

**RYTHIMIC TUNES(REACT)**

(Music Streaming Application)

**Team ID: SWTID1741423241154402**

**Team Leader:**

Kamini .K   
[kaminitamil17@gmail.com](mailto:kaminitamil17@gmail.com)

**Team Members :**

Kaviya .K   
[kkaviya6862@gmail.com](mailto:kkaviya6862@gmail.com)

Nivetha .M   
[nivethamani714@gmail.com](mailto:nivethamani714@gmail.com)

Thenmozhi .R   
[thenmozhiranganathan30@gmail.com](mailto:thenmozhiranganathan30@gmail.com)

Arun kumar .M

[arunvijay7743@gmail.com](mailto:arunvijay7743@gmail.com)

**Introduction:-**

Welcome to the future of musical indulgence – anunparalleledaudioexperienceawaits you with our cutting-edge Music Streaming Application, meticulously crafted using the power of React.js. Seamlessly blending innovation with user-centric design, our application is set to redefinehowyouinteractwithandimmerseyourselfintheworldof music.

Designed for the modern music enthusiast, our React-based Music Streaming Application offers a harmonious fusion of robust functionality and an intuitive user interface. From discovering the latest chart-toppers to rediscovering timeless classics, our platform ensures an all-encompassing musical journey tailored to your unique taste.

The heart of our Music Streaming Application lies in React, a dynamic and feature-rich JavaScript library. Immerse yourself in a visually stunning and interactive interface, where every click, scroll, and playlist creation feels like a musical revelation. Whether you're on a desktop, tablet, or smartphone, our responsive design ensures a consistent and enjoyable experience across all devices.

Say goodbye to the limitations of traditional music listening and welcome a world of possibilities with our React-based Music Streaming Application. Join us on this journeyas we transform the way you connect with and savor the universal language of music. Get ready to elevate your auditory experience – it's time to press play on a new era of music streaming.

**Scenario-BasedIntro:-**

Imaginesteppingontoabustlingcitystreet,thesoundsofcarshonking,peoplechatting, and street performers playing in the background. You're on your way to work, and you need a little something to elevate your mood. You pull out your phone and open your favorite music streaming app, "RythimicTunes.

Withjustafewtaps,you'retransportedtoaworldofmusictailoredtoyourtastes.As you walk, the app’s smart playlist kicks in, starting with an upbeat pop song that gets your feet tapping. As you board the train, the music shifts to a relaxing indie track, perfectly matching your need to unwind during the commute.

**TargetAudience:-**

MusicStreamingisdesignedforadiverseaudience,including:

●**MusicEnthusiasts:**PeoplepassionateaboutenjoyingandlisteningMusicThrough out

there free time to relax themselves.

**ProjectGoalsandObjectives:-**

The primary goal of Music Streaming is to provide a seamless platform for music

enthusiasts, enjoying, and sharing diverse musical experiences. Our objectives include:

**User-Friendly Interface:** Develop an intuitive interface that allows users to effortlessly

explore, save, and share their favorite music tracks and playlists.

**Comprehensive Music Streaming:** Provide robust features for organizing and managing

music content, including advanced search options for easy discovery.

**Modern Tech Stack:** Harness cutting-edge web development technologies, such as

React.js, to ensure an efficient and enjoyable user experience while navigating and

interacting with the music streaming application.

**KeyFeatures:-**

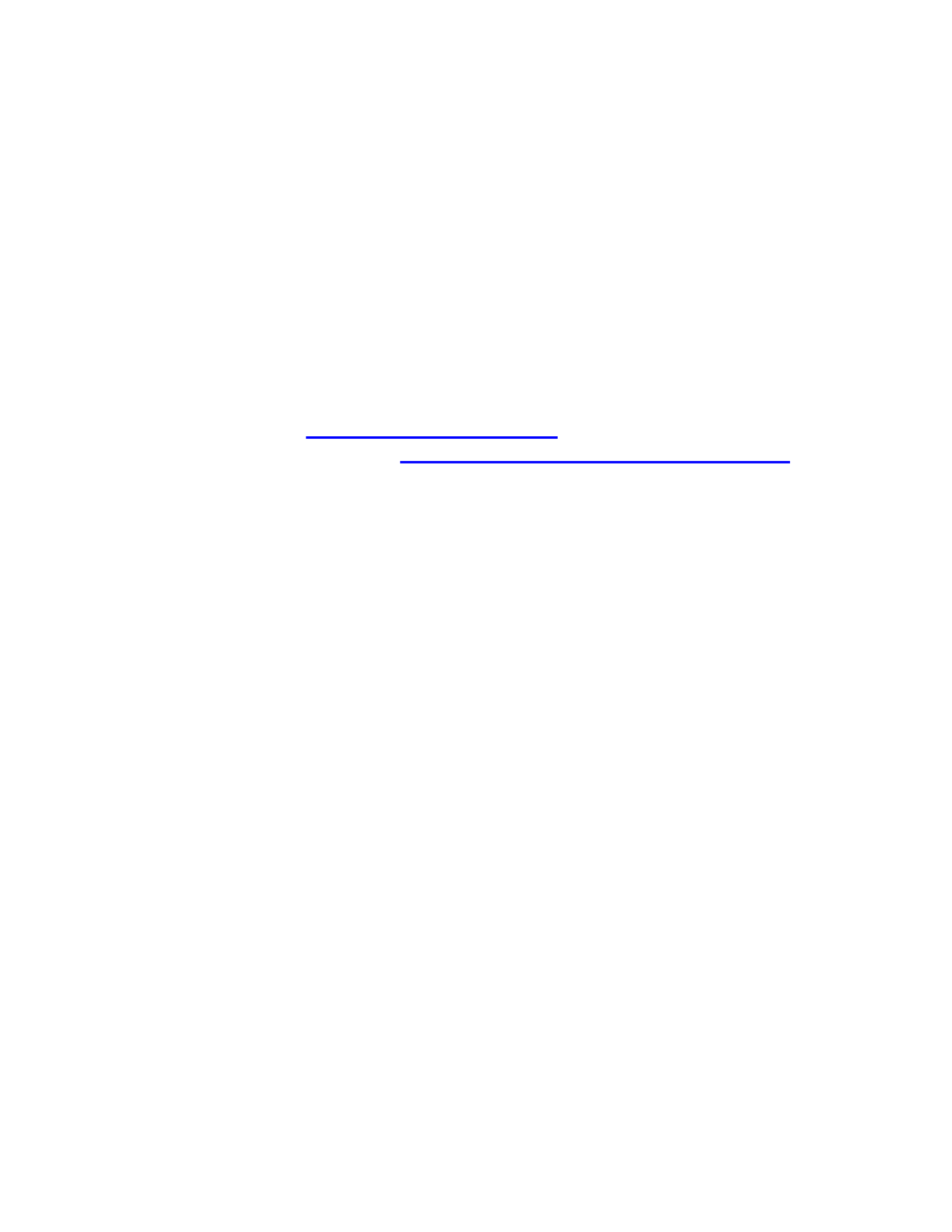
? **Song Listings:** Display a comprehensive list of available songswithdetailssuchas title, artist, genre, and release date.

? **PlaylistCreation:**Empoweruserstocreatepersonalizedplaylists,addingand organizing songs based on their preferences.

? **Playback Control:**Implement seamless playback control features, allowing users to play, pause, skip, and adjust volume during music playback.

? **Offline Listening:** Allow users to download songs for offline listening, enhancing the app's accessibility and convenience.

? **Search Functionality:** Implement a robust search feature for users to easily find specific songs, artists, or albums within the app.



**PRE-REQUISITES**:-

HerearethekeyprerequisitesfordevelopingafrontendapplicationusingReact.js:

? **Node.jsandnpm**:

Node.js is a powerful JavaScript runtime environment that allows you to run JavaScript code on the local environment. It provides a scalable and efficient platform for building network applications.

InstallNode.jsandnpmonyourdevelopmentmachine,astheyarerequiredto run JavaScript on the server-side.

●Download[:https://nodejs.org/en/download/](https://nodejs.org/en/download/)  
 ●Installationinstructions[:https://nodejs.org/en/download/package-manager/](https://nodejs.org/en/download/package-manager/) ? **React.js**:

React.js is a popular JavaScript library for building user interfaces. It enables developers to create interactive and reusable UI components, making iteasierto build dynamic and responsive web applications.

InstallReact.js,aJavaScriptlibraryforbuildinguserinterfaces.

●CreateanewReactapp:

npmcreatevite@latest

Enter and then type project-name and select preferred frameworks and then enter

●Navigatetotheprojectdirectory:

cdproject-name

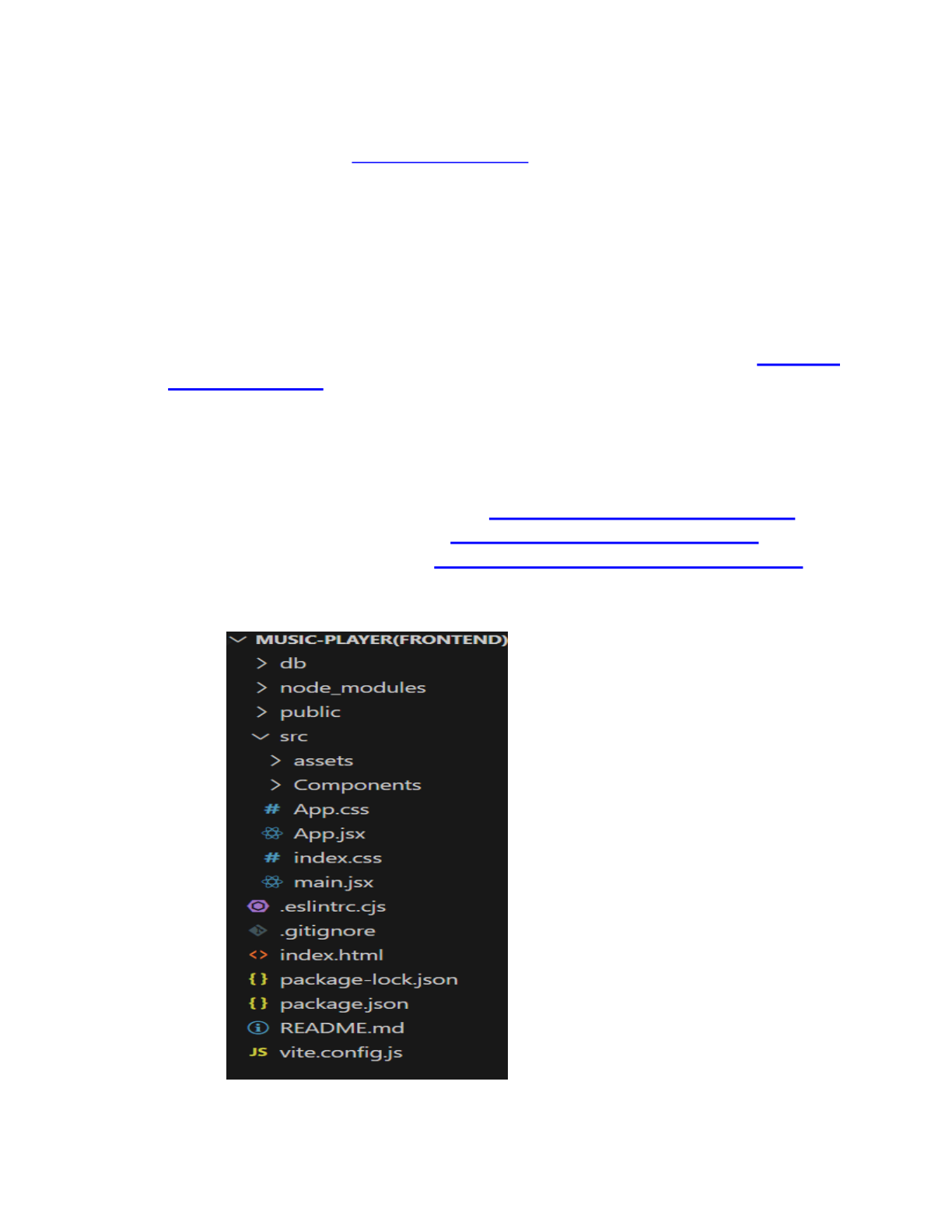
npm install

●RunningtheReactApp:

WiththeReactappcreated,youcannow start the development server and see your React application in action.

●Startthedevelopmentserver:

npmrundev



Thiscommandlaunchesthedevelopmentserver,andyoucanaccessyourRe

actappat[http://localhost:5173i](http://localhost:5173/)nyourwebbrowser.

?**HTML, CSS,andJavaScript**:BasicknowledgeofHTMLforcreatingthestructureof your app, CSS for styling, and JavaScript for client-side interactivity is essential.

?**Version Control**: Use Git for version control, enabling collaboration and tracking changes throughout the development process. PlatformslikeGitHuborBitbucket can host your repository.

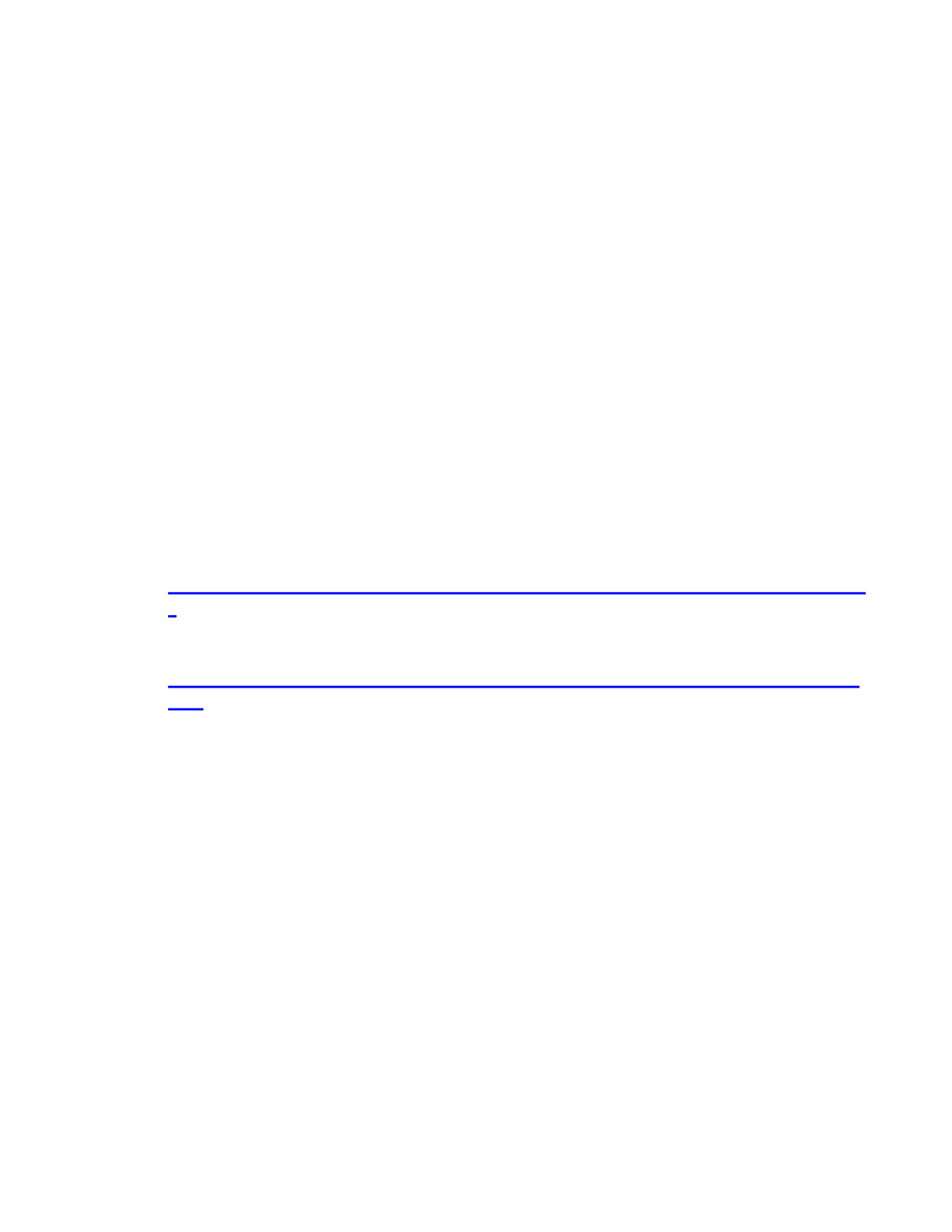
|  |  |  |
| --- | --- | --- |
| • | Git:Downloadandinstallationinstructionscanbefoundat: | https://git- |

[scm.com/downloads](https://git-scm.com/downloads)

?**Development Environment**: Choose a code editor or Integrated Development Environment (IDE) that suits your preferences, such as Visual Studio Code, Sublime Text, or WebStorm.

•VisualStudioCode:Downloadfrom<https://code.visualstudio.com/download>•SublimeText:Downloadfrom<https://www.sublimetext.com/download>•WebStorm:Downloadfrom<https://www.jetbrains.com/webstorm/download>

**Projectstructure:**



The project structure may vary depending on the specific library, framework, programming language, or development approach used. It's essential to organize the filesanddirectoriesinalogicalandconsistentmannertoimprovecodemaintainability and collaboration among developers.

app/app.component.css, src/app/app.component: These files are part of the main

AppComponent,whichservesastherootcomponentfortheReactapp.Thecomponent

handles the overall layout and includes the router outlet for loading different

components based on the current route.

**PROJECTFLOW:-**

**Projectdemo:**

Beforestartingtoworkonthisproject,let’sseethedemo.

Demolink:   
[https://drive.google.com/file/d/1zZuq62lyYNV\_k5uu0SFjoWa35UgQ4LA9/view?usp=drive\_lin k](https://drive.google.com/file/d/1zZuq62lyYNV_k5uu0SFjoWa35UgQ4LA9/view?usp=drive_link)

Usethecode in:

[https://drive.google.com/drive/folders/1BkYWfW\_K3ek\_UgtXNTAsDqlhdCuqz6nT?usp=drive \_link](https://drive.google.com/drive/folders/1BkYWfW_K3ek_UgtXNTAsDqlhdCuqz6nT?usp=drive_link)

**Milestone1:ProjectSetupandConfiguration:**

**1.Installrequiredtoolsandsoftware:**

●**Installationofrequiredtools**:

1.Opentheprojectfoldertoinstallnecessarytools In

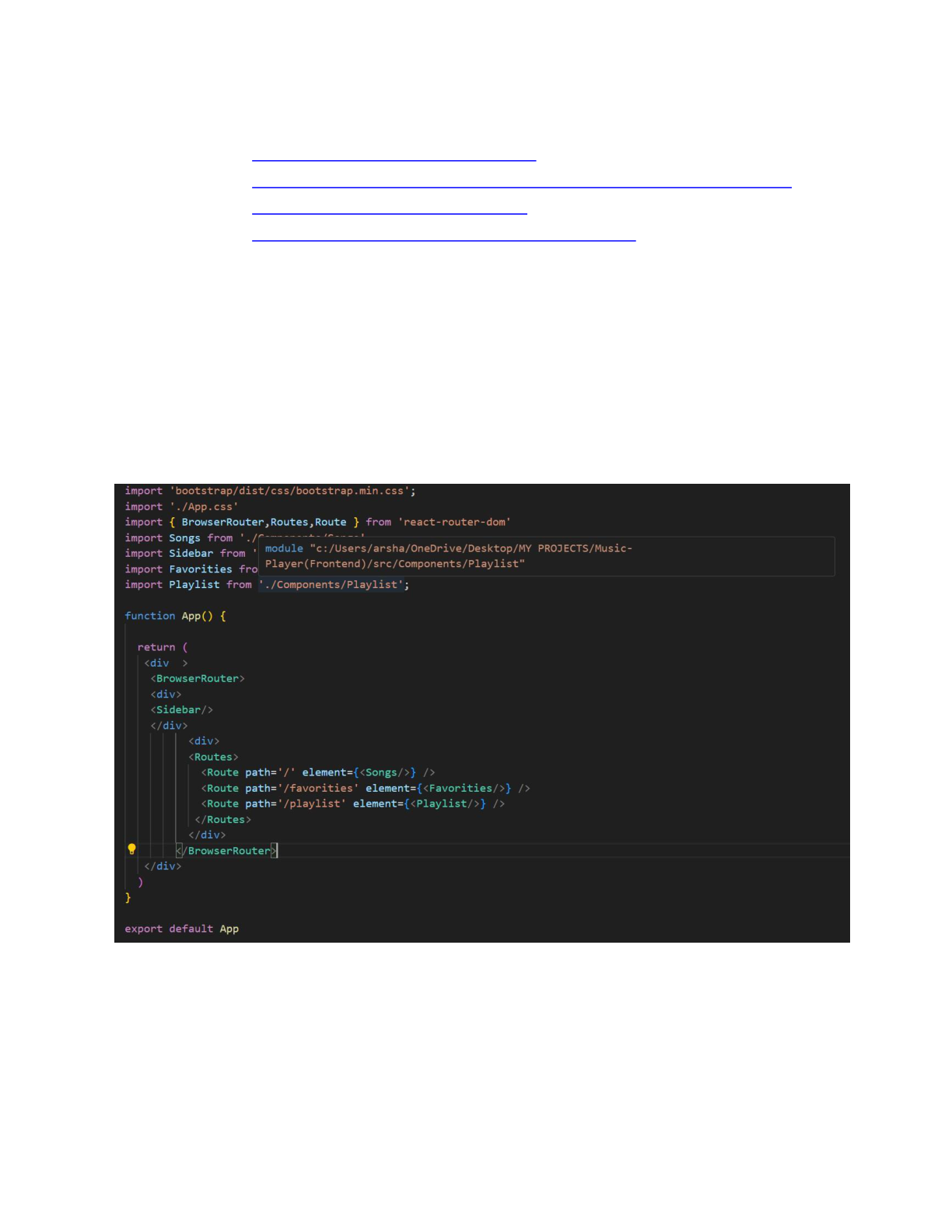
this project, we use:

oReactJs   
oReactRouterDom

oReactIcons

oBootstrap/tailwindcss

oAxios



●Forfurtherreference,usethefollowingresources   
o<https://react.dev/learn/installation>  
o<https://react-bootstrap-v4.netlify.app/getting-started/introduction/>o<https://axios-http.com/docs/intro>  
o<https://reactrouter.com/en/main/start/tutorial>

**Milestone2:ProjectDevelopment:**

**1.SetupReactApplication:**   
•CreateReactapplication.   
•ConfigureRouting.

•Installrequiredlibraries.

Setting Up Routes:-

**CodeDescription:-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ●Imports | Bootstrap | CSS | (bootstrap/dist/css/bootstrap.min.css) | for | styling |

components.

●ImportscustomCSS(./App.css)foradditionalstyling.

●Imports BrowserRouter, Routes, and Route from react-router-dom for setting up client-side routing in the application.

●Defines the App functional component that serves as the root component ofthe application.

●UsesBrowserRouterastheroutercontainertoenableroutingfunctionality.

●Includesadivastherootcontainerfortheapplication.

●WithinBrowserRouter,wrapscomponentsinsidetwodivcontainers:   
 oThe first div contains the Sidebar component, likely serving navigation or additional content.

oThe second div contains the Routes component from React Router, which handles rendering components based on the current route.

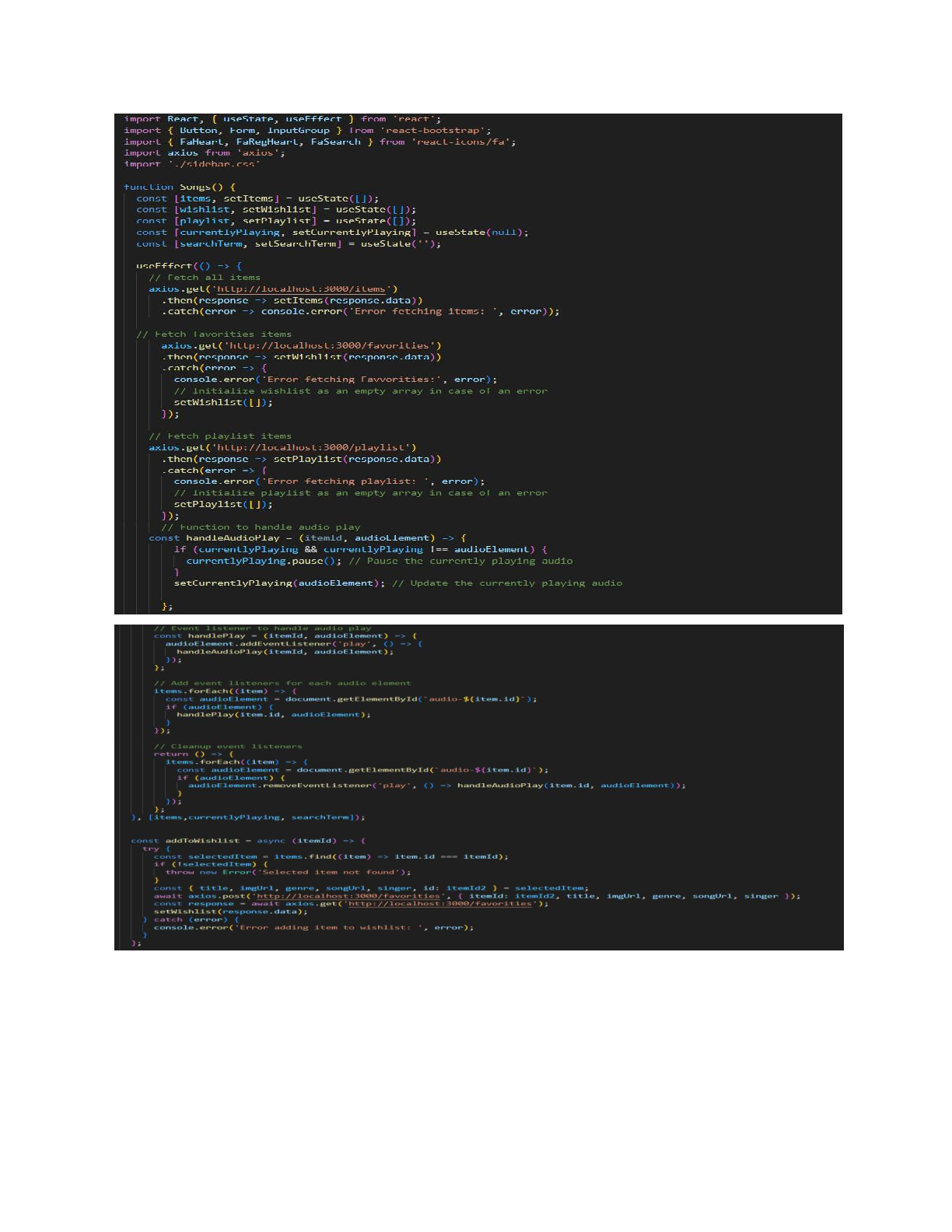
oInsideRoutes,definesseveralRoutecomponents:   
 oRoutewithpath='/'renderstheSongscomponent when the root path is accessed (/).

oRoutewithpath='/favorities'renderstheFavoritiescomponent when the /favorities path is accessed.

oRoutewithpath='/playlist'rendersthePlaylistcomponentwhenthe /playlistpathisaccessed.

●Exports the App component as the default export, making it available for use in other parts of the application.

**FetchingSongs:-**



**CodeDescription:-**

●**useState:**

oitems:Holdsanarrayofallitemsfetchedfrom

[http://localhost:3000/items.](http://localhost:3000/items)

owishlist:Storesitemsmarkedasfavoritesfetchedfrom

[http://localhost:3000/favorities.](http://localhost:3000/favorities)

oplaylist:Storesitemsaddedtotheplaylistfetched from

[http://localhost:3000/playlist.](http://localhost:3000/playlist)

ocurrentlyPlaying:Keepstrackofthecurrentlyplayingaudioelement. osearchTerm:Storesthecurrentsearchtermenteredbytheuser.

●**DataFetching:**   
 oUsesuseEffecttofetchdata:

▪ Fetchesallitems(items)from[http://localhost:3000/items.](http://localhost:3000/items)

▪ Fetchesfavoriteitems(wishlist)from

[http://localhost:3000/favorities.](http://localhost:3000/favorities)

▪ Fetchesplaylistitems(playlist)from

[http://localhost:3000/playlist.](http://localhost:3000/playlist)

oSetsstatevariables(items,wishlist,playlist)basedonthefetched data.

●**AudioPlayback Management:**   
 oSetsupaudioplayeventlistenersandcleanupforeachitem:

▪ handleAudioPlay: Manages audio playback by pausing the currently

playing audio when a new one starts.

▪ handlePlay:Addseventlistenerstoeachaudioelementtotrigger

handleAudioPlay.

oEnsuresthatonlyoneaudioelementplaysata time by pausing others when a new one starts playing.

●**addToWishlist(itemId):**

oAddsanitemtothewishlist(favorities)bymakingaPOSTrequestto

[http://localhost:3000/favorities.](http://localhost:3000/favorities)

oUpdatesthewishliststateafteraddinganitem.

●**removeFromWishlist(itemId):**

oRemoves an item from the wishlist (favorities) by makingaDELETErequest

to [http://localhost:3000/favorities/{itemId}.](http://localhost:3000/favorities/)

oUpdatesthewishliststateafterremovinganitem.

●**isItemInWishlist(itemId):**

oChecksifanitemexistsinthewishlist(favorities)basedonitsitemId.

●**addToPlaylist(itemId):**

oAddsanitemtotheplaylist(playlist)bymakingaPOSTrequestto

[http://localhost:3000/playlist.](http://localhost:3000/playlist)

oUpdatestheplayliststateafteraddinganitem.

●**removeFromPlaylist(itemId):**

oRemovesanitemfromtheplaylist(playlist)bymakingaDELETErequestto

[http://localhost:3000/playlist/{itemId}.](http://localhost:3000/playlist/)

oUpdatestheplayliststateafterremovinganitem.

●**isItemInPlaylist(itemId):**

oChecksifanitemexistsintheplaylist(playlist)basedonitsitemId.

●**filteredItems:**   
 oFiltersitemsbasedonthesearchTerm.

oMatches title, singer, or genrewith thelowercaseversionof

searchTerm.

●**JSX:**   
 oRenders aformwithaninputfield(Form,InputGroup,Button,FaSearch) for searching items.

oMapsoverfilteredItemstorendereachitemintheUI.

oIncludesbuttons(FaHeart,FaRegHeart)toadd/removeitemsfrom

wishlistand playlist.

oRendersaudioelementsforeachitemwithplay/pausefunctionality.

●**ErrorHandling:**   
 oCatchesandlogserrorsduringdatafetching(axios.get).

oHandleserrorswhenadding/removingitemsfromwishlistandplaylist.

**FrontendCodeForDisplayingSongs:-**



**CodeDescription:-**

●**ContainerSetup:**

oUses a divwith inline styles (style={{display:"flex",

justifyContent:"flex-end"}})toalignthecontenttotheright. oThe main container (songs-container) has a fixed width   
 (width:"1300px")andcontainsalltheUIelementsrelatedtosongs.

●**Header:**   
 oDisplays a heading (<h2>) with text "Songs List" centered

(className="text-3xlfont-semiboldmb-4text-center").

●**SearchInput:**   
 oUtilizesInputGroupfromReactBootstrapforthesearchfunctionality. oIncludesaninputfield(Form.Control)thatallowsuserstosearchbysinger, genre, or song name.

oBindstheinputfieldvaluetosearchTermstate(value={searchTerm}) and updates it on change (onChange={(e)

=>setSearchTerm(e.target.value)}). oStyled with className="search-input".

●**CardLayout:**   
 oUses Bootstrap grid classes (row, col) to create a responsive card layout

(className="rowrow-cols-1row-cols-md-2row-cols-lg-3 row-cols-xl-4 g-4").

oMapsoverfilteredItemsarrayandrenderseachitemasaBootstrapcard

(<div className="card h-100">).

●**CardContent:**   
 oDisplaystheitem'simage(<img>),title(<h5

className="card-title">),genre(<pclassName="card-text">), and singer (<p className="card-text">).

oIncludesanaudioplayer(<audiocontrolsclassName="w-100"

id={audio-${item.id}}>) for playing the song with a source (<source

src={item.songUrl} />).

●**WishlistandPlaylistButtons:**   
 oAddsahearticonbutton(<Button>)toaddorremoveitemsfromthewishlist (isItemInWishlist(item.id)determines which button to show). oIncludesan"AddtoPlaylist"or"RemoveFromPlaylist"button(<Button>) based on whether the item is already in the playlist

(isItemInPlaylist(item.id)).

●**ButtonClickHandlers:**   
 oHandles adding/removing items from the wishlist

(addToWishlist(item.id),removeFromWishlist(item.id)).

oManages adding/removing items from the playlist

(addToPlaylist(item.id),removeFromPlaylist(item.id)).

●**CardStyling:**   
 oAppliesBootstrapclasses(card,card-body,card-footer)forstylingthe card components.

oUsescustomstyles(rounded-top,w-100)forspecificelementslikeimages and audio players.

**ProjectExecution:**

Aftercompletingthecode,runthereactapplicationbyusingthecommand“npmstart”or “npm run dev” if you are using vite.js

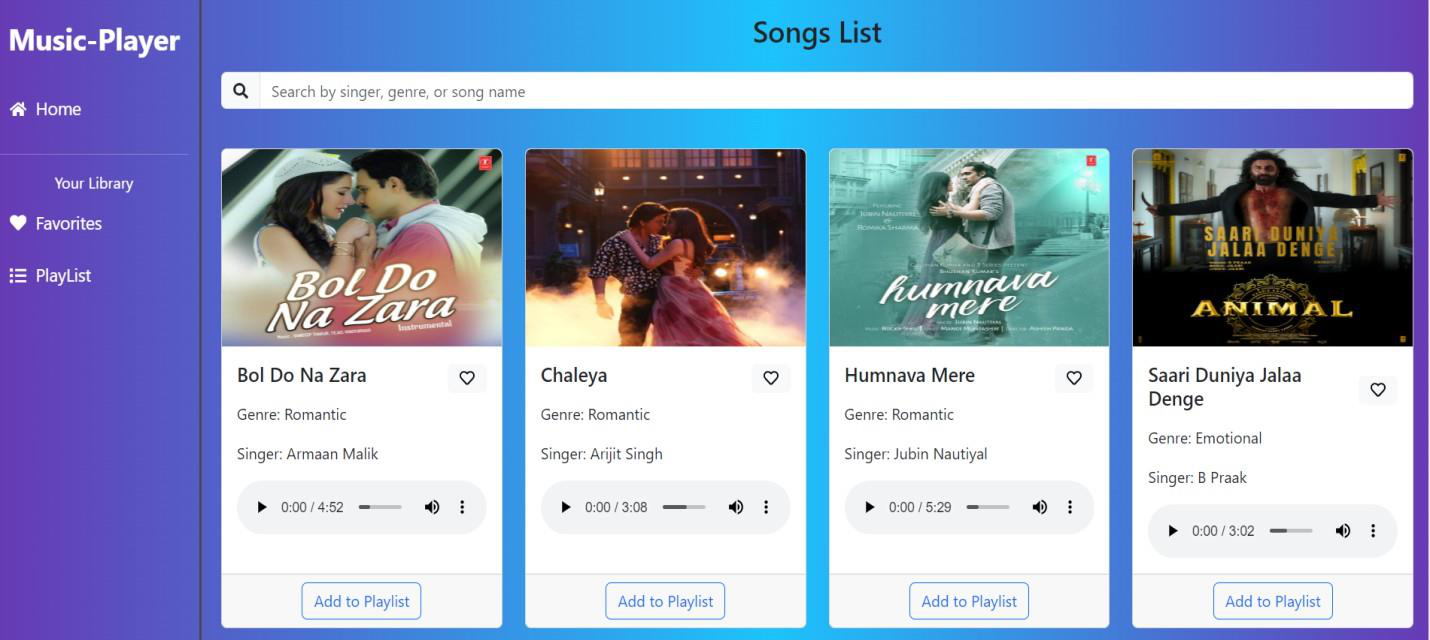
And the Open new Terminal type this command “json-server --watch./db/db.json” to start

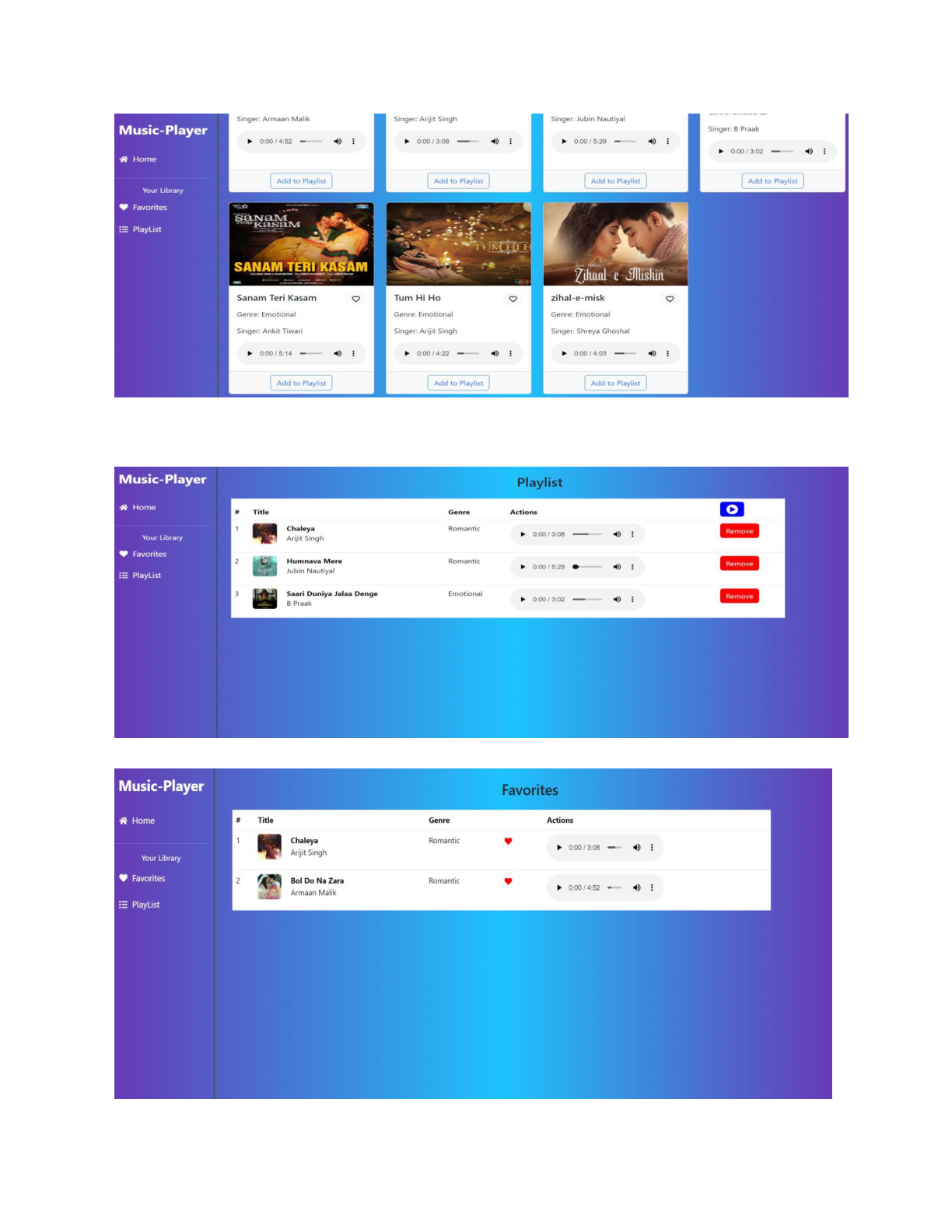
the json server too.

AfterthatlaunchtheRythimicTunes.

Herearesomeofthescreenshotsoftheapplication.

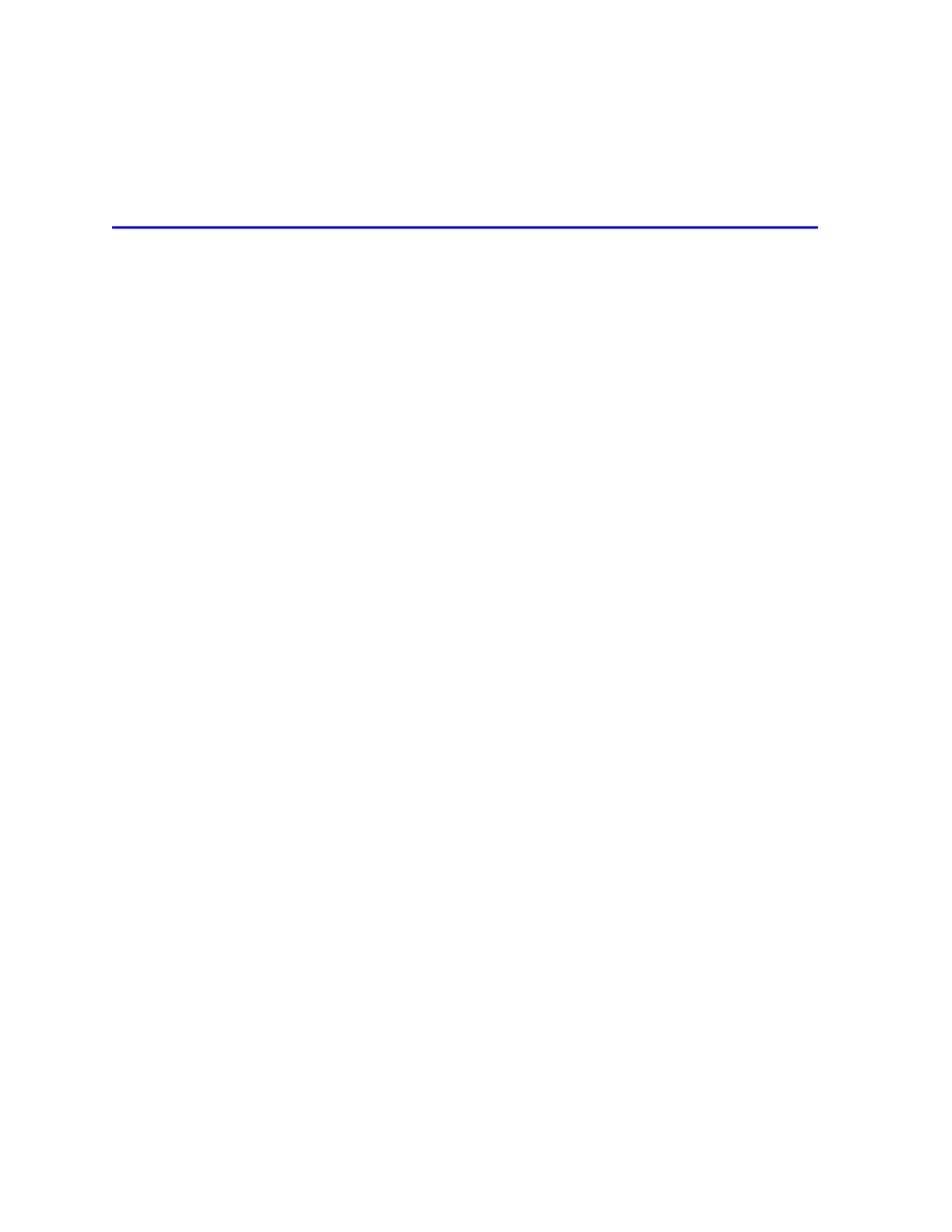
**? Herocomponents**





**?Playlist**

**?Favorites**



**ProjectDemolink:**

<https://drive.google.com/file/d/1zZuq62lyYNV_k5uu0SFjoWa35UgQ4LA9/view?usp=drive_link>

**\*\*\* Happy coding!! \*\*\***