RhythmicTunes: Your Melodic Companion

1. Introduction

• Project Title:rhythmictunesyourmelodiccomanion

• TeamID:NM2025TMID32311

• TeamLeader:dhanushya s$dhanushyadhanu574@gmail.com

• TeamMembers:deepa m

dhanalakshmi k

dhivya s

2. Project Overview

• Purpose: RhythmicTunes is a modernmusic webapplicationthat allows users toexplore, stream,

andorganize their favorite tracks. It provides personalizedplaylists, intelligent recommendations,

anda smoothlisteningexperience. The platformis designedtobe a user's melodic companionfor

discoveringandenjoyingmusic.

• Features:

•

•

•

Music streamingwithhigh-qualityplayback

Personalizedplaylists andrecommendations

Searchanddiscover newsongs andartists

User libraryfor favorites andplaylists

Community feedback and ratings system

Admin panel for managing content

•

•

•

3. Architecture

•

Frontend: React.js withBootstrapandMaterial UI

Backend: Node.js andExpress.js for server logic andAPIs

Database: MongoDBfor storinguser data, playlists, andtracks

•

•

4. SetupInstructions

•

•

Prerequisites: Node.js, MongoDB, Git, React.js, Express.js, Visual StudioCode

Clone the repository: git clone

Install client dependencies: cdclient &&npminstall

Install server dependencies: cd../server &&npminstall

•

•

5. Steps toExecute Project

•

•

•

•

DownloadNode.js LTSversionfromhttps://nodejs.org/en/download/

Install Node.js followingthe installer instructions for Windows/macOS

Openterminal/PowerShell andset executionpolicyif needed

Access project resources fromSmartInternz portal

Downloadandextract project source code

OpeninVSCode andrunnpminstall toinstall dependencies

Start the development server withnpmrundev

Access applicationat http://localhost:5173

Run JSONserver: cd db &&json-server --watchdb.json--port 3000

•

•

•

•

•6. API Documentation

•

•

/api/user/register – User registration

/api/user/login– User login

/api/music/search– Searchsongs andartists

/api/music/playlist – Manage playlists

/api/music/recommend – Get personalized recommendations

•

•

•

7. Authentication

• JWT-basedauthenticationfor secure login

• Middleware protects private routes

8. User Interface

•

•

Landing Page with featured tracks and recommendations

User Dashboardwithpersonalizedplaylists

Admin Panel for content management

Music Player withqueue andplaybackcontrols

•

•

9. Testing

• Manual testingperformedduringdevelopment milestones

• Tools used: Postman, Chrome DevTools

10. Future Enhancements

•

•

Mobile applicationfor iOSandAndroid

Offline playbacksupport

Integrationwithexternal streamingservices

AI-driven smart recommendations

Social sharingfeatures

•

•

•