A Project Report
On
MelodyHub

Submitted by

Garima Mishra – 2115000396

Ishita Tripathi – 2115000481

Hanshika Agarwal – 2115000423

Ritika Agrawal – 2115000859

Supervisor

Mr. Shiv Kumar Verma
Faculty
Coding Blocks

Department of Computer Engineering & Application
G.L.A. UNIVERSITY

GLA University, Mathura - 281406

13/05/2024

BONAFIDE CERTIFICATE

Certified that this project report "**MelodyHub**" is the Bonafede work of "Garima Mishra – 2115000396

Ishita Tripathi – 2115000481

Hanshika Agarwal – 2115000423

Ritika Agrawal – 2115000859"
who carried out the project work under my supervision.

SIGNATURE (HOD) **SIGNATURE (SUPERVISOR)**

HEAD OF THE DEPARTMENT
CSE Department

Mr. Shiv Kumar Verma
(Coding Blocks)
CSE Department

Submitted for the project viva-voce examination held on 13th May 2024

ACKNOWLEDGEMENT

Presenting the ascribed project paper report in this very simple and official form, we

would like to place my deep gratitude to GLA University for providing us with the instructor Mr. Shiv Kumar Verma, our technical trainer and supervisor.

He has been helping us since Day 1 of this project. He provided us with the roadmap, and the basic guidelines explaining on how to work on the project. He has been conducting regular meetings to check the progress of the project and providing us with the resources related to the project. Without his help, we wouldn't have been able to complete this project.

And at last, but not least we would like to thank our dear parents for helping us to grab this opportunity to get trained and also my colleagues who helped me find resources during the training.

Thanking You

Name of Candidates:

Garima Mishra (2115000396)

Ishita Tripathi (2115000481)

Hanshika Agarwal (2115000423)

Ritika Agrawal (2115000859)

CERTIFICATE

This is to certify that the above statement made by the students is correct to the best of my knowledge and belief.

Date:

Place: Mathura

Name and Signature with Affiliation of Supervisor

Mr. Shiv Kumar Verma

CONTENTS: -

Table of Contents

1. ABSTRACT

Melody Hub is a music streaming platform where users can access a wide range of music content across various genres. In order to access the platform's features, users are required to create an account on Melody Hub. Once registered, users can explore and enjoy music from different categories.

KEY FEATURES: -

1. User Authentication and Management:

- **User Registration/Login:** Allow users to create accounts and log in securely to access Melody Hub's features.
- **User Profiles:** Enable users to manage their profiles, including personal preferences, playlists, and account settings.

2. Music Catalogue and Search:

- **Music Listing:** Display a diverse range of music content with clear titles, artists, genres, and album artwork.
- **Music Details:** Show detailed information about each track, including duration, album name, and release date.
- **Search Feature:** Provide users with a search bar to find specific songs, artists, or albums quickly.

3. Playback Controls:

- **Play/Pause:** Allow users to play or pause music tracks seamlessly.
- **Skip/Rewind:** Enable users to skip or rewind tracks within a playlist for personalized listening.

CHAPTER-1

INTRODUCTION

1.1 CONTEXT: -

Melody Hub is a specialized music streaming platform catering to music enthusiasts of all genres and tastes. With a focus on delivering quality music content, fostering

community engagement, and providing a seamless user experience, Melody Hub aims to become the go-to destination for music lovers worldwide.

1.2 MOTIVATION: -

Melody Hub is motivated by a passion for bringing music enthusiasts together and providing them with a comprehensive platform to explore, discover, and enjoy music of all genres. The driving force behind Melody Hub is the desire to democratize access to high-quality music content and create a vibrant community of music lovers.

1.3 OBJECTIVE: -

Melody Hub is an online platform committed to providing music enthusiasts with a diverse range of high-quality music content and features. Our objective is to offer users a seamless music streaming experience, presenting a curated collection of tracks, playlists, and features tailored to their musical preferences. Through Melody Hub, we aim to foster a thriving community of music lovers and promote a deeper engagement with music across all genres and styles.

1.4 EXISTING SYSTEM: -

The existing systems for a music-focused streaming platform like Melody Hub involve studying industry-leading platforms (such as Spotify, Apple Music) to understand user preferences, content curation strategies, and technological advancements in the music streaming industry.

CHAPTER -2

SOFTWARE REQUIREMENT ANALYSIS

2.1 IMPACT OF THIS ON DAILY LIFE: -

Accessibility and Convenience: Melody Hub provides individuals with easy access to a diverse range of music content from anywhere, eliminating the need to visit physical stores or

download multiple music files. This convenience saves time and effort, allowing users to enjoy their favorite music seamlessly.

Mental Well-being: Listening to music has been shown to have numerous mental health benefits, including reducing stress, improving mood, and enhancing overall well-being. Melody Hub contributes to mental wellness by offering a wide variety of music genres and playlists to suit different moods and preferences, providing users with a source of relaxation and enjoyment in their daily lives.

2.2 PROBLEM STATEMENT: -

Current limitations in accessing diverse music content and expert recommendations prompt the need for an all-inclusive music streaming platform. This project seeks to fill this gap by offering a wide range of high-quality music tracks and curated playlists, creating a supportive community hub for music enthusiasts. Melody Hub aims to empower users in their daily pursuit of discovering and enjoying music across various genres and styles.

2.3 HARDWARE AND SOFTWARE REQUIREMENTS: -

Hardware Requirement

Processor: any smartphone processor

• RAM: 8 GB (or higher)

• Hard disk: 256GB

Software Requirement

• Software used: Visual Studio Code

• Language used: HTML, CSS, JavaScript, Bootstrap, Json,Node,Express,MongoDb

• User Interface Design: Website

2.4 MODULES AND FUNCTIONALITIES

Modules:

1. **User Management:**
    - User Registration/Login
    - User Profiles
2. **Song Management:**
    - Music Catalog

- Music Reviews and Ratings
- Playlist Management

3. **Playlist Management:**
   - Playlist Creation
   - Edit Playlists

Functionalities:

1. **User Experience:**
   - Responsive Design for various devices
   - Intuitive and User-friendly Interface
   - Online/Offline Status
2. **Music Presentation:**
   - High-quality audio
   - Track Details
   - Playlist Management
3. **Search and Navigation:**
   - Advanced Search with Filters (by songs, artists, or albums.)
   - Easy Navigation through Categories and Subcategories
4. **Security and Trust:**
   - Trusted Authentication
   - Trusted Music Sources

CHAPTER – 3

IMPLEMENTATION AND USER INTERFACE

Week 1: Planning and Design

Day 1-2: Requirement Gathering: - Brainstorm phase to gather inspiration and vision for the Melody Hub user interface design.

Day 3-4: Wireframing and Design: - Sketch rough layouts for webpages, including homepage, playlist pages, and user profile. Select color schemes, typography, and imagery that align with the music theme.

Day 5-7: Prototype Development: - Create the skeleton for the website using basic HTML tags and add basic CSS functionalities for layout and styling.

Week 2-3: Frontend Development

Day 8-10: Tech Stack Selection

- Choose frontend technologies suitable for building a music streaming platform.

Day 11-14: Frontend Setup and Structure

- **Setup Project:** Initialize the project structure, set up version control (like Git), and create the basic folder hierarchy.
- **Homepage Development:** Begin coding the homepage layout, integrating elements from the design phase.

Day 15-17: Playlist Listing and Detail Pages

- Implement functionality to display playlists categorically.

- Develop individual playlist pages with detailed track listings and playback options.

Day 18-21: User Management and Playback Controls

- Implement user authentication and profile management features.
- Develop playback controls for seamless music streaming.

Week 4: Refinement and Testing

Day 22-24: Optimization and Responsive Design

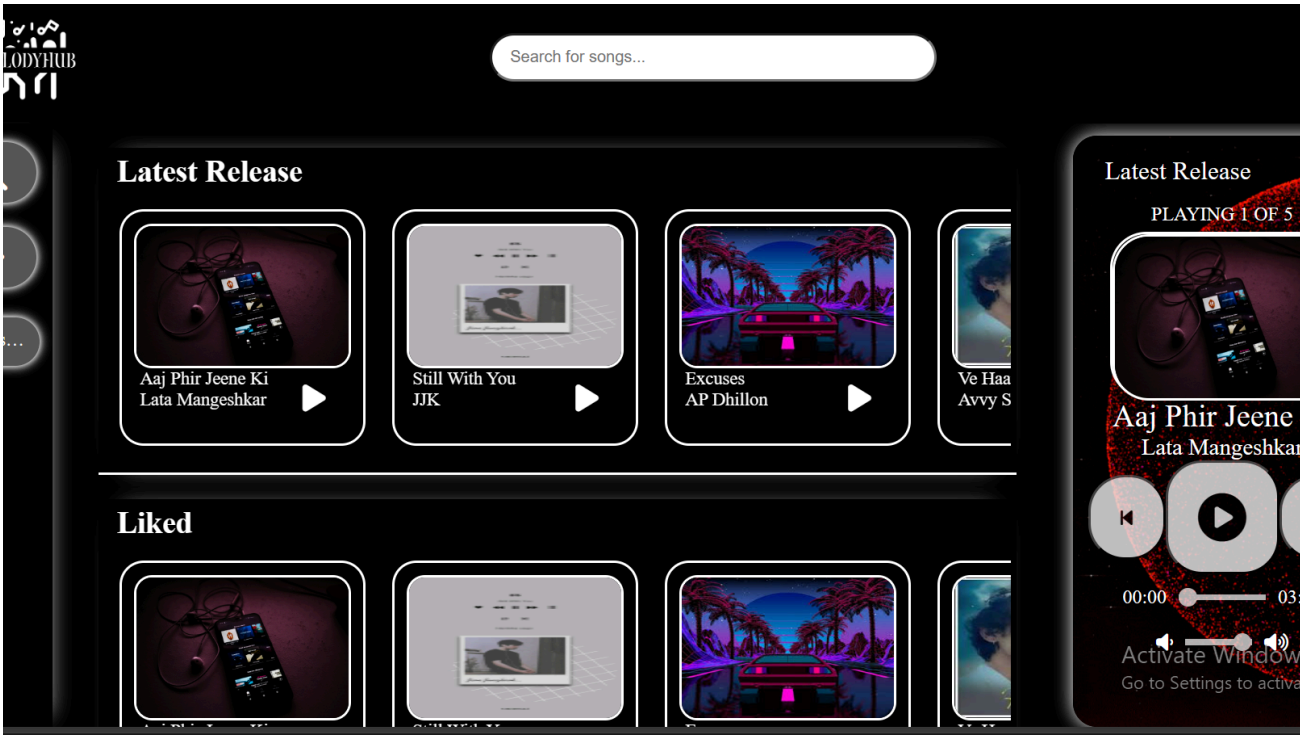
- **Optimization:** Refine code for performance and speed optimization.
- **Responsive Design:** Ensure the website is fully responsive across devices (desktop, mobile, tablet).

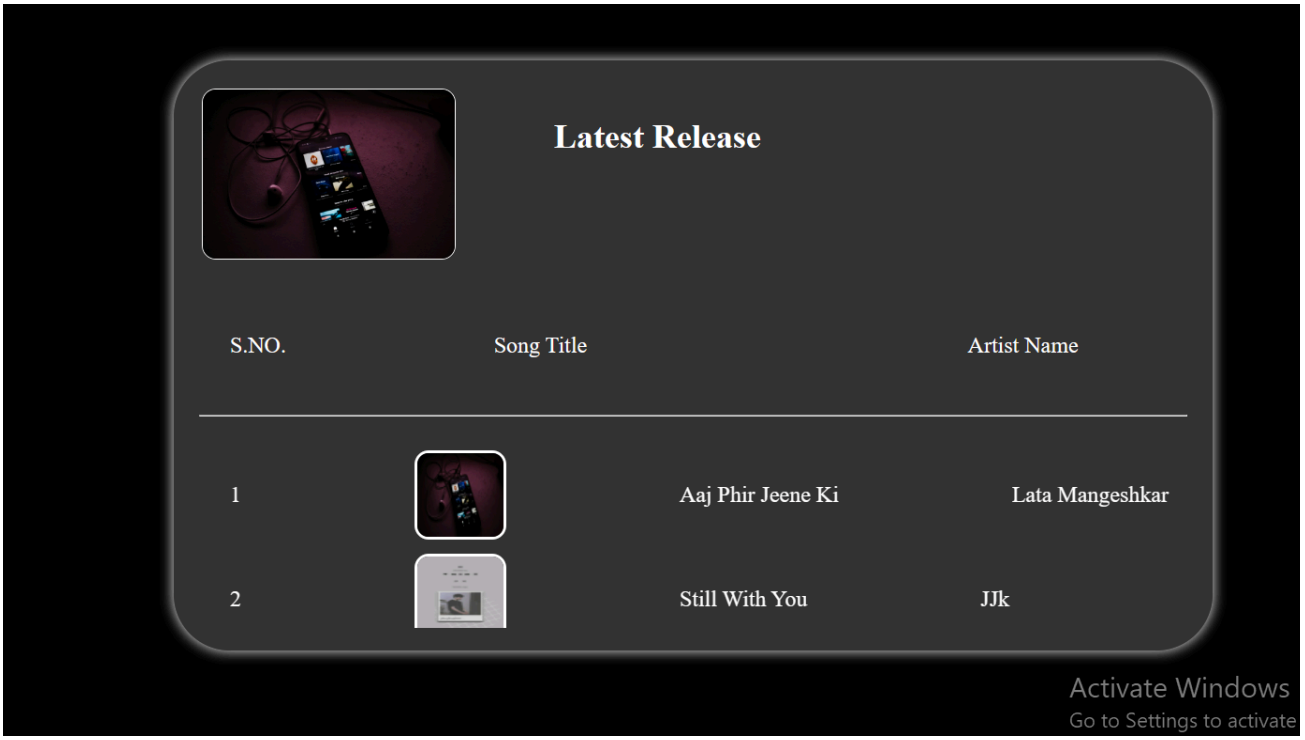
Day 25-30: User Testing and Feedback and Final Polishing

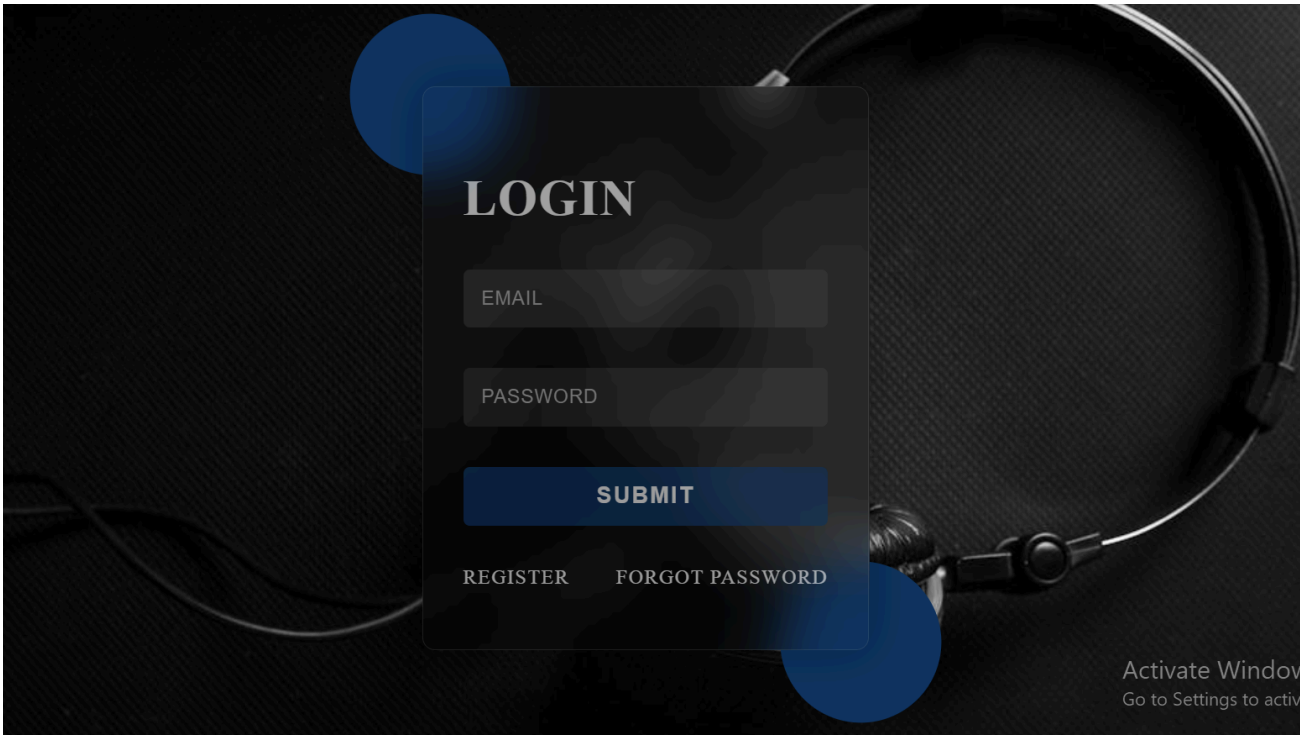
Testing and Bug Fixes: Basic testing is conducted to identify and resolve critical bugs and ensure functional stability.
Deployment and Hosting: The website is deployed and hosted on a chosen platform, ready for users to access and utilize.

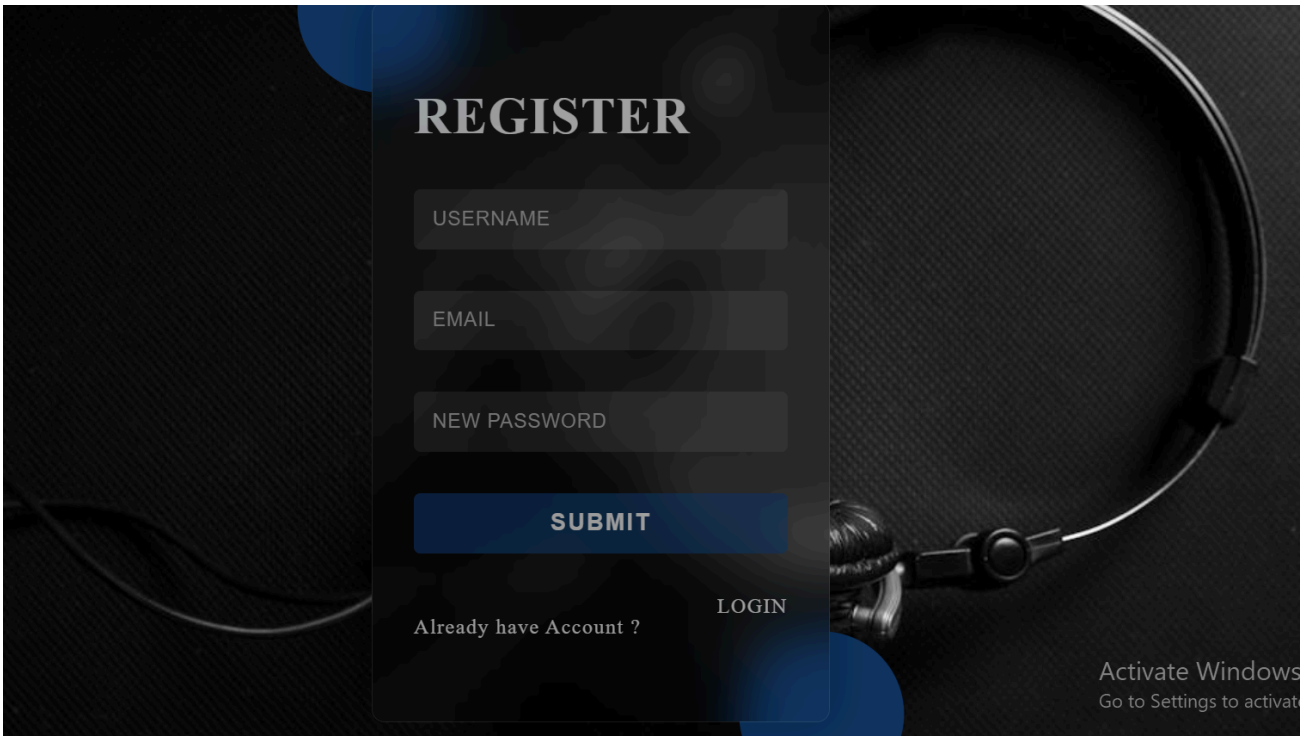
CHAPTER – 4

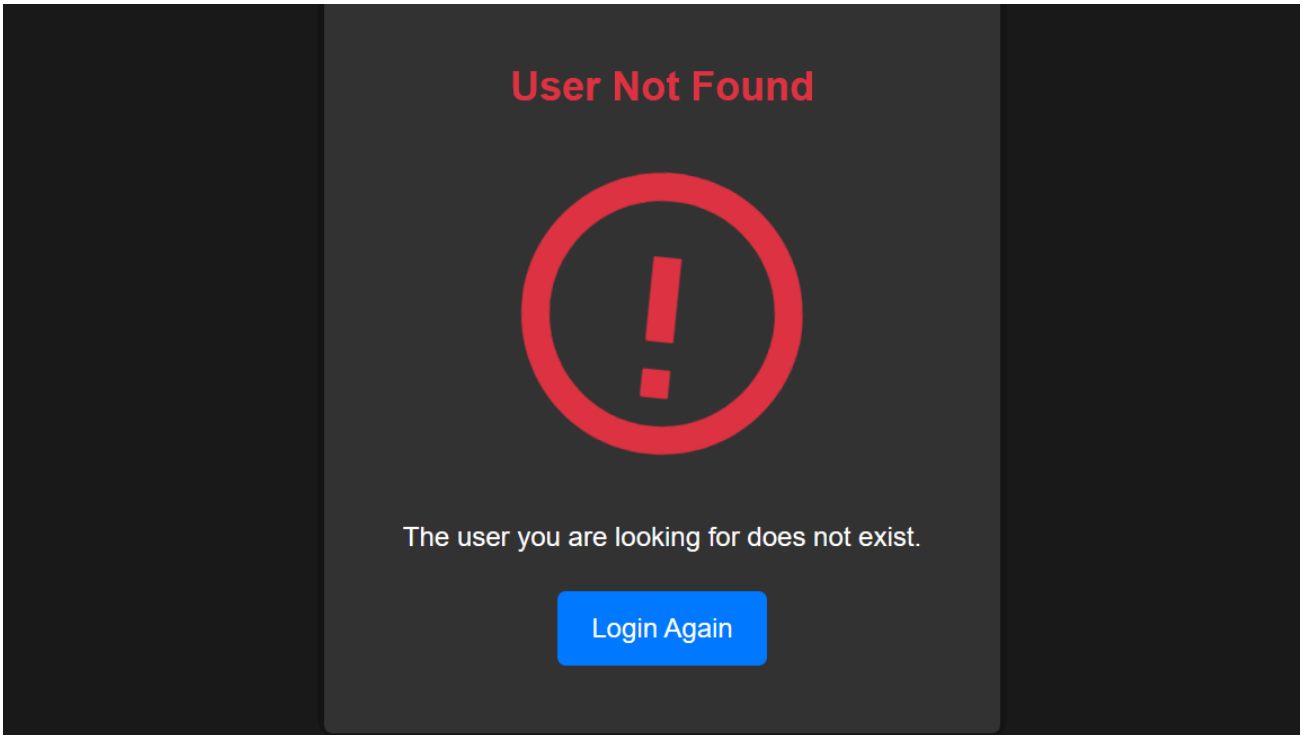
USER INTERFACE AND CODE SNIPPETS



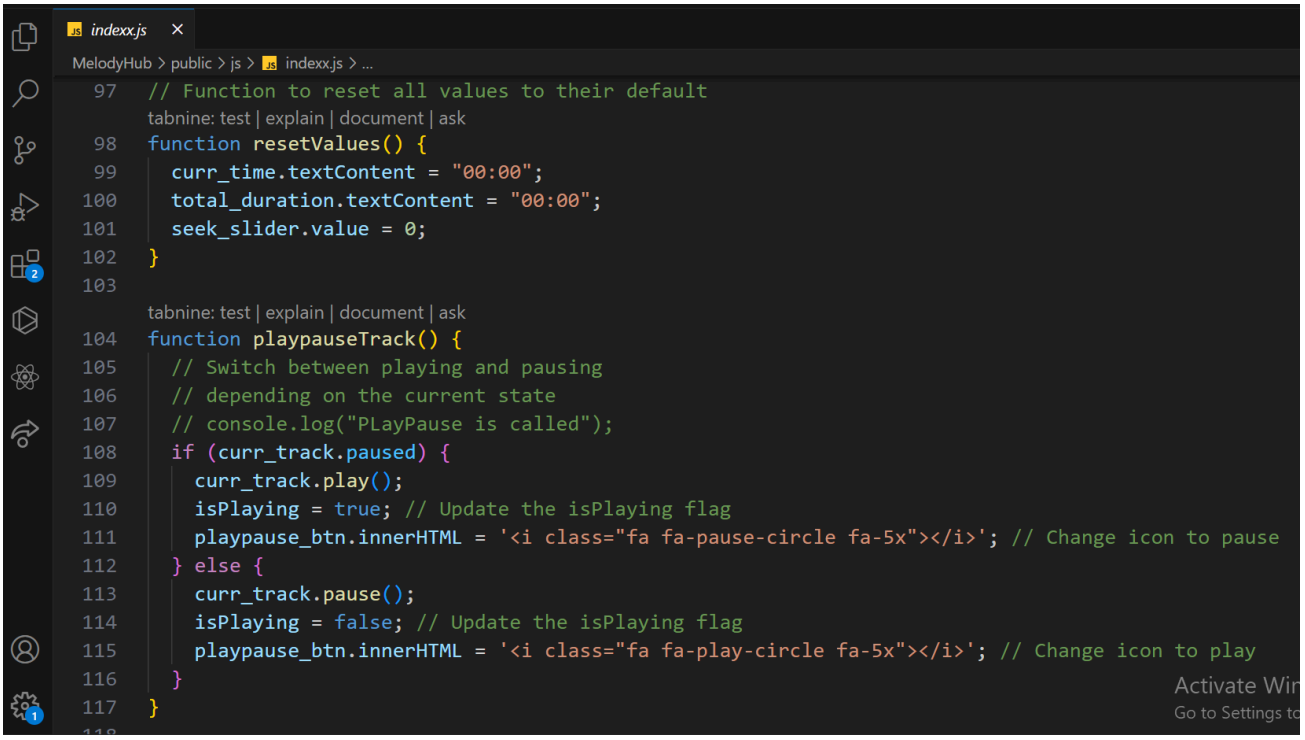



LOGIN
EMAIL
PASSWORD
SUBMIT
REGISTER
FORGOT PASSWORD
Activate Windo
Go to Settings to acti


REGISTER
USERNAME
EMAIL
NEW PASSWORD
SUBMIT
LOGIN
Already have Account ?
Activate Windows
Go to Settings to activat




```
const express = require("express");
const app = express();
const path = require("path");
const bodyParser = require("body-parser");
const cookieParser = require("cookie-parser");
const userRoutes = require("./routes/user.routes.js");
const songRoutes = require("./routes/songs");

app.use(bodyParser.json());
app.use(cookieParser());

const mongoose = require("mongoose");

const Song = require("./models/Song");

const Playlist = require("./models/Playlist");
const authRoutes = require("./routes/user.routes.js");

const { songs, seedSongs } = require("./seeds/songSeed");
```



```
indexx.js
MelodyHub > public > js > indexx.js > ...
// Function to reset all values to their default
tabnine: test | explain | document | ask
function resetValues() {
  curr_time.textContent = "00:00";
  total_duration.textContent = "00:00";
  seek_slider.value = 0;
}

tabnine: test | explain | document | ask
function playpauseTrack() {
  // Switch between playing and pausing
  // depending on the current state
  // console.log("PLayPause is called");
  if (curr_track.paused) {
    curr_track.play();
    isPlaying = true; // Update the isPlaying flag
    playpause_btn.innerHTML = '<i class="fa fa-pause-circle fa-5x"></i>'; // Change icon to pause
  } else {
    curr_track.pause();
    isPlaying = false; // Update the isPlaying flag
    playpause_btn.innerHTML = '<i class="fa fa-play-circle fa-5x"></i>'; // Change icon to play
  }
}
```

```
const User = require("../models/User.js");
const { songs, seedSongs } = require("../seeds/songSeed");

const generateToken = async (id) => {
  try {
    const user = await User.findOne(id);
    const accessToken = await user.generateAcessToken();
    const refreshToken = await user.generateRefreshToken();
    user.refreshToken = refreshToken;
    await user.save({ validityBeforeSave: false }); //
    return { accessToken, refreshToken };
  } catch (error) {
    return "error while generating tokens";
  }
};

const registerUser = async (req, res) => {
  let { name, email, password } = req.body; //destructuring
  console.log(req.body);
  if ([name, email, password].some((field) => field.trim() === "")) {
```

CHAPTER -5

CONCLUSION

In developing this music streaming platform using Node.js, Express.js, and MongoDB, our aim was to create a seamless and immersive experience for music enthusiasts while incorporating essential functionalities. The implementation involved careful structuring with Node.js, expressive styling using CSS, and dynamic functionality through Express.js.

Through deliberate UI/UX design choices, we endeavored to provide users with an intuitive and enjoyable interface. From easy navigation to the visual presentation of music content and the smooth playback experience, every element was designed to enhance the user's interaction with the platform.

By prioritizing responsive design principles, optimizing performance, and ensuring secure authentication methods, our objective was to build a versatile and trustworthy platform accessible across various devices while safeguarding user data and privacy.

In conclusion, this project embodies the harmonious integration of technology and user-centric design, aimed at meeting the diverse needs of modern music enthusiasts. As we wrap up this endeavor, our focus remains on delivering a functional, efficient, and visually appealing music streaming solution that enhances the user's musical journey and contributes to a vibrant online music community.

REFERENCES

GitHub Link:-

https://github.com/Garima9604/MiniProject_2

HTML:

1. **W3Schools**:
   - HTML Tutorial

CSS:

1. **W3Schools**:
   - CSS Tutorial

JavaScript:

1. **MDN Web Docs**:
   - JavaScript Guide
2. **W3Schools**:
   - JavaScript Tutorial

Node:

1. **MDN Web Docs**:
   - Node.js Guide
2. **W3Schools**:
   - Node.js Tutorial

MongoDb:

3. **MDN Web Docs**:
   - MongoDb Guide
4. **W3Schools**:
   - MongoDb Tutorial

Express:

1. **MDN Web Docs**:
   - Express Guide
2. **W3Schools**:
   - Express Tutorial

Version Control (Git):

Git documentation: https://git-scm.com/doc

Music Streaming Platform Development Guides:

Softermii: Music Streaming App Development Guide: Key Features & Cost - Softermii