML(Hypertext Markup Language) and CSS(Cascading Style Sheets) of the webpage accordingly.

Instead of using Hot-Reloading to render the site, using the APIs make the site load faster. Thus, we are developing this project using APIs.

Now in this project, we are requesting the data to the iTunes Databases based on the search Event and the iTunes APIs are maintained by Apple Inc. So the iTunes will send the response to the web browser and the result needed to be parsed into the JSON format and JavaScript will organize and design the HTML according to the JSON's data and later on, we can style the HTML by using the CSS as per our will.

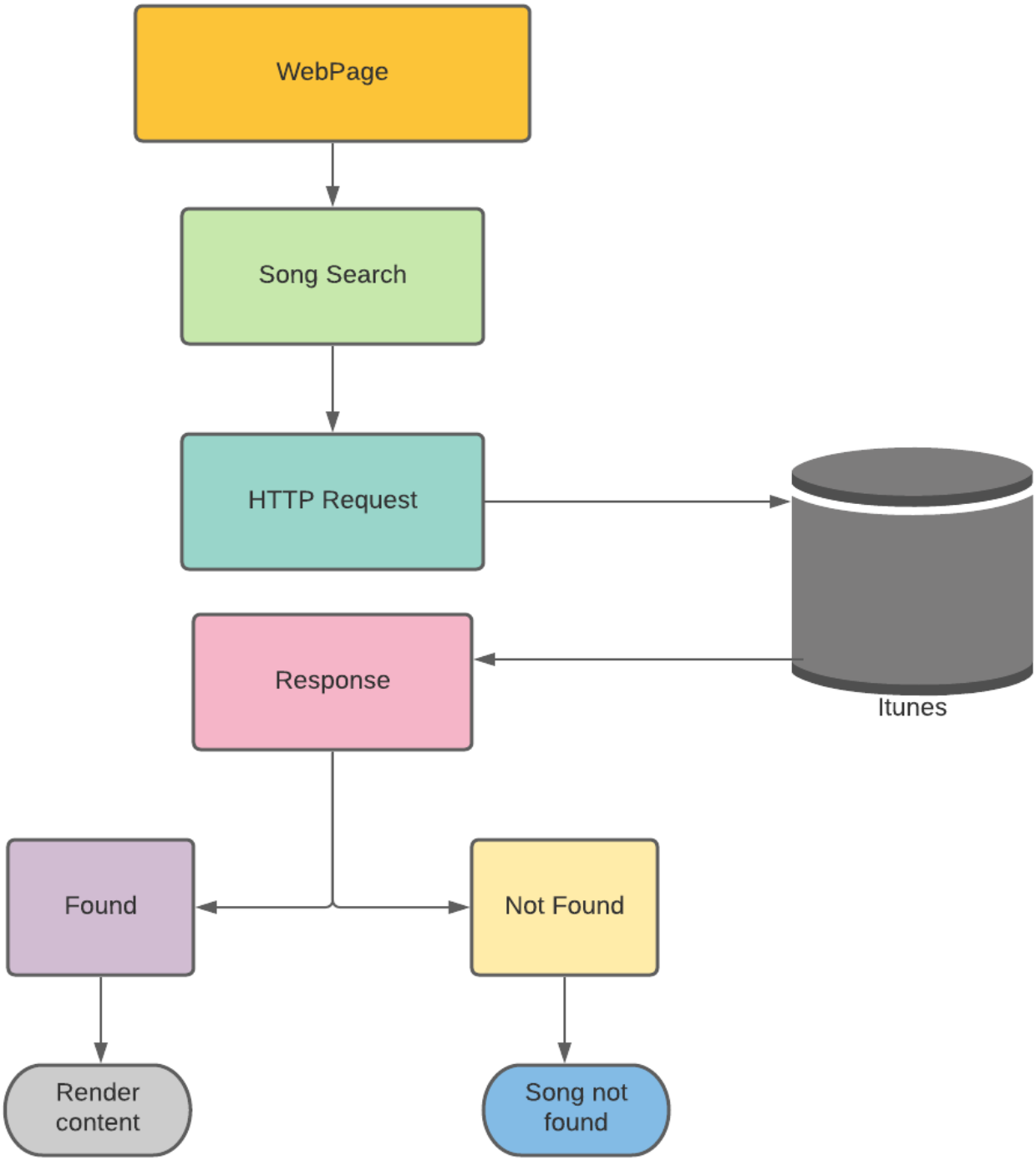
Now the Response from the iTunes API contains these data fields song name, track name, artist name, collection name, collection censored name, album cover, song cover, a preview of the track, audio files … so on. Now by using these files we can design the webpage using the flex displaying methods and making the site very responsive for different screen sizes.

Functional Specifications:

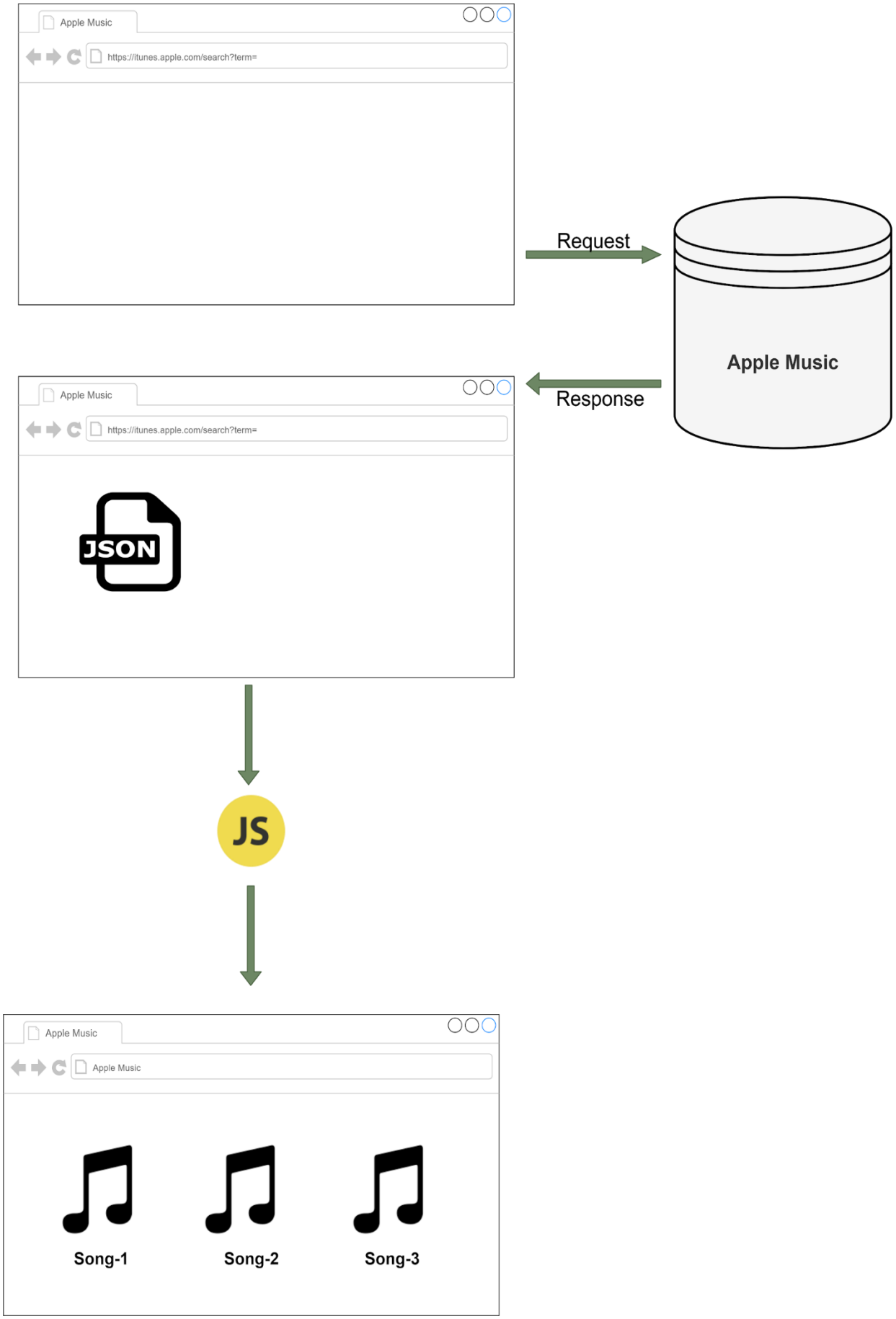
The iTunes music clone app consists of a single webpage which eventually gets triggered by the search events and the input that we provided to the search bar in the HTML form will get secreted by the JavaScript and then the Data will get processed and create the GET request to the server (iTunes Server).

- **Process under the Hood:** when the data was received as the response from the server then the data will get ready to render to the webpage thus that is the reason for rendering the web page without any Hot reload.
- **Data that will get rendered:** Out of the many items inside the response object, we just render the artist name, preview URL, track name, Collection name and album cover picture.
- **What if the searched data doesn't exists?**
  If the length of the response object was zero then that is the case where the searched data doesn't exist inside the database of iTunes.

Flow chart:


WebPage
Song Search
HTTP Request
Itunes
Response
Found
Not Found
Render content
Song not found

Browser's Interpretations:



Source Code:

HTML:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <link rel="stylesheet" href="style.css" />
    <title>iTunes App</title>
    <link rel="icon" type="image/png" href="./icons8-itunes-
64.png">
  </head>
  <body>
    <header>
      <form id="form">
        <input type="text" id="search" class="search"
placeholder="Search" />
      </form>
    </header>
    <div id="info">
    </div>
    <main id="main">
    </main>

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

CSS:

```css
:root {
  --primary-color: #ff7a00;
  --secondary-color: #010101 ;
}

* {
  box-sizing: border-box;
}

body {
  background-color: var(--primary-color);
  font-family: "Poppins", sans-serif;
  margin: 0;
}

header {
  padding: 1rem;
  display: flex;
  justify-content: flex-end;
  background-color: var(--secondary-color);
}

.search {
  background-color: transparent;
  border: 2px solid var(--primary-color);
  border-radius: 50px;
  font-family: inherit;
  font-size: 1rem;
  padding: 0.5rem 1rem;
  color: #000;
}

.search::placeholder {
  color: #ff7a00;
}
```

```css
.search:focus {
  outline: none;
  background-color: var(--primary-color);
}

main {
  display: flex;
  flex-wrap: wrap;
  justify-content: center;
}

.Song {
  width: 300px;
  margin: 1rem;
  background-color: var(--secondary-color);
  box-shadow: 0 4px 5px rgba(0, 0, 0, 0.2);
  position: relative;
  overflow: hidden;
  border-radius: 10px;
}

.Song img {
  width: 100%;
}

.Song-info {
  color: #eee;
  display: flex;
  align-items: center;
  justify-content: space-between;
  padding: 0.5rem 1rem 0rem;
  letter-spacing: 0.5px;
  margin-bottom: 0;
}
.Song-info1 {
  color: rgba(224, 140, 168, 0.828);
  display: flex;
```

```
  align-items: center;
  justify-content: space-between;
  padding: 0.2rem 1rem 0rem;
  margin-top: 0;
  letter-spacing: 0.5px;
}

.Song-info h3 {
  margin-top: 0;
}

.Song-info span {
  background-color: var(--primary-color);
  padding: 0.25rem 0.5rem;
  border-radius: 3px;
  font-weight: bold;
}

.Song-info span.green {
  color: lightgreen;
}

.Song-info span.orange {
  color: orange;
}

.Song-info span.red {
  color: red;
}

.overview {
  background-color: #fff;
  padding: 2rem;
  position: absolute;
  left: 0;
  bottom: 0;
  right: 0;
  max-height: 100%;
```

```
  transform: translateY(101%);
  overflow-y: auto;
  transition: transform 0.3s ease-in;
}

.Song:hover .overview {
  transform: translateY(0);
}
```

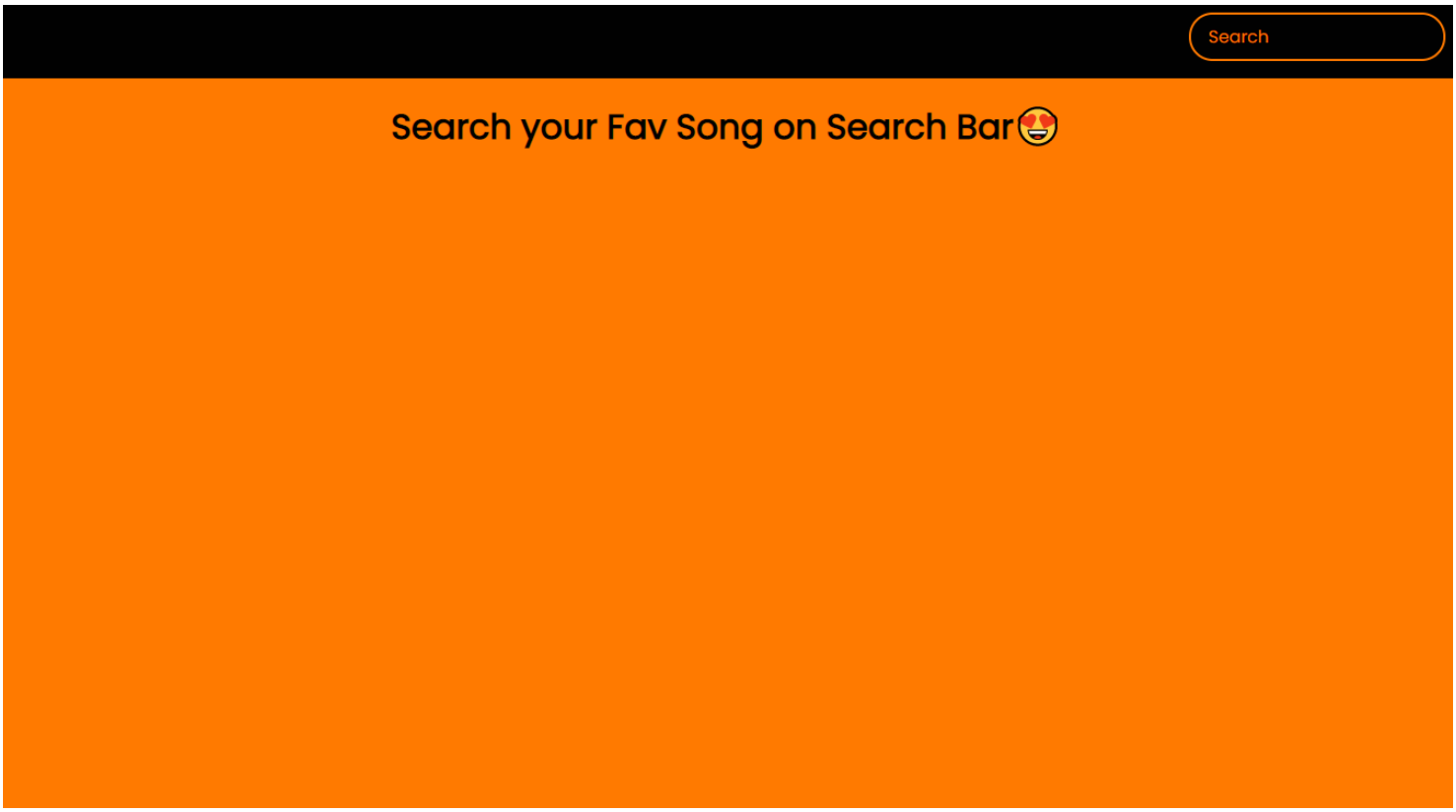
JavaScript:

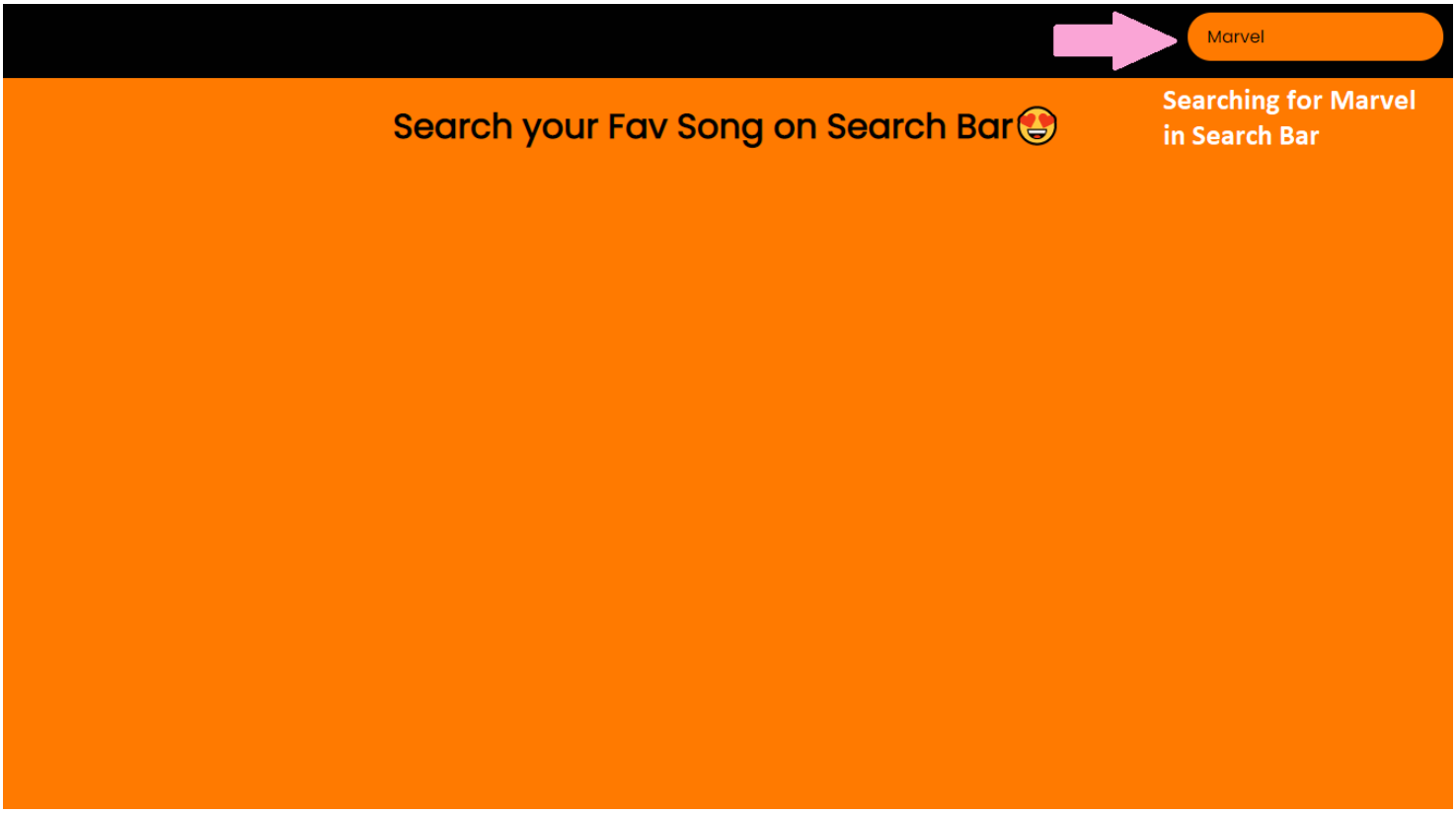
```
const SEARCH_API = "https://itunes.apple.com/search?term=";

const imp = document.getElementById("main");
const form = document.getElementById("form");
const search = document.getElementById("search");


function greet() {
  info.innerHTML = "<center><h1>Search your Fav Song on
Search Bar😍</h1></center>";
}


info.addEventListener("load", greet());

function getSongs(url) {
  fetch(url)
    .then((x) => {
      x.json().then((data) => {
        console.log(data.results);
        showSongs(data.results);
      });
    })
    .catch((err) => {
     console.log("...")
    });
```

```
}

function showSongs(Songs) {
  imp.innerHTML = "";
  if (Songs.length === 0) {
    main.innerHTML = `<h1 style="font-weight: 5rem" >Sorry
Song Not Found!😑</h1>`;
  }
  Songs.forEach((Song) => {
    var  {
      artworkUrl100,
      artistName,
      previewUrl,
      trackName,
      collectionName,
      collectionCensoredName
    } = Song;
    if(trackName===undefined){
      return;
    }

    const SongEl = document.createElement("div");
    SongEl.classList.add("Song");

    SongEl.innerHTML = `
    <img
    src="${pictureClarity(artworkUrl100)}"
    alt="Pic"
  />
  <div class="Song-info">
  <h4>${trackName}</h4>
  </div>
  <div class="Song-info1">
  <h5>${trim(artistName)}</h5>
  </div>
  <div class="Overview">
    <audio controls>
      <source
```

```
        src=${previewUrl}
        type="audio/ogg"
      />
    </audio>
  </div>
        `;
    imp.appendChild(SongEl);
  });
}

function trim(x) {
  if (x.length > 35) {
    var y = "";
    for (var i = 0; i < 35; i++) {
      y = y + x[i];
    }
    y + "...";
    return y;
  } else {
    return x;
  }
}
function pictureClarity(link) {
  var index = link.replace("100x100bb", "500x500bb");
  return index;
}
form.addEventListener("submit", (e) => {
  info.innerHTML=""
  e.preventDefault();
  const searchTerm = search.value;
  if (searchTerm && searchTerm !== "") {
    getSongs(SEARCH_API + searchTerm);
    search.value = "";
  } else {
    window.location.reload();
  }
});
```

Sample Cases:





Captain Marvel

Anna Boden & Ryan Fleck

0:00 / 2:18

Avengers: Infinity War

Anthony Russo & Joe Russo

0:00 / 1:47

Guardians of the Galaxy

James Gunn

0:00 / 2:24

Doctor Strange (2016)

Scott Derrickson

0:00 / 2:23

Demon Slayer - Kimetsu no Yaiba the Movie: Mugen Train

Haruo Sotozaki

0:00 / 0:00

Dragon Ball Super: Broly

Tatsuya Nagamine

0:00 / 0:00

The Last: Naruto the Movie

Tsuneo Kobayashi

0:00 / 0:00

Dragon Ball Z: Resurrection F

Unknown

0:00 / 0:00

Pokémon: The First Movie (Dubbed)

Kunihiko Yuyama

0:00 / 0:00

Future Scope: By using the same iTunes API we can make the videos preview too by which we can make something similar to YouTube and moreover by using Async JS we can further make the change beyond and change the loading interface of the webpage.

Similarly by using the other music platform API's we can even make the user experience very native and further development can take place in this project to make user experience very better.

Who can use this project ?

This can be used by the people who are really interested to listen to the music.

Conclusion:

we can conclude that this project purely rely on the basic web technologies such as HTML,CSS and JavaScript. Moreover developing this project brought more interest upon web development and this web application helps the user to listen the songs.

References:

1. Bootstrap: https://getbootstrap.com/
2. MDN Documentation: https://developer.mozilla.org/en-US/

Deployment: https://WebTechProj.gocool321.repl.co