**FrontendDevelopmentwithReact.js**

**ProjectDocumentationforRhythmicTunes**

* **Introduction**
* **Project Title**: 
* **TeamMembers**:

**hemachandran e** (**Team Leader**) chandranhema851@gmail.com

**hemanathan m hemanathanhemanathan077@gmail.com**

**siva k siva26062005@gmail.com**

**chandravel v s chandru2004hc@gmail.com**

**baskar p bb6709708@gmail.com**

* **ProjectOverview**
* **Purpose**: RhythmicTunesisawebapplicationdesignedtoprovideuserswithaseamlessmusic listeningexperience.Theapplicationallowsusers tobrowse,search,andplaymusic tracks,createplaylists,anddiscovernewmusicbasedontheirpreferences.
* **Features**:
* Musicplayerwithplay,pause,skip,andvolumecontrol.
* Searchfunctionalitytofindsongs,albums,andartists.
* Userauthentication(login/signup).
* Playlistcreationandmanagement.
* Responsivedesignformobileanddesktop.
* **Architecture**
* **ComponentStructure**:

TheapplicationisbuiltusingReact.jswithacomponent-basedarchitecture.Major components include:

* **Header**:Containsthenavigationbarandsearchbar.
* **Player**:Musicplayercontrols(play,pause,volume,etc.).
* **Sidebar**:Displaysuserplaylistsandnavigationlinks.
* **HomePage**:Displaysfeaturedtracks,recommendedplaylists,andnew releases.
* **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**:
* Node.js(v16orhigher)
* npm(v8orhigher)
* Git
* **Installation**:
* Clonetherepository:gitclone
* <https://github.com/pappu-0211/music-streaming-app.git>
* Installdependencies:npminstall
* Configureenvironmentvariables:Createa.envfileintheclientdirectoryand add the necessary variables (e.g., API keys).
* Startthedevelopmentserver:npmstart
* **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**:HandlesAPIrequeststothebackend.
* **auth.js**:Managesuserauthenticationandtokenstorage.
* **hooks/usePlayer.js**:Customhookformanagingthemusicplayerstate.
* **RunningtheApplication Frontend**:
* Tostartthefrontendserver,runthefollowingcommandintheclientdirectory: npm start
* npminstall
* npxjson-server./db/db.json
* npmrundev
* Theapplicationwillbeavailableat[http://localhost:3000](http://localhost:3000/)
* **ComponentDocumentation**
* **KeyComponents**:
* **Header**:Displaysthenavigationbarandsearchbar.
* Props:onSearch(functiontohandlesearchqueries).
* **Player**:Controlsthemusicplayback.
* Props:currentTrack(objectcontainingtrack details), onPlay, onPause, onSkip.
* **PlaylistCard**:Displaysaplaylistwithitsnameandcoverimage.
* Props:playlist(objectcontainingplaylistdetails),onClick(functionto handle playlist selection).
* **ReusableComponents**:
* **Button**:Acustomizablebuttoncomponent.
* Props:text,onClick,disabled.
* **Input**:Areusableinputfieldforformsandsearch.
* Props:type,placeholder,value,onChange.
* **StateManagement**
* **GlobalState**: TheReduxstoremanagesthefollowingglobalstates:
* **user:**Currentauthenticateduser.
* **player:**Currentplayingtrack,playbackstatus(playing/paused),andvolume.
* **playlists:**User-createdplaylists.
* **searchResults:**Resultsfromthesearchfunctionality.
* **Local State**: LocalstateismanagedusingReact'suseStatehookwithincomponents.Forexample, the SearchPage component manages the search query input locally.
* **UserInterface**
* **Screenshots**
* **HomePage:**Displayfeaturedtracksandrecommendedplaylists.



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



* **PlaylistPage:**Displaysuser-createdplaylistsandallowsplaylistmanagement.



* **Styling**
* **CSS Frameworks/Libraries**:

Theapplicationuses**Styled-Components**forstyling.Thisallowsformodularand scoped CSS within components.

* **Theming**:

AcustomthemeisimplementedusingStyled-Components,withsupportforlightand dark modes.

* **Testing**
* **TestingStrategy**:
* **UnitTesting:**Using**Jest**and**ReactTestingLibrary**.
* **IntegrationTesting**:Isperformedtoensurethatcomponentsworktogetheras expected.
* **End-to-EndTesting:Cypress**isusedforend-to-endtestingofuserflows.
* **CodeCoverage**:
* CodecoverageismonitoredusingJest’sbuiltincoveragetool.Thecurrent coverage is 85%.
* **ScreenshotsorDemo**
* **Demo Link:** [**https://github.com/unm115u22cs138/Rythmic-**https://github.com/Hema133723/rythemic-tunes1/commit/6d8028af0625c26a9a5f281d6be1a6bee81ff62a](https://github.com/unm115u22cs138/Rythmic-tunes/commit/9ea9ad641eaca25607e3bc3ab5c0444a9ba9024b)

**Screenshots:Seesection9forUIscreenshots.**

* **KnownIssues**
* **Issue1**:Themusicplayersometimesskipstracksunexpectedly.
* **Issue2**:Thesearchfunctionalityisslowwithlargedatasets.
* **FutureEnhancements**
* **FutureFeatures**:
* Addsupportforuserprofilesandsocialsharing.
* Implementarecommendationengineforpersonalizedmusicsuggestions.
* Addanimationsandtransitionsforasmootheruserexperience.

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