

Kanpur Institute of Technology, Kanpur

Dr. A.P.J Abdul Kalam Technical University,

Lucknow

\*“Web Music Player ”

**A Project Report Submitted in Partial Fulfillment of the Requirements for the Degree**

**of**

## Bachelor of Technology

**in**

## Computer Science & Engineering

**By**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Student Name** | **Roll Number** | **Branch and Section** |
| 1 | Adarsh Kashyap | 2301650100014 | CSE, A |
| 2 | Alok Vishwakarma | 2301650100022 | CSE, A |
| 3 | Adarsh Chaurasiya | 2301650100012 | CSE, A |
| 4 | Anmol yadav | 2301650100039 | CSE,A |
| 5 | Devansh Shukla | 2301650100066 | CSE,A |

**Under the Supervision of**

**Mr. Anand Sharma**

**(Assistant Professor)**

# DECLARATION

*We hereby declare that this submission of our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma or the university or other institute of higher learning, except where due acknowledgement has been made in the text.*

|  |  |
| --- | --- |
| ***Signature:***  ***Name: Adarsh Kashyap***  ***Roll no:***  ***2301650100014***  ***Date:*** | ***Signature:***  ***Name: Alok Vishwakarma***  ***Roll no: 2301650100022***  ***Date:*** |
| ***Signature:***  ***Name: Adarsh Chaurasia***  ***Roll no: 2301650100012***  ***Date:*** | ***Signature:***  ***Name: Anmol Yadav***  ***Roll no: 2301650100039***  ***Date:*** |
| ***Signature:***  ***Name: Deavash Shukla***  ***Roll no: 2301650100066***  ***Date:*** |  |

# CERTIFICATE

*This is to certify that the Project Report entitled* ***“Image and Audio Steganography with Suspicious Message Detection”*** *which is submitted by* ***AAAA (11111)****,* ***BBBB (22222)****,* ***CCCC (33333), DDDDD (4444444), EEEEE (5555555)*** *of the 5th semester, in the year 2022-2023.*

*In partial fulfillment of the requirement for the award of degree of B.Tech in department of* ***Computer Science & Engineering*** *of* ***Dr. APJ Abdul Kalam Technical University*** *is a record of the candidate own work carried out by him/her under my supervision. The matter embodied in this thesis is original and is not been submitted for the award of any degree.*

|  |  |
| --- | --- |
| *Signature* | *Signature* |
| *Mr. Rahul Singh)* | *(Mr. Anand Sharma)* |
| *(Head of Department)* | *(Assistant Professor)* |
| *CSE Department*  *KIT Kanpur* | *CSE Department*  *KIT Kanpur* |

*Date:*

# ACKNOWLEDGEMENT

*It gives us a great sense of pleasure to present the report of the B.Tech Project undertaken during* ***B.Tech Final/Third/Second Year****. We owe special debt of gratitude to our guide* ***Mr. XYZ*** *of Department of* ***Computer Science & Engineering, Kanpur Institute of Technology, Kanpur (U.P)*** *for his constant support and guidance throughout course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have been light of the day.*

*We also take the opportunity to acknowledge the contribution of* ***Mr. Rahul Singh , Head of Department of Computer Science & Engineering, Kanpur Institute of Technology, Kanpur (U.P.)*** *for his full support and assistance during the development of the project.*

*We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their assistance and cooperation during the development of my project. Last but not the least, we acknowledge our friends for their contributions in the completion of the project.*

|  |  |
| --- | --- |
| ***Signature:***  ***Name: Adarsh Kashyap***  ***Roll no:***  ***2301650100014***  ***Date:*** | ***Signature:***  ***Name: Alok Vishwakarma***  ***Roll no: 2301650100022***  ***Date:*** |
| ***Signature:***  ***Name: Adarsh Chaurasia***  ***Roll no: 2301650100012***  ***Date:*** | ***Signature:***  ***Name: Anmol Yadav***  ***Roll no: 2301650100039***  ***Date:*** |
| ***Signature:***  ***Name: Deavash Shukla***  ***Roll no: 2301650100066***  ***Date:*** |  |
| ***Signature:***  ***Name: AAAAAAA Roll no: 11111111***  ***Date:*** | ***Signature:***  ***Name: BBBBBBBBBB Roll no: 22222222***  ***Date:*** |
| ***Signature:***  ***Name: CCCCCCC Roll no: 3333333***  ***Date:*** | ***Signature:***  ***Name: DDDDDDDD Roll no: 44444444***  ***Date:*** |
| ***Signature:***  ***Name: EEEEEEE Roll no: 555555555***  ***Date:*** |  |

# ABSTRACT

*Saavn is a modern, web-based music player designed to offer a premium, ad-free audio experience. It caters to music enthusiasts with high-quality streaming support of up to 320kbps, ensuring crystal-clear sound. The platform features a sleek, user-friendly interface optimized for seamless operation on both desktop and mobile devices, providing a hassle-free and enjoyable music experience.*

*Saavn simplifies music enjoyment with its ability to directly download MP3 files along with complete metadata, making music management convenient for users. It boasts a large and diverse library of songs fetched dynamically from reliable sources, ensuring an expansive catalog to suit every taste.*

*As an open-source project, Saavn prioritizes performance, accessibility, and community contribution. Its code is optimized for speed and reliability, allowing users to deploy the application easily on a variety of hosting services like Vercel, Heroku, GitHub Pages, or any platform capable of serving static files. No complex installations are needed—simply open the page in a browser, and you’re ready to start listening.*

*Equipped with all the essential features of a modern music player, Saavn continues to grow with planned enhancements like playlist creation, song sharing, and real-time lyrics integration, promising an even richer user experience.*

*Discover Saavn today to enjoy a modern, high-performing music platform. Visit* [*saavn-web-ui.vercel.app*](https://saavn-web-ui.vercel.app) *or* [*saavn.codeforkers.tk*](https://saavn.codeforkers.tk) *to explore its capabilities.*

*.*

* + **General Introduction**

# INTRODUCTION

Music is a vital part of modern life, offering relaxation and entertainment to millions. Saavn is a web-based music player designed to deliver a seamless and high-quality audio experience. With streaming support of up to 320kbps, Saavn ensures crystal-clear sound, coupled with an ad-free, modern interface tailored for both desktop and mobile devices.

Saavn provides users with features like direct MP3 downloads with metadata, simplifying music collection and organization. Its vast library of songs, fetched from reliable sources, caters to diverse musical tastes.

As an open-source project, Saavn prioritizes performance and accessibility, allowing effortless deployment on platforms like Vercel, Heroku, and GitHub Pages. It requires no complex installations—just open the browser and start listening.

Currently equipped with all essential music player features, Saavn is constantly evolving, with future updates promising playlists, song sharing, and lyrics integration. Discover Saavn and redefine your music experience today.

### Problem Statement

Many music platforms compromise user experience with ads, poor audio quality, and limited features. Users struggle to find a high-quality, ad-free, and customizable solution. Saavn addresses this gap by offering a seamless, open-source web-based music player with high-quality streaming, direct MP3 downloads, and an evolving feature set.

### Problem detection

In today’s digital age, users face multiple issues while enjoying music online. Many music streaming platforms offer subpar audio quality, limited features, or cluttered interfaces, which disrupt the user experience. A significant challenge is the prevalence of ads, which interrupt listening and diminish the overall enjoyment. Additionally, platforms often lack user-friendly music management tools, such as easy downloads with metadata, and have poorly optimized designs for mobile or desktop interfaces. Furthermore, many music services are either closed-source or subscription-based, limiting customization and the ability to deploy the platform across various hosting services..

The need for an all-in-one, high-quality, and ad-free music player has become increasingly evident. Users require a solution that not only offers high-definition audio (up to 320kbps) but also allows seamless access to a diverse library of songs, MP3 downloads, and a straightforward interface. Furthermore, there is a growing demand for open-source platforms that offer flexibility, performance, and the ability to integrate advanced features like playlist creation, song sharing, and lyrics.

Saavn detects this gap by providing an open-source, web-based music player that ensures a smooth, high-quality, and customizable listening experience. It simplifies music management, offering direct downloads, mobile optimization, and is designed to meet the needs of modern users who seek more control and fewer limitations in their music streaming experience.

### Problem Solving

Saavn solves the challenges of digital music consumption by offering a high-quality, ad-free web-based music player with streaming support up to 320kbps. The platform provides a clean, user-friendly interface optimized for both desktop and mobile devices, ensuring seamless music enjoyment across all platforms.

It allows users to download MP3s with complete metadata, simplifying music management. As an open-source project, Saavn is highly customizable and easy to deploy on various hosting platforms without complicated installations. With upcoming features like playlists, song sharing, and lyrics integration, Saavn continuously improves to meet user demands for an all-in-one music solution.

### Proposed System Overview

Saavn is a web-based, open-source music player designed to address the shortcomings of existing digital music platforms. The system will offer high-quality streaming, up to 320kbps, with an intuitive, ad-free interface suitable for both desktop and mobile devices. It will provide a large, dynamic library of songs, easily accessible through a modern, responsive UI.

Key features of the system include:

* **High-Quality Streaming**: Songs streamed at up to 320kbps for a premium audio experience.
* **MP3 Downloads**: Users can download songs along with metadata for easy management and offline use

.

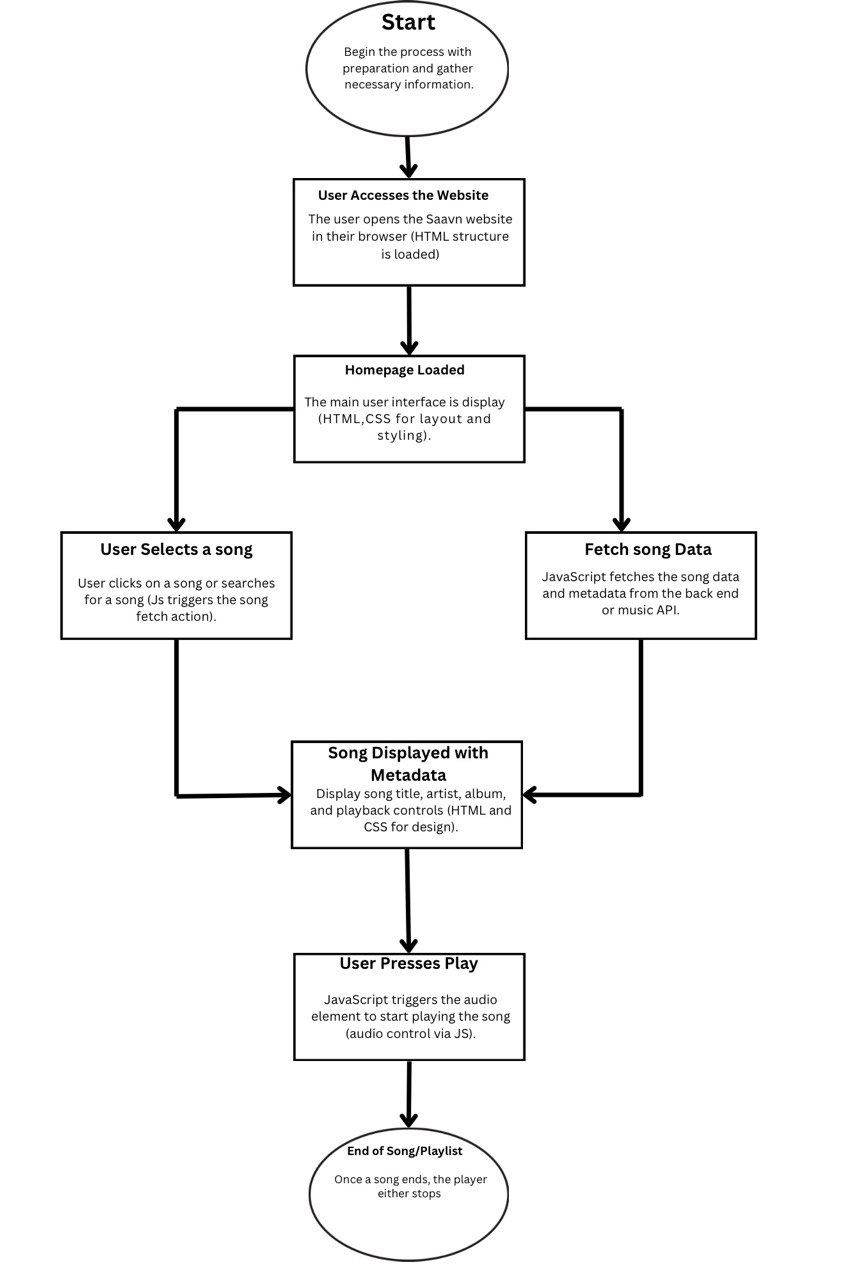
* **Cross-Platform Compatibility**: Optimized for both mobile and desktop devices, providing a consistent user experience across all screen sizes.
* **Open-Source & Customizable**: As an open-source platform, Saavn can be easily modified, customized, and deployed on various hosting platforms like Vercel, Heroku, and GitHub Pages.
* **Ad-Free Experience**: Users will enjoy uninterrupted music streaming without ads.
* **Future Features**: The system will continue to evolve with planned additions like playlist creation, song sharing, and lyrics integration.

Saavn's system is designed to offer a seamless, high-quality, and flexible music player experience, aiming to meet the growing demands of modern music listeners

**Principles of Saavn**

* **High-Quality Audio**: Providing users with premium audio quality, supporting streaming up to 320kbps for an enhanced listening experience. Welcome changing requirements.
* **Ad-Free Experience**: Ensuring uninterrupted music enjoyment with no advertisements, promoting a smooth and engaging user experience.
* **User-Centric Design**: Creating a modern, intuitive interface optimized for both desktop and mobile platforms, making it easy for users to navigate.
* **Open-Source Philosophy**: Empowering the community by providing access to the platform’s source code, allowing customization, development, and transparency
* **Cross-Platform Compatibility**: Offering seamless functionality across different devices and platforms, ensuring users can access music anywhere, anytime.
* **Easy Music Management**: Simplifying the download and organization of music with MP3 downloads and metadata support for easy file handling.
* **Continuous Evolution**: Regularly updating and improving the platform with new features like playlists, song sharing, and lyrics integration based on user feedback.
* **Performance Optimization**: Ensuring fast load times, reliable streaming, and smooth operation by optimizing the code for maximum performance.

### Flow Chart



**Fig. 1.4.2 Flow Chart**

**CODING OF BACKEND**

The backend for Saavn using JavaScript can be built with Node.js. It handles API requests, serves static files, and streams music data. Key functionalities include routing with express, fetching metadata, and managing downloads. Integration with external APIs or a local music library ensures dynamic song fetching and seamless playback functionality.

JS code screenshort :-

function PlayAudio(audio\_url, song\_id) {

  var audio = document.getElementById('player');

  var source = document.getElementById('audioSource');

  source.src = audio\_url;

  var name = document.getElementById(song\_id+"-n").textContent;

  var album = document.getElementById(song\_id+"-a").textContent;

  var image = document.getElementById(song\_id+"-i").getAttribute("src");

document.title = name+" - "+album;

var bitrate = document.getElementById('saavn-bitrate');

var bitrate\_i = bitrate.options[bitrate.selectedIndex].value;

var quality = "";

if (bitrate\_i == 4) {quality = 320} else {quality = 160;}

    document.getElementById("player-name").innerHTML = name;

        document.getElementById("player-album").innerHTML = album;

document.getElementById("player-image").setAttribute("src",image);

var promise = audio.load();

if (promise) {

    //Older browsers may not return a promise, according to the MDN website

    promise.catch(function(error) { console.error(error); });

}//call this to just preload the audio without playing

  audio.play(); //call this to play the song right away

};

function searchSong(search\_term) {

document.getElementById('search-box').value=search\_term;

var goButton = document.getElementById("search-trigger");

            goButton.click();

}

var DOWNLOAD\_API = "https://openmp3compiler.astudy.org"

function AddDownload(id) {

    var bitrate = document.getElementById('saavn-bitrate');

    var bitrate\_i = bitrate.options[bitrate.selectedIndex].value;

    // MP3 server API

    var MP3DL = DOWNLOAD\_API+"/add?id="+id;

    // make api call, if 200, add to download list

    fetch(MP3DL)

    .then(response => response.json())

    .then(data => {

        if (data.status == "success") {

            // add to download list

            var download\_list = document.getElementById("download-list");

            var download\_item = document.createElement("li");

           /\*

           <li>

                    <div class="col">

                        <img src="https://i.pinimg.com/originals/ed/54/d2/ed54d2fa700d36d4f2671e1be84651df.jpg" width="50px">

                        <div style="display: inline;">

                        <span id="download-name">Song</span>

                        <span id="download-album">Album</span>

                        <br>

                        <span id="download-size">Size</span>

                        <span id="download-status" style="color:green">Compiling.</span>

                        </div>

                    </div>

                    <hr>

                    </li>

           \*/

            // download\_item.innerHTML = '<div class="col"><img src="'+data.image+'" width="50px"><div style="display: inline;"><span id="download-name">'+id+'</span><span id="download-album">'+data.album+'</span><br><span id="download-size">'+data.size+'</span><span id="download-status" style="color:green">Compiling.</span></div></div><hr>';

            download\_item.innerHTML = `

            <div class="col">

            <img class="track-img" src="${data.image}" width="50px">

            <div style="display: inline;">

              <span class="track-name"> ${id}</span> -

              <span class="track-album"> ${data.album}</span>

              <br>

              <span class="track-size"> Size : Null</span>

              <span class="track-status" style="color:green"> </span>

            </div>

          </div>

          <hr>

            `;

            // set download\_item track\_tag to song id

            download\_item.setAttribute("track\_tag",id);

            // set css class no-bullets

            download\_item.className = "no-bullets";

            download\_list.appendChild(download\_item);

            // every 5 seconds, check download status

            var STATUS\_URL = DOWNLOAD\_API+"/status?id="+id;

            // get download\_status\_span by track\_tag and class

            var download\_status\_span = document.querySelector('[track\_tag="'+id+'"] .track-status');

            var download\_name = document.querySelector('[track\_tag="'+id+'"] .track-name');

            var download\_album = document.querySelector('[track\_tag="'+id+'"] .track-album');

            var download\_img = document.querySelector('[track\_tag="'+id+'"] .track-img');

            var download\_size = document.querySelector('[track\_tag="'+id+'"] .track-size');

            // set text content to song name and album name

            download\_name.innerHTML= results\_objects[id].track.name;

            download\_status\_span.innerHTML = data.status;

            download\_album.innerHTML = results\_objects[id].track.album.name;

            download\_img.setAttribute("src",results\_objects[id].track.image[2].link);

            // change mpopupLink background and border color to green and back to blue after 1 second

            var float\_tap = document.getElementById('mpopupLink');

            float\_tap.style.backgroundColor = "green";

            float\_tap.style.borderColor = "green";

            setTimeout(function() {

                float\_tap.style.backgroundColor = "#007bff";

                float\_tap.style.borderColor = "#007bff";

            }, 1000);

            // check status every 5 seconds

            var interval = setInterval(function() {

                fetch(STATUS\_URL)

                .then(response => response.json())

                .then(data => {

                    if (data.status) {

                        // update status

                        download\_status\_span.textContent = data.status;

                        if(data.size) {

                            download\_size.textContent = "Size: "+data.size;

                        }

                        if (data.status == "Done") {

                            // download complete, add download button

                            download\_status\_span.innerHTML = `<a href="${DOWNLOAD\_API}${data.url}" target="\_blank">Download MP3</a>`;

                            // clear interval

                            clearInterval(interval);

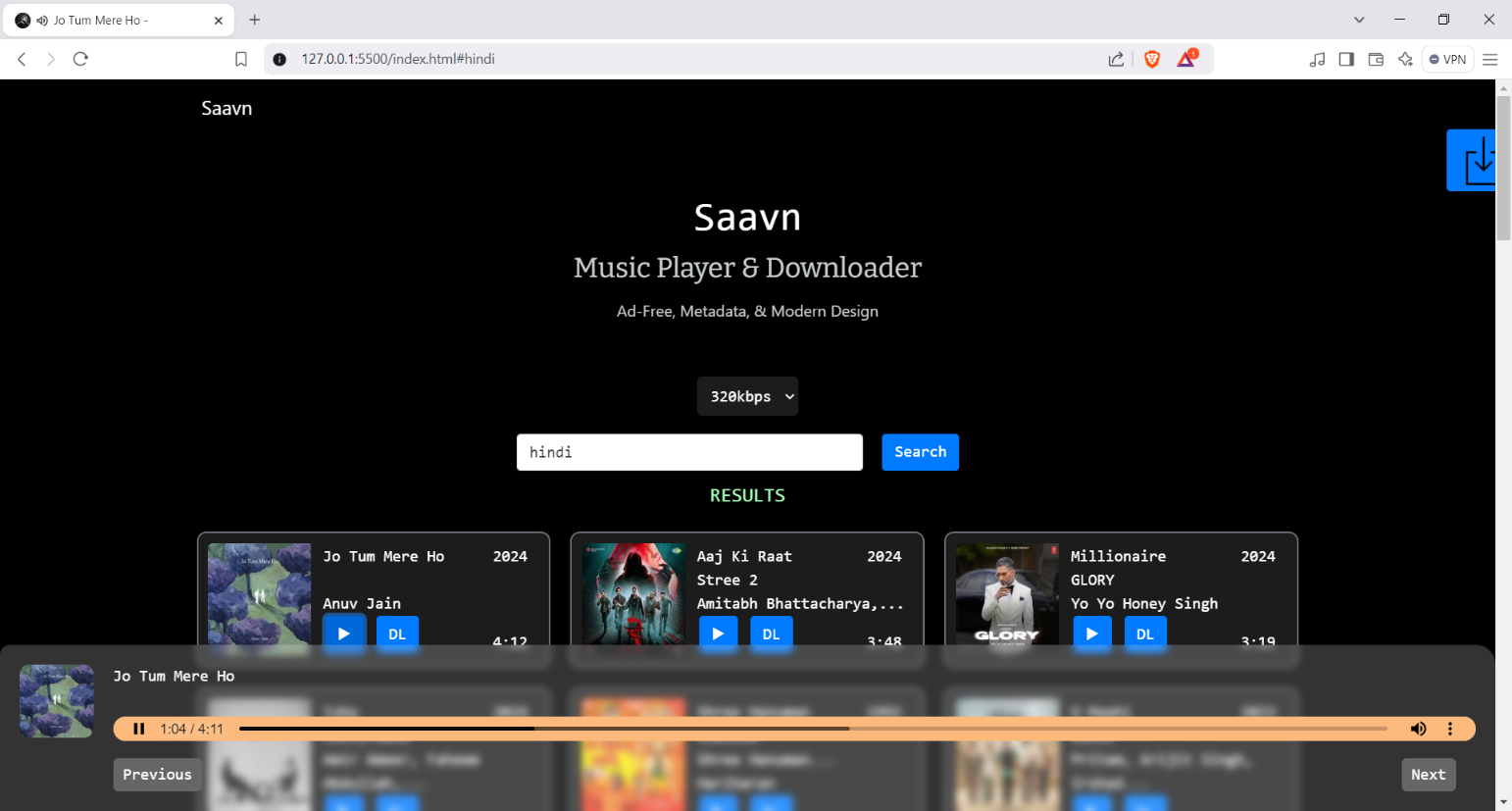
                            return;

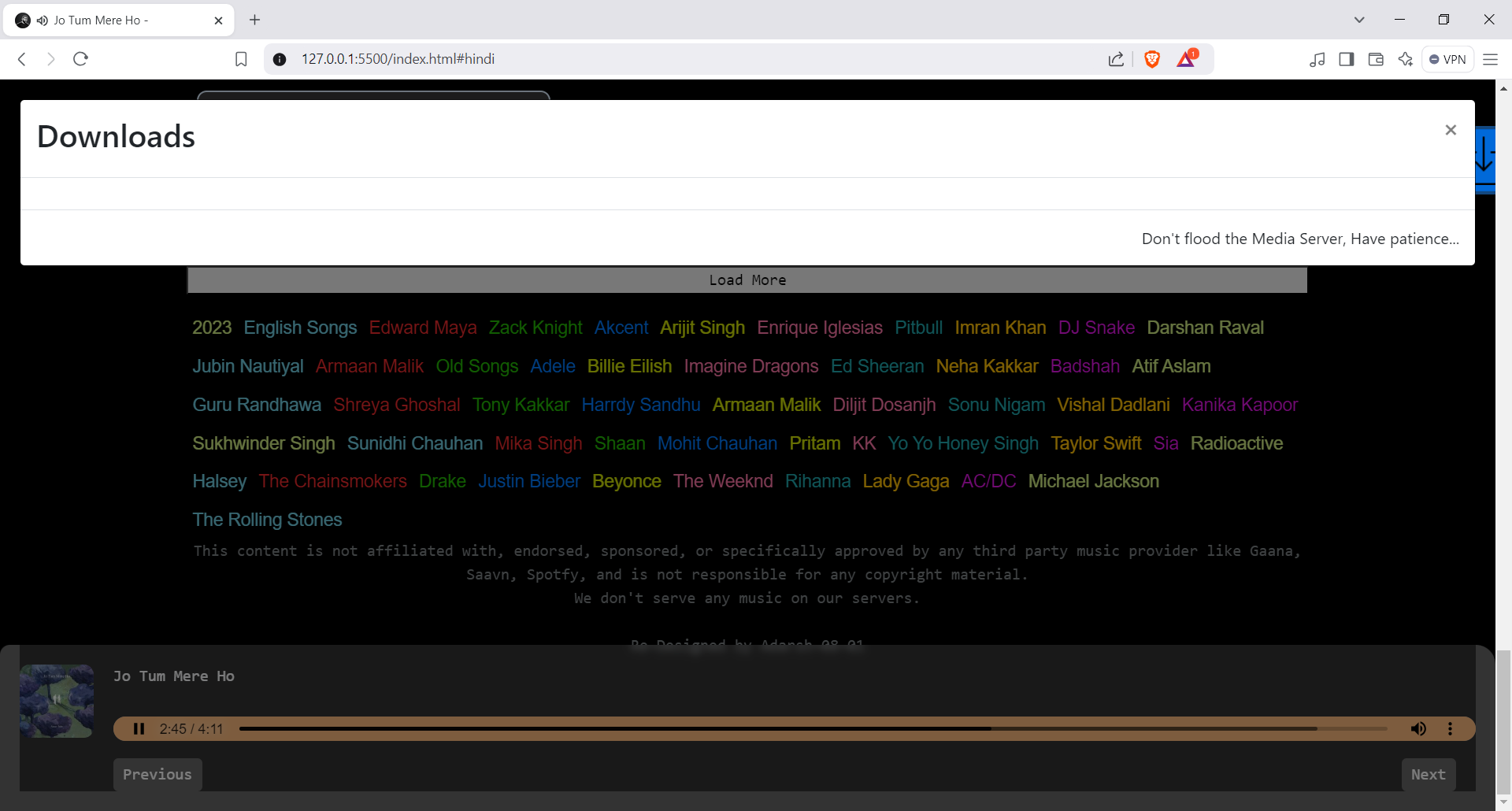
                  }}

              });}, 3000); // end interval

        } });}

Screenshot:-





**CONCLUSION**

The creation of the web music player using HTML, CSS, JavaScript, and the Saavn API demonstrates an innovative approach to building a dynamic and functional application. By leveraging the Saavn API, the project successfully integrates a vast library of music, allowing users to explore and enjoy their favorite tracks effortlessly. The use of JavaScript facilitates real-time interactions, ensuring seamless playback, search functionality, and dynamic content updates, while HTML and CSS provide the foundation for a visually appealing and responsive design.

This project highlights the importance of API integration in modern web applications, enabling access to extensive resources and enhancing user experience. Key features such as playback controls, volume adjustments, and song recommendations reflect a comprehensive understanding of frontend development and API utilization.

The web music player not only addresses the technical aspects of application development but also emphasizes user-centric design principles. Future enhancements could include personalized playlists, user authentication, or offline playback options to elevate the platform further.

Overall, this project is a testament to the effective combination of web technologies and external APIs, showcasing both technical proficiency and creative problem-solving in the development of an engaging and robust music player.

### Future Scope

.

The web music player built with HTML, CSS, JavaScript, and the Saavn API holds significant potential for future enhancements. Here are some key areas for development:

1. **Personalized Playlists**  
   Implement user authentication to allow users to create, save, and manage their playlists. By tracking user preferences, the application can suggest curated playlists and songs tailored to individual tastes.
2. **Offline Playback**  
   Introduce offline functionality, enabling users to download their favorite tracks for listening without an active internet connection.
3. **Enhanced Search and Discovery**  
   Upgrade the search feature with advanced filters such as genre, mood, artist, or album to improve user experience. Adding music recommendations based on user history or trending tracks could boost engagement.
4. **Social Integration**  
   Allow users to share songs and playlists on social media platforms, fostering community interaction. Features like collaborative playlists could further enhance user engagement.
5. **Audio Visualizations and Themes**  
   Add visually dynamic audio visualizations synchronized with the music. Customizable themes and color schemes can enhance personalization.
6. **Multi-Platform Compatibility**  
   Optimize the application for various devices, including mobile, tablet, and smart TVs, ensuring a consistent experience across platforms.
7. **Monetization Opportunities**  
   Explore monetization options like ad integration or premium subscriptions for additional features like ad-free streaming, exclusive content, or high-quality audio playback.

.

### Uses & Targets

The web music player caters to music enthusiasts seeking a seamless and engaging platform for discovering and streaming their favorite tracks. With features like dynamic search, playback controls, and integration with the Saavn API, it offers an intuitive experience for casual listeners and avid music lovers alike.

Target users include individuals across all age groups who enjoy music, tech-savvy audiences seeking innovative tools, and professionals like gym trainers or makeup artists requiring curated playlists. Businesses can also leverage the player for ambiance in retail, cafes, or events.

This versatile tool ensures accessibility, personalization, and entertainment for diverse user needs.

### Applications

1. **Personal Use**

Enjoy seamless music streaming for relaxation, workouts, or daily activities.

1. **Professional Settings**

Create ambiance in gyms, salons, or during makeup sessions with curated playlists.

1. **Event Management**

Use themed playlists for parties, weddings, and corporate events.

1. **Business Use**

Enhance customer experience with background music in cafes, retail stores, or offices.

### Result & Analysis

The web music player successfully integrates the Saavn API to provide a seamless music streaming experience. Users can explore a vast library of tracks, utilize playback controls, and enjoy a responsive and visually appealing interface. The integration of HTML, CSS, and JavaScript ensures smooth performance and interactivity.

**Results:**

1. **Functionality**: Core features like play, pause, search, and volume control operate efficiently.
2. **User Experience**: Aesthetic design and responsive layout offer a user-friendly interface.
3. **Accessibility**: The platform is accessible across various devices, enhancing usability.

**Analysis:**

* The player meets its intended goals of dynamic music streaming and user engagement.
* Potential improvements, such as advanced search filters and personalized playlists, could further elevate its value.
* Overall, the project demonstrates effective use of web technologies and API integration to deliver a robust and scalable solution.

.

### Limitations

1. **Dependency on Saavn API**

The functionality relies heavily on the Saavn API; any changes or outages in the API could disrupt the service.

1. **Limited Offline Capability**

Currently, the player does not support offline playback, restricting usage in areas with poor internet connectivity.

1. **Personalization Gaps**

The absence of user authentication limits features like saved playlists, preferences, or recommendations.

1. **Device Compatibility**

While responsive, the player may require further optimization for seamless performance across all devices, including older browsers or operating systems.

1. **Monetization Restrictions**

Without a premium version or ad support, the platform lacks revenue-generation capabilities.

1. **Advanced Features Missing**

Lacks features like audio visualization, equalizer settings, or integration with other music platforms.