# Introduction

**PROJECT TITLE : RHYTHMICTUNES: YOURMELODICCOMPANION**

TEAMID :NM2025TMID44555

TEAMLEADER: R Asvitha & asvitharavikumar75@gmail.com

TEAM MEMBER : M Ruthrashavarni & muruganruthra545@gmail.com

TEAM MEMBER : S Dharshini & dharshinisenthilkumar2007@gmail.com

TEAM MEMBER : B Gopika & gopika1442007@gmail.com

TEAM MEMBER : S Kavya & kaviyasubramani2102007@gmail.com

# Project Overview

* + **Purpose**: ThepurposeoftheRhythmicTunesMusicPlayeristoprovidea user-friendly application for playing music..
  + **Features**:Keyfeaturesinclude:musicplayback,playlistcreation,andsearch functionality.

# Architecture

* + **ComponentStructure**:Keyfeaturesinclude:musicplayback,playlist creation, and search functionality.
  + **StateManagement**:TheapplicationusestheContextAPIforglobalstate management to handle the current song, playback status, and user-created playlists.
  + **Routing**:ReactRouterisusedtomanagenavigationbetweendifferentviews, such as the home page and the playlist page

# SetupInstructions

* + **Prerequisites**:TherequiredsoftwaredependenciesareNode.jsand npm

,React.js.

* + **Installation**:Provideastep-by-stepguidetoClonetherepositoryfrom GitHub. Navigate to the client directory.
  + \*Run npm install to install dependencies.
  + \*Configureenvironment variablesfor anyAPIkeys

# FolderStructure

* + **Client**:ThemainReactapplicationisorganizedintofolderssuchas components, pages, and assets for images and audio files.
  + **Utilities**:Helperfunctionsformusicplaybackandcustomhooksformanaging audio state are located in the utils folder

# Code:

* + project-name/
  + ├── public/
  + │├── index.html
  + │└── favicon.ico
  + ├──src/
  + │├── assets/
  + ││├──images/
  + ││└──styles/
  + │├── components/
  + ││├──Header.js
  + ││├──Footer.js
  + ││└──...
  + │├── containers/
  + ││├──App.js
  + ││└──...
  + │├── actions/
  + ││├──index.js
  + ││└──...
  + │├── reducers/
  + ││├──index.js
  + ││└──...
  + │├── utils/
  + ││├──api.js
  + ││└──...
  + │├── index.js
  + │└── setupTests.js
  + ├── tests/
  + │├── components/
  + ││├──Header.test.js
  + ││└──...
  + │└── ...
  + ├──.git.ignore
  + ├──package.json
  + └── README.md

# RunningtheApplication

* + Providecommandstostartthefrontendserverlocally.
    - **Frontend**:npmstartintheclientdirectory.

# ComponentDocumentation

* + **KeyComponents**:Managesmusicplayback controls.Itreceivesasongprop.
  + \*PlaylistComponent:Displaysthelistofsongsandmanagesplaylist creation.
  + \*SearchComponent: Handlessearchingforsongs
  + **ReusableComponents**:ButtonComponent:Aconfigurablebutton component used throughout the application.
  + \*SongCardComponent:Acomponentfordisplayingindividualsongsin lists.

# StateManagement

* + **GlobalState**:TheContextAPIisusedtomanagetheglobalstate,ensuring that the current song and playback status are accessible to all components..
  + **LocalState**:seuseStatehookstomanagetheirownlocalstate,suchasform inputs or UI toggles

# UserInterface

* + ProvidescreenshotsorGIFsshowcasingdifferentUIfeatures,suchaspages, forms, or interactions.

# Styling

* **CSSFrameworks/Libraries**:DescribeanyCSSframeworks,libraries,orpre- processors (e.g., Sass, Styled-Components) used.
* **Theming**:Acustomdesignsystemwithlightanddarkthemesis implemented

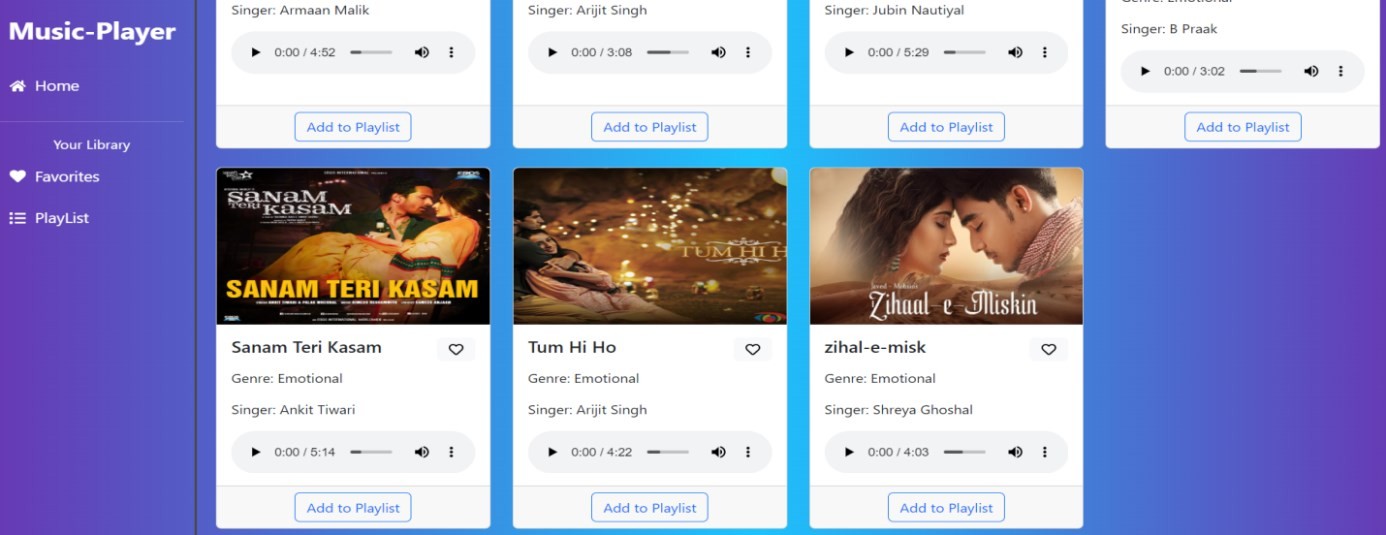
# Testing

* **TestingStrategy**:TheapplicationusesJestandReactTestingLibraryforunitand integration testing of components.

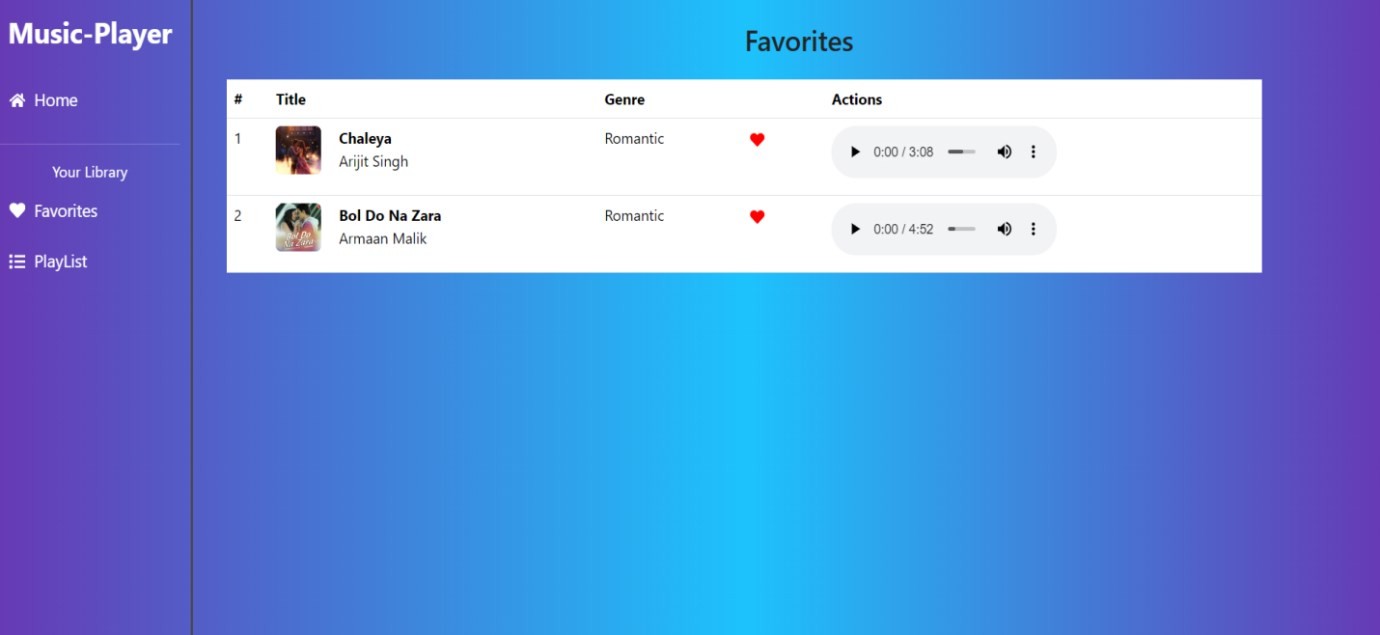
1. **CodeCoverage**:Codecoverageistrackedusingatooltoensureallkeycomponents are adequately tested.
2. **ScreenshotsorDemo:**

A screenshot of a playlist

AI-generated content may be incorrect.



**Favorites:**

****

1. **KnownIssues**
   * Anyknownbugsorissues,suchasoccasionalplaybackglitchesoncertainbrowsers, should be documented here

# FutureEnhancements

* + Potentialfuturefeaturescouldincludeauserauthenticationsystem,enhancedsearch functionality, and adding more animations to the user interface.

# DEMOLINK:

* + https://drive.google.com/file/d/1ZLRccyfc5sx9d75jYc\_B7eH12Tcd7QPq/view?usp=drive\_link