Frontend Development with React.js Project Documentation for Rhythmic Tunes

1. Introduction

- **ProjectTitle**: **Rhythmic Tunes**
- **TeamMembers**:

+ **Shree V(TeamLeader)** [EmailI:vshreecs22.krmmc@gmail.com]
+ **Dhanushree S** [EmailId:sdhanushreecs22.krmmc@gmail.com]
+ **Keerthiga R** [EmailId:rkeerthigacs22.krmmc@gmail.com]
+ **PrithikaNisha G** [EmailId:prithikanisha2003 @gmail.com]
+ **Lakshminarasimhan PM** [EmailId:pmlakshminarasimhancs22.krmmc@gmail.com]

2. ProjectOverview

- **Purpose**:
  Rhythmic Tunesisawebapplicationdesignedtoprovideuserswithaseamlessmusic listening experience. The application allows users to browse, search, and play music tracks, create playlists, and discover new music based on their preferences.
- **Features**:
- **SongListings:**Displayacomprehensivelistofavailablesongswithdetailssuchastitle artist, genre, and release date.
- **PlaylistCreation:**Empower users to create personalized playlists, adding and organizingsongs based on their preferences**.**
- **OfflineListening:**Allow users to download songs for offline listening, enhancing the app's accessibility and convenience.

3. Architecture

- **ComponentStructure**:
  TheapplicationisbuiltusingReact.jswithacomponent-basedarchitecture.Majorcomponents include:

- **Header**:Containsthenavigationbarandsearchbar.
- **Player**:Musicplayer controls(play,pause,volume,etc.).
- **Sidebar**:Displaysuserplaylistsandnavigationlinks.
- **HomePage**:Displaysfeaturedracks,recommendedplaylists,andnewreleases.
- **SearchPage**:Allowsuserstosearchforsongs,albums,andartists.
- **PlaylistPage**:Displaysuser-createdplaylistsandallowsplaylistmanagement.

- **StateManagement**:
  Theapplicationuses**Redux**forglobalstatemanagement.TheReduxstoremanagesuser authentication, current playing track, playlist data, and search results.

- **Routing**:
  Theapplicationuses**ReactRouter**fornavigation.Routesinclude:
  - /:Homepage
  - /search:Searchpage
  - /playlist/:id:Playlistdetailspage
  - /login:Userloginpage

- **Setup instructions**

  - **Prerequisites**:
    - React.js
    - Node.js(v16orhigher)
    - npm(v8orhigher)
    - Git
    - VsCode

  - **Installation**:
    1. Clonetherepository:gitclone https://github.com/dnks-1234/Music-App.gi
    2. CreateanewReatapp:npmcreate vite@latest
    3. Navigatetotheclientdirectory:cdproject-name
    4. Installdependencies:npminstall
    5. Configureenvironmentvariables:Createa.envfileintheclientdirectoryandaddthe necessary variables (e.g., API keys).
    6. Startthedevelopmentserver:npmrundev

4. FolderStructure

- **Client**:
    - **src/components:**#Reusablecomponents(Header,Player,etc.)
    - **src/pages:**#Pagecomponents(HomePage,SearchPage,etc.)
    - **src/assets:**#Images,icons,andotherstaticfiles
    - **src/redux:**#Reduxstore,actions,andreducers
    - **src/utils:**#Utilityfunctionsandhelpers
    - **App.js:**#Mainapplicationcomponent
    - **index.js:**#Entrypoint
- **Utilities**:
    - **api.js**:HandlesAPIrequeststothe backend.
    - **auth.js**:Managesuserauthenticationandtokenstorage.
    - **hooks/usePlayer.js**:Customhookformanagingthemusicplayerstate.

---

5. Runningthe Application

- **Frontend**:
    - Tostartthefrontendserver,runthefollowingcommandintheclientdirectory:
    - Npmi
    - Npxjson-server–watchdb.json
    - Npmrundev
    - Theapplicationwillbeavailablehttp://localhost:3000

---

6. ComponentDocumentation

- **KeyComponents**:
    - **Header**:Displaysthenavigationbarandsearchbar.
        - Props:onSearch(functiontohandlesearchqueries).
    - **Player**:Controlsthemusicplayback.
        - Props:currentTrack(objectcontainingtrack details), onPlay, onPause, onSkip.

- **PlaylistCard**:Displaysaplaylist withitsnameandcoverimage.
    - Props:playlist(objectcontainingplaylistdetails),onClick(function to handle playlist selection).
- **ReusableComponents**:
  - **Button**:Acustomizablebuttoncomponent.
    - Props: text,on Click,disabled.
  - **Input**:Areusableinputfieldforformsandsearch.
    - Props:type,placeholder,value,onChange.

---

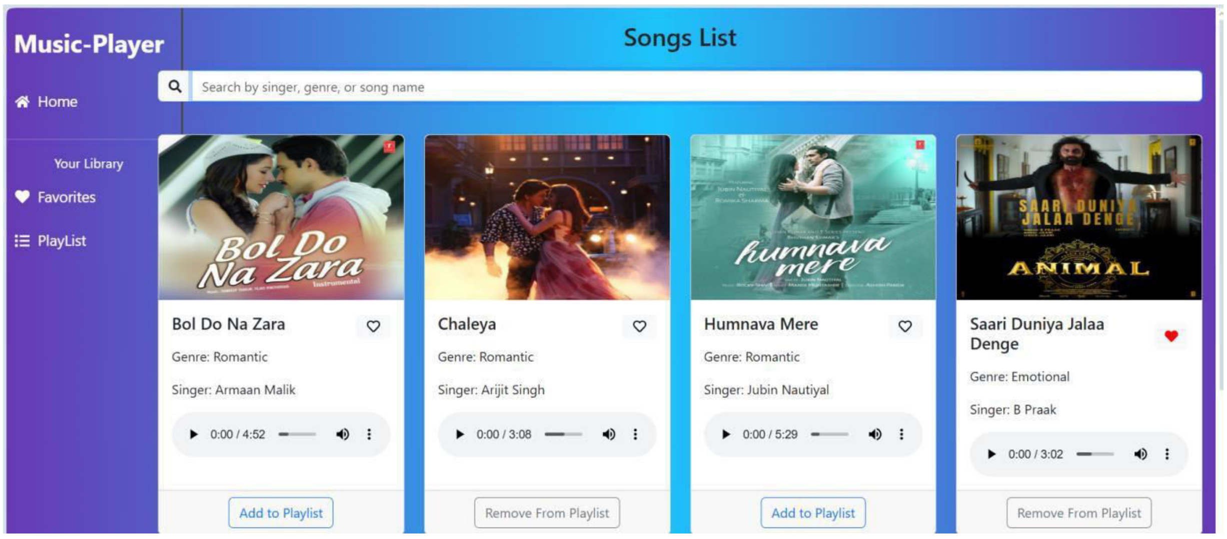
7. StateManagement

- **GlobalState**:
  TheReduxstoremanagesthefollowingglobalstates:
  - **User:**Currentauthenticateduser.
  - **Player:**Currentplayingtrack,playbackstatus(playing/paused),andvolume.
  - **Playlists:**User-createdplaylists.
  - **SearchResults:**Resultsfromthesearchfunctionality.
- **LocalState**:
  LocalstateismanagedusingReact'suseStatehookwithincomponents.Forexample,the SearchPage component manages the search query input locally.

---

8. UserInterface

- **Screenshots**
  - **HomePage:**Displayfeaturedtracksandrecommendedplaylists.



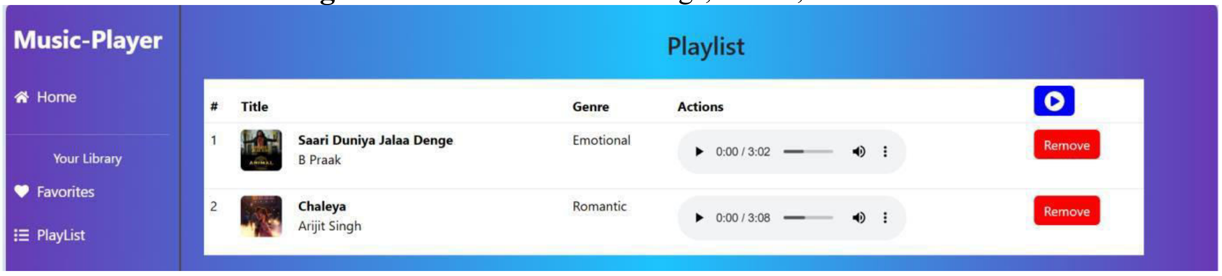
Bol Do Na Zara
Genre: Romantic
Singer: Armaan Malik
0:00 / 4:52

Chaleya
Genre: Romantic
Singer: Arijit Singh
0:00 / 3:08

Humnava Mere
Genre: Romantic
Singer: Jubin Nautiyal
0:00 / 5:29

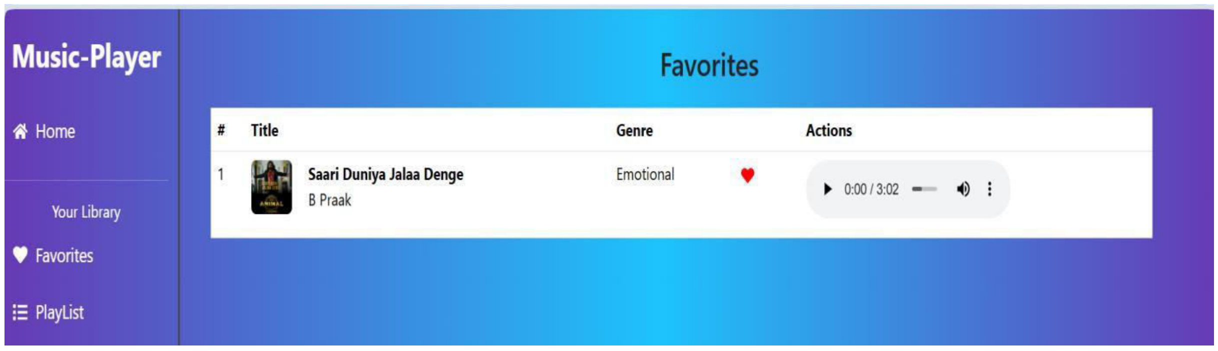
Saari Duniya Jalaa Denge
Genre: Emotional
Singer: B Praak
0:00 / 3:02

- **SearchPage:**Allowsuserstosearchforsongs,albums,andartists.



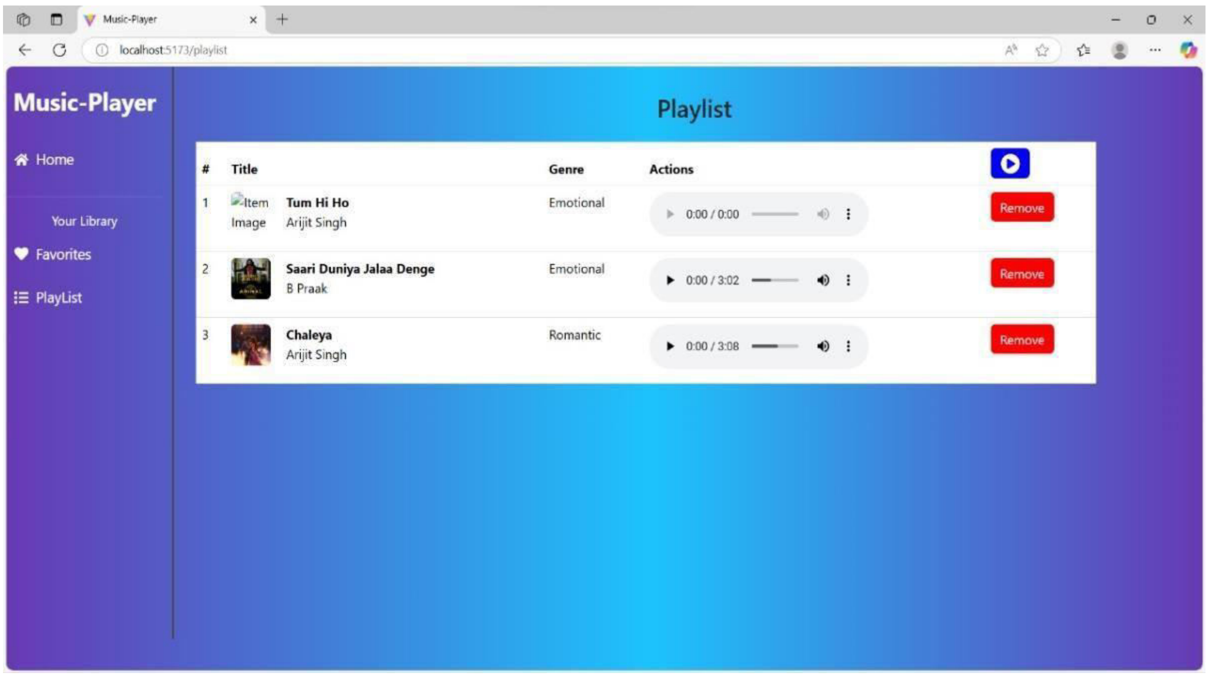
#	Title	Genre	Actions	
1	Saari Duniya Jalaa Denge B Praak	Emotional	0:00 / 3:02	Remove
2	Chaleya Arijit Singh	Romantic	0:00 / 3:08	Remove

- **FavoritesPage:**Displaysuser-selectedfavoritessongsandallowsplaylistmanagement.



#	Title	Genre	Actions
1	Saari Duniya Jalaa Denge B Praak	Emotional	0:00 / 3:02

- **playlistpage**:thatdisplayusers-createdplaylistandallowsplaylistmanagement



#	Title	Genre	Actions	
1	Item Image Tum Hi Ho Arijit Singh	Emotional	0:00 / 0:00	Remove
2	Saari Duniya Jalaa Denge B Praak	Emotional	0:00 / 3:02	Remove
3	Chaleya Arijit Singh	Romantic	0:00 / 3:08	Remove

---

9. Styling

- **CSSFrameworks/Libraries**:
- **HTML,CSS,andJavaScript:BasicknowledgeofHTMLforcreatingthestructureofyourapp,CSSforstyling,andJavaScriptforclient-sideinteractivityisessential.Theming**:

---

10. Testing

- **TestingStrategy**:
  - **UnitTesting:**Using**Jest**and**ReactTestingLibrary**
  - **IntegrationTesting**:Isperformedtoensurethatcomponentsworktogetheras expected.
  - **End-to-EndTesting:Cypress**isusedforend-to-endtestingofuserflows.
- **CodeCoverage**:
- **CodecoverageismonitoredusingJest'sbuiltincoveragetool.Thecurrentcoverageis85%.**

---

11. ScreenshotsorDemo

- **DemoLink:** https://drive.google.com/file/d/1HctrdM71F7G9VntHSUwu4UFbXqvEGiGK/view?usp=drive_link

- **Screenshots:**Seesection9forUIscreenshots.

12. KnownIssues

- **Issue1**:Themusicplayersometimesskipstracksunexpectedly.
- **Issue2**:Thesearchfunctionalityislowwithlargedatasets.

---

13. FutureEnhancements

- **FutureFeatures**:
  - Addsupportforuserprofilesandsocialsharing.
  - Implementarecommendationengineforpersonalizedmusicsuggestions.
  - Addanimationsandtransitionsforasmootheruserexperience.

Thisdocumentationprovidesacomprehensiveoverviewofthe**RhythmicTunes**project,including its architecture, setup instructions, and future plans.