

# PaperPass免费版检测报告

## 简明打印版

比对结果(相似度):

总体: 10% (总体相似度是指本地库、互联网的综合对比结果)  
本地库: 10% (本地库相似度是指论文与学术期刊、学位论文、会议论文、大学生联合比对库的比对结果)  
期刊库: 4% (期刊库相似度是指论文与学术期刊库的比对结果)  
学位库: 6% (学位库相似度是指论文与学位论文库的比对结果)  
会议库: 0% (会议库相似度是指论文与会议论文库的比对结果)  
图书库: 2% (图书库相似度是指论文与图书库的比对结果)  
联合库: 4% (联合库相似度是指论文与大学生联合比对库的比对结果)

报告编号: 645A363F836C005Q4

检测版本: 免费版

论文题目: 网页版音乐播放器毕业设计

论文作者: 佚名

论文字数: 16143

段落个数: 277

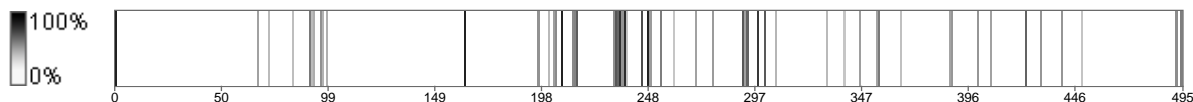
句子个数: 499

提交时间: 2023-5-9 20:02:07

比对范围: 学术期刊、学位论文、会议论文、书籍数据、大学生联合比对库

查询真伪: <https://www.paperpass.com/check>

句子相似度分布图:



本地库相似资源列表(学术期刊、学位论文、会议论文、书籍数据、大学生联合比对库):

- 相似度: 4.2%  
来源: 大学生联合比对库
- 相似度: 0.5% 篇名: 《Web前端性能问题研究》  
来源: 学术期刊 电脑编程技巧与维护 2016年2期

互联网相似资源列表:

免费版不检测互联网资源库

92% **网页版音乐播放器设计实现**

Design and Implementation of Web Music Player

|       |          |    |              |
|-------|----------|----|--------------|
| 学生姓名  | 李志立      | 学号 | 201911921718 |
| 所在学院  | 数学与计算机学院 | 班级 | 数软 1198      |
| 所在专业  | 信息与计算科学  |    |              |
| 申请学位  | 理学学士     |    |              |
| 指导教师  | 黄小玲      | 职称 | 副教授          |
| 副指导教师 | 叶国栋      | 职称 | 教授           |
| 答辩时间  | 年 月 日    |    |              |

## 目 录

|                    |    |
|--------------------|----|
| 设计总说明 .....        | I  |
| introduction.....  | II |
| 1 概述 .....         | 1  |
| 1.1 设计的目的和意义.....  | 1  |
| 1.2 开发的目的是意义.....  | 1  |
| 2 开发的技术与工具简介 ..... | 2  |
| 2.1 开发的技术.....     | 2  |
| 2.1.1 前端开发 .....   | 2  |
| 2.1.2 后端开发 .....   | 3  |
| 2.2 开发工具.....      | 3  |
| 2.3 开发环境.....      | 3  |
| 3 系统分析 .....       | 4  |
| 3.1 可行性分析.....     | 4  |
| 3.2 需求分析.....      | 4  |
| 3.3 功能及模块分析.....   | 4  |
| 4 系统设计 .....       | 5  |
| 4.1 程序设计.....      | 5  |
| 4.2 系统结构设计.....    | 5  |
| 5 系统实现 .....       | 7  |
| 5.1 登录注册.....      | 7  |
| 5.2 搜索.....        | 8  |
| 5.3 发现音乐.....      | 11 |
| 5.3.1 排行榜 .....    | 11 |
| 5.3.2 歌单 .....     | 12 |
| 5.3.3 歌手 .....     | 14 |
| 5.3.4 最新音乐 .....   | 15 |
| 5.4 视频.....        | 15 |
| 5.5 关注.....        | 16 |
| 5.6 私人 FM.....     | 17 |
| 5.7 底部音乐的控制.....   | 18 |
| 5.8 用户主页信息.....    | 19 |
| 6 系统测试 .....       | 20 |
| 7 总结 .....         | 21 |

---

|            |    |
|------------|----|
| 鸣 谢 .....  | 21 |
| 参考文献 ..... | 22 |

## 设计总说明

在科技的发展人类文明的进步下，人们都在为获得更好生活努力拼搏着，在生活上难免会有各种各样的压力，从而需要一些解压的方式。听音乐成为了人们的日常生活放松身心的其中一种方式，如今音乐的发展也很迅速，在科技趋势下很多为了方便用户听歌，诞生了手机端以及电脑版的音乐播放器。本设计是为了更好的满足用户需要简洁方便的音乐播放器需求而张开论述。

本设计以网页版音乐播放器为案例，<sup>51%</sup>根据对 web 音乐播放器用户市场的需求分析，本设计以未登录和登录状态的用户进行不同页面的展示相互结合，更有利于方便用户的选择，满足用户不需要登录即可使用音乐播放等功能，也可在选择登录注册进行更加完善的功能使用。在分析完设计的需求后，网页版音乐播放器需要具备一些基本的功能：用户的登录注册模块、提供用户搜索的功能模块、对用户<sup>43%</sup>提供推荐的内容模块，音乐播放暂停、切换上下首、控制音乐进度条、音量的控制、音乐的下载、歌词的获取、播放列表的获取等基本功能。本设计为了方便用户的使用，在主页状态会首先设计用轮播图的形式展示一些新品单曲、歌单、首发专辑、策划等，在未登录状态用户可以进行音乐、歌手、用户、视频、歌单的搜索、对歌手的分类筛选歌单的筛选、歌曲的详情等功能。对于登录后的用户开放歌曲评论、歌单评论、以及动态信息分享功能，根据登录的状态获取用户个人信息关注动态以及编辑个人信息、注销登录、其他用户的公开信息、获取歌手的个人简介、视频、单曲等功能。这些功能设计可以有效的提供给用户的使用，方便简洁易操作是设计的初衷。

本系统设计技术采用前后端分离技术开发，采用前后端分离式开发的原因是该模式是目前流行的 web 应用开发模式，前端和后端分离成两个独立的项目，可以加快开发进度，而且他们是运行在两个不同的服务器中这样就减少传统的一个服务器的负担压力。本设计的前端部分采用的技术有 HTML、CSS、JavaScript、以及以一套以 MVVM 设计模式用于构建用户界面的渐进式的 VUE 框架等。VUE 的 M 是数据模型、V 是视图页面、VM 是数据模型和视图的控制器 viewmodel 数据双向绑定。这样可以实现页面数据改变的时候同时后台数据改变的时候可以相互连通实现数据与视图层同时更新。<sup>43%</sup>根据 Element UI 框架快速搭建音乐播放器的静态展示页面，组件化开发。CSS 是对页面的样式进行调整如定位、动画、色彩装饰等，可以更好的展示页面，让页面的外观更加美观简洁。使用 vuex 管理系统的部分状态信息数据，然后通过 Axios 与开源的后端 API 进行开发对接，根据开源网易云音乐后端的接口部署到本地 Node 服务上进行数据交互展示页面。再通过 vue-router 进行控制路由跳转，重定向，对登录和未登录状态用户的一些功能使用进行限制操作。

本设计的主要开发与测试工具有：Google Chrome 符合 W3C 规范，而且稳定性、速度和安全性都很高，具备开发者工具：网络请求检查、代码调试等开发调试功能，它功能具备强大、齐全等优点；<sup>63%</sup>Visual Studio Code 该工具内置诸如构建脚本、环境、调试等功能，再结合强大的文本编辑器的作用，对开发人员使用非常简单且有用。Visual Studio Code 在一个环境中处理所有事务且不需要旧

版的高度集成开发环境 IDE。本设计的主要开发环境有 Windows 11、Node18。

根据上述的系统需求在开发 web 应用时进行系统分析，进行开发一项 web 应用时需要确定系统开发需要的技术信息可行性、经济可行性、法理可行性、操作可行性、确定 web 应用系统的开发流程、根据设计系统的信息完成设计系统的整体架构、完成架构图绘制并进行细分应用开发功能模块、根据细分应用的功能模块进行代码编写、完成代码的编写后需要对系统的安全性、效率性、容错性等方面进行系统完整的测试，测试的目的是为了发现该系统是否有代码缺陷、逻辑错误、以及性能缺陷。检测系统的功能配置是否有错，功能是否可以正常运行，对发现应用出现的问题需要进行修复减少应用的问题出现、提高应用的稳定性、提高用户的体验性。

**关键词：**网页版音乐播放器；Vue；Node；Axios

## INTRODUCTION

With the development of technology and the progress of human civilization, people are striving to achieve a better life. It is inevitable that there will be various pressures in life, which requires some ways to relieve stress. Listening to music has become one of the ways for people to relax and unwind in their daily lives. Nowadays, the development of music is also very rapid. With the trend of technology, many music players on mobile phones and computers have emerged to facilitate users' listening. This design is aimed at better meeting the needs of users for a simple and convenient music player.

This design takes the web version music player as a case study. Based on the analysis of the demand for the web music player user market, this design combines the display of different pages for users who are not logged in and logged in, which is more convenient for users to choose from. It meets the requirement that users can use music playback and other functions without logging in, and can also choose to log in and register for more comprehensive functions. After analyzing the design requirements, the webpage music player needs to have some basic functions: the user login and registration module, the function module to provide user search, the content module to provide users with recommendations, the music playing pause, switching up and down, controlling the music progress bar, volume control, music download, lyrics acquisition, playlist acquisition and other basic functions. In order to facilitate users' use, this design will first display some new singles, playlists, debut albums, plans, etc. in the form of a rotating chart on the homepage. In the non logged in state, users can search for music, singers, users, videos, playlists, classify and filter singers, filter playlists, and provide details of songs. For users who log in, open song comments, playlist comments, and dynamic information sharing functions. Based on the login status, obtain user personal information, follow the updates, edit personal information, log out, other users' public information, obtain singer's personal profile, video, single, and other functions. These functional designs can effectively provide users with convenience, simplicity, and ease of operation, which is the original intention of the design.

The design technology of this system adopts front-end and back-end separation technology for development. The reason for adopting front-end and back-end separation development is that this mode is currently a popular web application development mode. The front-end and back-end are separated into two independent projects, which can accelerate the development progress. Moreover, they run on two different servers, reducing the burden and pressure of a traditional server. The front-end part of this design adopts technologies such as HTML, CSS, JavaScript, and a progressive VUE framework for building user interfaces using MVVM design patterns. The M of VUE is the data model, V is the view page, and VM is

the controller of the data model and view. The viewmodel data is bidirectional bound. This can achieve mutual connectivity when page data changes and backend data changes, enabling simultaneous updates of data and view layers. Quickly build a static display page for a music player based on the Element UI framework and develop it in a componentized manner. CSS adjusts the style of a page, such as positioning, animation, color decoration, etc., to better display the page and make its appearance more beautiful and concise. Use Vuex to manage some status information data of the system, and then develop and interface with the open source backend API through Axios. Deploy the data to the local Node service for interactive display page based on the interface of the open source NetEase Cloud Music backend. Then, control route jumps, redirects, and restrict the use of some functions for logged in and logged out users through the Vue router.

The main development and testing tools of this design include: Google Chrome complies with W3C specifications, and has high stability, speed, and security. It has developer tools such as network request checking, code debugging, and other development and debugging functions. Its functions are powerful and complete; Visual Studio Code is a tool that has built-in features such as building scripts, environment, debugging, etc., combined with the powerful text editor, making it very simple and useful for developers to use. Visual Studio Code handles all transactions in one environment and does not require an older version of the highly integrated development environment IDE. The main development environments for this design include Windows 11 and Node18.

Based on the above system requirements, a system analysis is conducted when developing a web application. When developing a web application, it is necessary to determine the technical information feasibility, economic feasibility, legal feasibility, operational feasibility, determine the development process of the web application system, complete the overall architecture of the design system based on the information of the design system, draw an architecture diagram, and subdivide the application development functional modules According to the functional modules of the segmented application, code writing is required. After completing the code writing, a complete system testing is required on the security, efficiency, fault tolerance, and other aspects of the system. The purpose of the testing is to discover whether the system has code defects, logical errors, and performance defects. Check if there are any errors in the functional configuration of the system, whether the functions can operate normally, and repair any problems found in the application to reduce the occurrence of problems, improve the stability of the application, and improve the user experience.

**Keywords:** version music player;Vue;Node;axiosstability



# 92% 网页版音乐播放器设计实现

信息与计算科学，201911921718，李志立

指导教师：黄小玲

## 1 概述

本设计的目的是为了为了更好的满足用户对音乐播放器界面美观、功能简洁、方便使用而设计的网页版音乐播放器。

### 1.1 设计的目的和意义

在科技的发展人类文明的进步下，大家都在为生活拼搏着，人们在为努力改善生活上的压力下听音乐成为了人们的日常生活放松身心的一种方式，音乐的发展也很迅速，在科技趋势下很多为了方便用户听歌，诞生了手机端以及电脑版的音乐播放器。例如国内的一些品牌“网易云音乐”、“酷狗音乐”等诸多品牌这些都是很优秀的音乐播放器。但是也有一些小瑕疵这些手机端以及电脑版的音乐播放器都是需要本地下载完成后安装使用的，这些音乐软件都会占用手机或者电脑的本地内存，为了满足用户不想要下载软件或者用户手机电脑内存不足的一些因素情况，于是设计开发一款网页版的音乐播放器是很有必要的，网页版的音乐播放器不占用用户太多内存就实现的在线听歌的需求网页版的功能和客户端软件功能差不多，也可以看视频。网页版的音乐播放器只需要手机或者电脑有浏览器就可以访问，这个给那些手机或者电脑不太先进的产品都可以实现在线听歌，或者用户使用的不是自己的手机电脑不方便下载音乐软件那么网页版的音乐播放器就实现了它的优点了。

### 1.2 开发的目的是意义

国外的音乐播放器的发展水平可以说是非常先进的，知名的有 Spotify、Tidal 等品牌。Spotify 是由瑞典国家研发的一款音乐播放器，在国外大部分国家的用户群体比较广的，这款播放器主要是音质很好，界面外观优雅简洁，音乐的品类丰富，几乎主流的音乐它都具备。这些正是它发展水平高的优点。

现在国内的音乐播放器品牌种类很多，可以满足不同用户使用的需求，音乐播放器的发展可以说是很高的水平了，在国际市场上也具有一定的地位，但是这些播放器功能繁杂，要实现一个任务需要操作很多步骤，不便于部分用户的使用。这些音乐播放器很多为了使软件功能多样化，添加了很多不必要的功能，这些功能在无形中就占用了很多手机或者电脑的内存资源，盲目追求外观的好看，功能的实用性有待加强。

网页版音乐播放器主要的功能还是给用户带来让人心情愉悦的音乐，所以设计这个网页版音乐

播放器重在提高音乐的音质问题可以给用户带来纯粹的音乐赏识，带给用户优质的体验，虽然拥有一副好看的外表也会带给人们愉悦的心情，但音乐播放器还是更注重音乐的品质。

基于上述的分析我觉得做好一个优秀的网页版音乐播放器我觉得第一是音乐的品质音质问题，第二网页的界面外观的优雅简洁，第三功能的针对性。

## 2 开发的技术与工具简介

本系统设计要设计网页版音乐播放器需要确定使用的开发技术和工具并搭建好开发环境。

### 2.1 开发的技术

技术主要是使用 visual studio code 开发工具进行代码开发、代码编写采用了 HTML、CSS、JavaScript、vue 框架、vue router 路由管理器进行页面路由管理、通过 Axios 与后端 API 进行对接、Elementer-UI 框架、vuex 状态管理、npm、nodejs 部署网易云开源后端服务器等技术进行开发，使用 vue 脚手架的 Webpack 打包工具打包。

#### 2.1.1 前端开发

本系统设计的前端开发创建 web 页面搭建页面展示给用户的过程，前端开发重要性是它直接对用户，可以提供页面给查看以及直接对用户功能操作并能及时响应数据给用户查看。本设计前端主要采用的技术有 HTML、CSS、JavaScript、element-ui、Vue 框架、axios 等技术开发。

HTML、CSS、JavaScript 统称“前端三剑客”是前端开发的技术基石<sup>[1]</sup>。HTML 是前端开发的基本条件，HTML 是一种超文本标记语言主要是对页面的结构布置<sup>[2]</sup>。CSS 是对页面的样式进行调整如定位、动画、色彩装饰等，可以更好的展示页面，让页面的外观更加美观简洁<sup>[3]</sup>。JavaScript 是一种具有函数优先的轻量级，解释型或即时编译型的编程语言<sup>[4]</sup>。适用于 web 网页开发，用于处理用户操作和数据交互等。

Element-UI 是为了简化一些常用的组件化开发，降低开发难度而产生的，它是由饿了么团队做的基于 vue 封装的组件库，里面封装了许多 icon 图标、按钮、表单、标签页、分页等组件，这些组件大大的提高了开发人员的效率。使用也非常简单根据需求在 vue-cli 下载 Element-UI 引入即可使用。

Vue 框架是一个 MVVM 的渐进式、轻量级、自底向上增量开发、用于构建用户界面的渐进式框架，Vue 的核心只关注视图层，通过简单的 Api 实现响应式的数据绑定和组合<sup>[5]</sup>。是一套以 MVVM 设计模式用于构建用户界面的渐进式的 VUE 框架等。VUE 的 M 是数据模型、V 是视图页面 view、VM 数据模型和视图的控制器 viewmodel 数据双向绑定<sup>[6]</sup>。可以实现页面数据改变的时候同时后台数据改变的时候可以相互连通实现数据与视图层同时更新，它不同与原始的 HTML，原始的 HTML 是不会自动更新 DOM，需要手动更新。Vue 可以检测到用户更改的数据进行自动更新，然后渲染到页面展示给用户，vue 的 vue-cli 非常方便，它里面提供了很多库给开发者使用，例如 vuex 状态管理库可以用于统一管理数据和状态；vue router 是对控制页面的跳转以及重定向的重要库，它可以设置用户在输入路由参数时设置对登录注册页面跳转到页面的权限；Webpack 代码打包压缩的工具，less、scss

样式预处理器等等，可以通过 npm 或 yarn 包管理工具安装下载第三方库例如 Axios、Babel、UI 框架、express。

Axios 是通过 Promise 实现对 ajax 技术的一种封装，一个基于 JavaScript 的 ECMAScript 6 的 Promise 的 HTTP（对 XHR 封装的网络请求）库，即可用于浏览器也可以用于 node 中，它的优势在于采用 Promise 解决了回调地狱问题、拦截请求和响应、客户端支持防御 XSRF 等。传统的 HTML 页面，采用 AJAX 进行页面的异步数据交互，通过网页内的 javascript 脚本向服务器请求访问后台资源某些数据内容，把数据反馈给页面进而对页面的部分内容进行渲染更新以及相应的逻辑操作，最终与后端数据库实现交互。

## 2.1.2 后端开发

后端开发主要是部署服务器，下载开源的网易云音乐后端接口然后通过在本地的 node 环境下进行部署服务器。根据开源接口的 API 处理相应的前端数据请求和逻辑处理。

## 2.2 开发工具

开发一款互联网 web 应用时，选择好的开发工具可以大大的提高有效的开发效率，本系统设计采用的是现在流行的 Visual Studio Code 以及 Google Chrome 开发工具。

Visual Studio Code 是一款开放源代码的轻型文本编辑器，面向 Web 开发人员的，不仅功能完善，还支持面向所有开发人员的多种扩展。Visual Studio Code 将是一款非常有用的开发工具。该工具内置诸如构建脚本、环境、调试等功能，再结合强大的文本编辑器的作用，对开发人员使用非常简单且有用。Visual Studio Code 在一个环境中处理所有事务且不需要旧版的高度集成开发环境 IDE。用于开发网页版音乐播放器是一个很好的选择。

Google Chrome 是一款由谷歌公司开发的网页浏览器。该浏览器是基于其他开源软件 WebKit，并采用了 WebKit 内核撰写的，符合 W3C 规范，而且稳定性、速度和安全性都很高，具备开发者工具：网络请求检查、代码调试等开发调试功能，它功能具备强大、齐全等优点。可以造出简单且有效率的界面。本设计采用 Chrome 来开发可以更好的解决兼容性问题。

## 2.3 开发环境

开发 Web 应用的时候一般首先需要搭建好开发环境，然后再等完成代码的编写后再打包好上线到线上环境。

本系统设计是基于微软的 Windows 11 操作系统进行代码的开发和测试，以及在 node 环境下部署网易云开源的后端接口服务。

Node.js 是由 Ryan Dahl 开发，是一个基于 Chrome V8 引擎的 JavaScript 运行环境，使用了一个事件驱动、非阻塞式 I/O 模型<sup>[7]</sup>。它跟浏览器不同的是，浏览器运行在客户端上，而它运行在服务端上，它具有速度快、性能好、跨平台等特点。Node 也可以通过 npm 或 yarn 包管理器，安装第三方库等拓展插件，用作项目的编译、打包、开发、服务器运行等。Node 作为一个新兴的前端框架，

53% 后台语言。选择 Node 是一个很好的有利于 Web 应用开发。本系统设计就采用 Node18 版本进行部署服务器运行。

### 3 系统分析

#### 3.1 可行性分析

可行性分析是为了搞清楚开发的项目是不是可以实现的值得进行的，通常从经济可行性、62% 技术可行性、法理可行性、操作可行性方面进行分析<sup>[8]</sup>。

经济可行性分析：本设计主要是使用的开发技术工具是以 Google Chrome 浏览器以及网易云音乐接口都是开源的不存在消费问题。

技术可行性分析：技术可行性是设计的关键内容，本设计是由个人独立开发前端部分，后端使用开源的网易云音乐接口实现，主要通过学习前端开发以及 node 环境部署等一些计算机网络知识便可以实现开发。40% 采用了主流的 web 应用开发模式前后端分离式开发，目前大部分的 web 应用都是采用前后端分离式开发，可以很好的证明前后端分离式开发的技术可行性，采用 vue 框架主要是现在的 vue 框架有大量的官方文档详细的解析说明它的使用流程，开发速度快，技术成熟，简单易于开发人员上手。而且对软件的调试比较简单，需要的计算机配置也不高。

法理可行性分析：本设计是由本人进行界面设计，以开源的网易云音乐后端接口进行设计的都是开源免费的不存在侵权问题，网页版的音乐播放器设计没有非法收集用户的个人隐私信息在法律上是可行的。

操作可行性分析：网页版的音乐播放器是非常容易上手的，根据提示完全可以自主操作，只要懂得一些电脑的使用，鼠标移动点击输入等都可以使用本系统设置。

#### 3.2 需求分析

51% 手机电脑现在普遍存在，人们日常生活都离不开这些。根据市场的需求进行分析实现对用户更友好的网页设计，现在的市场虽然音乐播放器的品牌很多，也有很多非常优秀的播放器，但是有些还是满足不了用户的需求，这样我可以设计一款简洁优雅的网页版音乐播放器就适合那些不想下载软件就可以听歌的用户。

本设计根据用户需要对 web 应用的使用需求进行分析用户可自由选择登录或者注册、用户可进行音乐播放声音大小及进度条控制、可以搜索歌曲歌单、获取关注的信息、观看歌曲 mv 视频、登录用户的信息编辑，用户可以下载部分免费音乐等。

#### 3.3 功能及模块分析

53% 根据需求分析可以大致分成以下几个功能模块：

- (1) 登录注册模块。
- (2) 搜索模块。
- (3) 发现音乐模块。

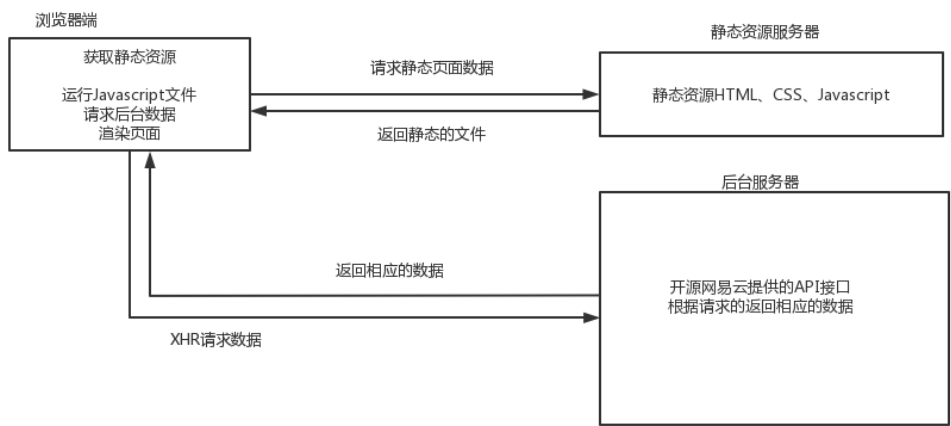
- (4) 视频模块。
- (5) 登录用户的关注模块。
- (6) 登录用户的私人 FM 播放模块。
- (7) 歌手信息模块。
- (8) 歌曲排行榜模块。
- (9) 最新音乐模块。
- (10) 控制音乐播放歌词显示模块。

4 系统设计

4.1 程序设计

在开发设计 web 应用需要对进行性能优化，在前端方面我们可以分离文档、表现、行为，<sup>73%</sup>就是将 HTML、CSS、JavaScript 方面进行分离，<sup>52%</sup>从而形成独立的文件。只需要在 HTML 文档的头部加上链接 CSS 文件或者 JavaScript 脚本的代码。<sup>70%</sup>CSS 文件、JavaScript 脚本能够同时被多次应用，这样能够降低前端文件的大小，同时就能够提高加载的速度<sup>[9]</sup>。

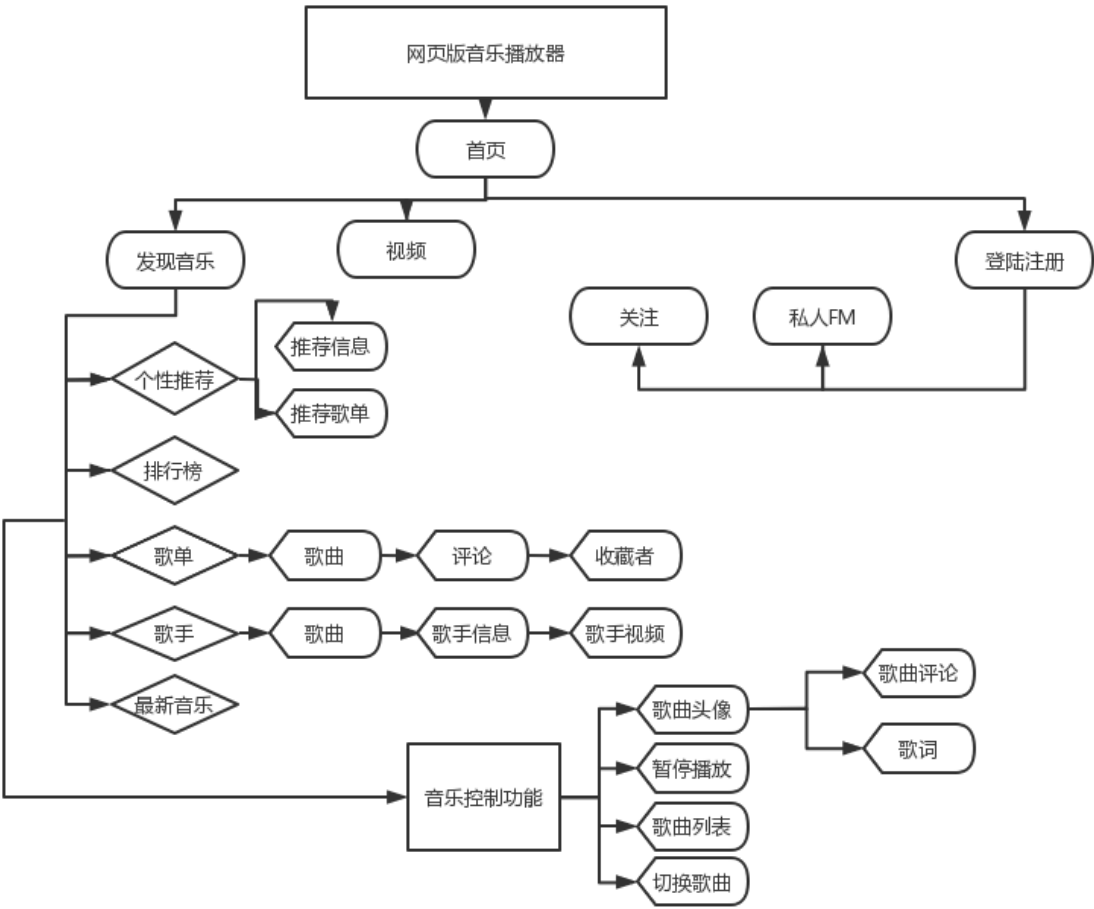
本系统程序设计采用前后端分离开发模式，前端负责利用 vue 框架结合 Element UI 框架编写 web 网页版音乐播放器的静态视图代码以及一些逻辑控制和控制页面的渲染、登录注册、路由的跳转、数据的异步请求、用户的操作控制页面与后端的数据实现交互，这样能大大降低系统的耦合度<sup>[10]</sup>。如图 4-1 所示。



<sup>83%</sup>图 4-1 前后端分离开发模式

4.2 系统结构设计

根据系统的需求和功能及模块分析，可以设计好网页版音乐播放器的整体系统结构如图 4-2 所示。



73%

图 4-2 音乐播放器系统结构示意图

5 系统实现

5.1 登录注册

由于后端接口拿的是开源网易云接口，用户点击注册调用注册接口，传入手机号码和验证码、密码、昵称，可进行注册音乐账号(同时可修改密码)二维码登录后台返回 base64 的图片前端拿到渲染上页面，用户可以通过 app 扫码登录，用户同时可以进行验证码登录。登录后把用户的登录状态信息保存在浏览器的 cookie 中，对于需要登录的页面跳转设置路由守卫重定向到登录页面，如下图 5-1，图 5-2，图 5-3 所示。

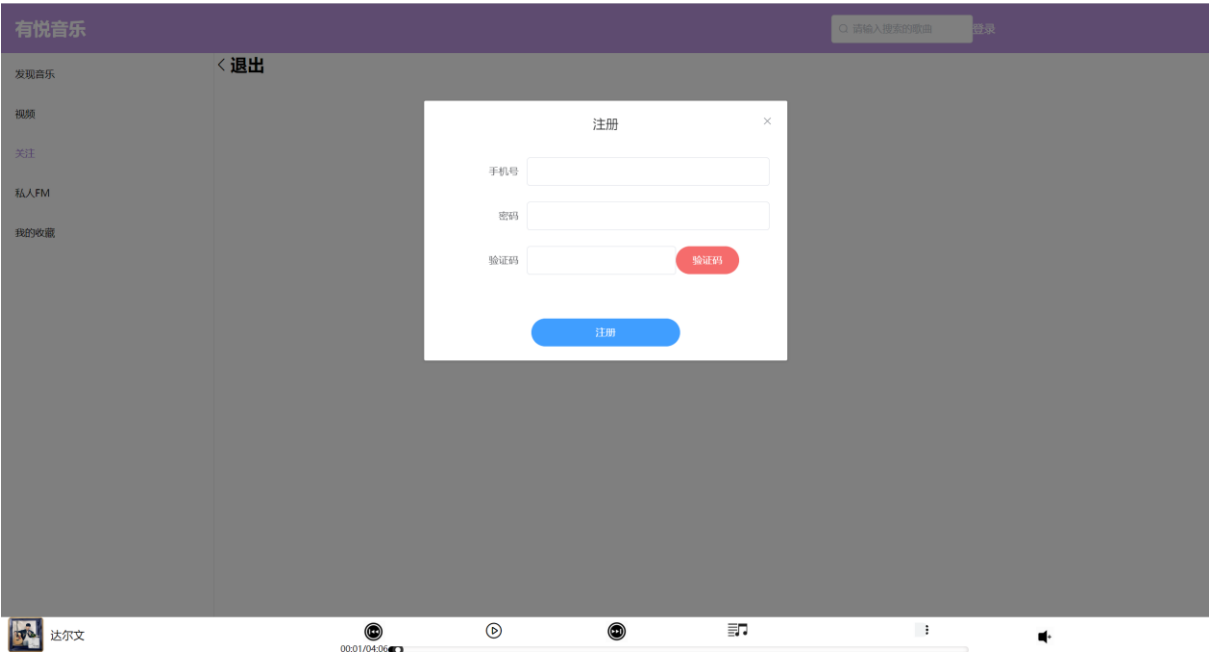


图 5-1 用户注册



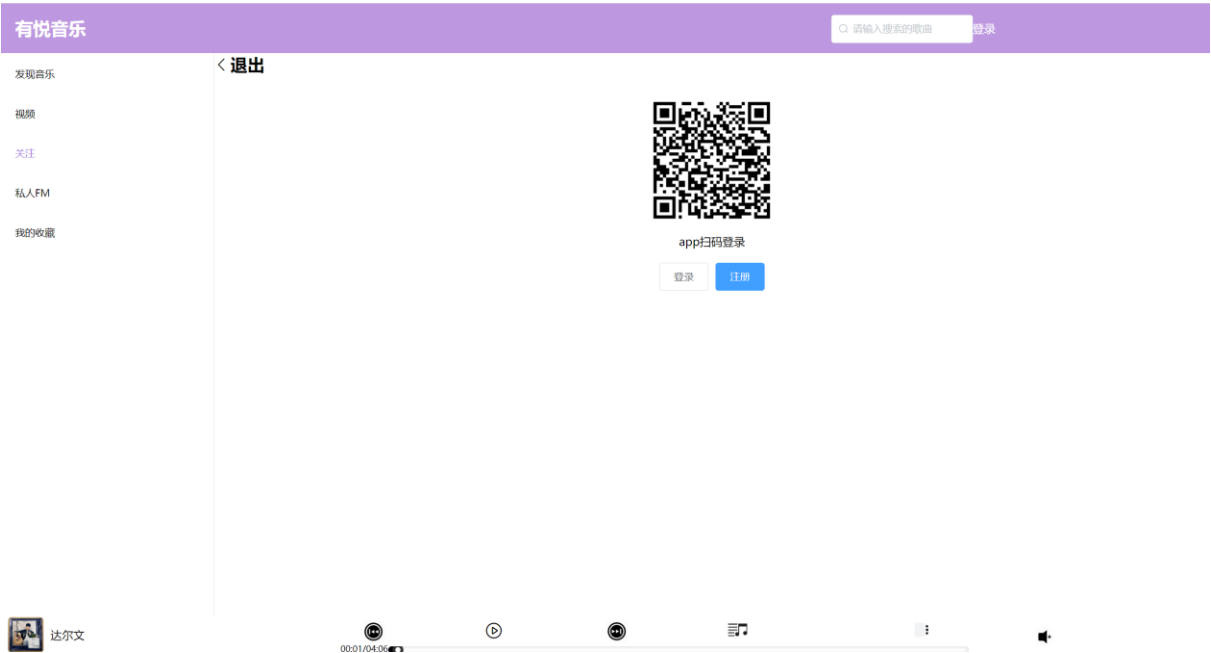


图 5-2 扫码登录

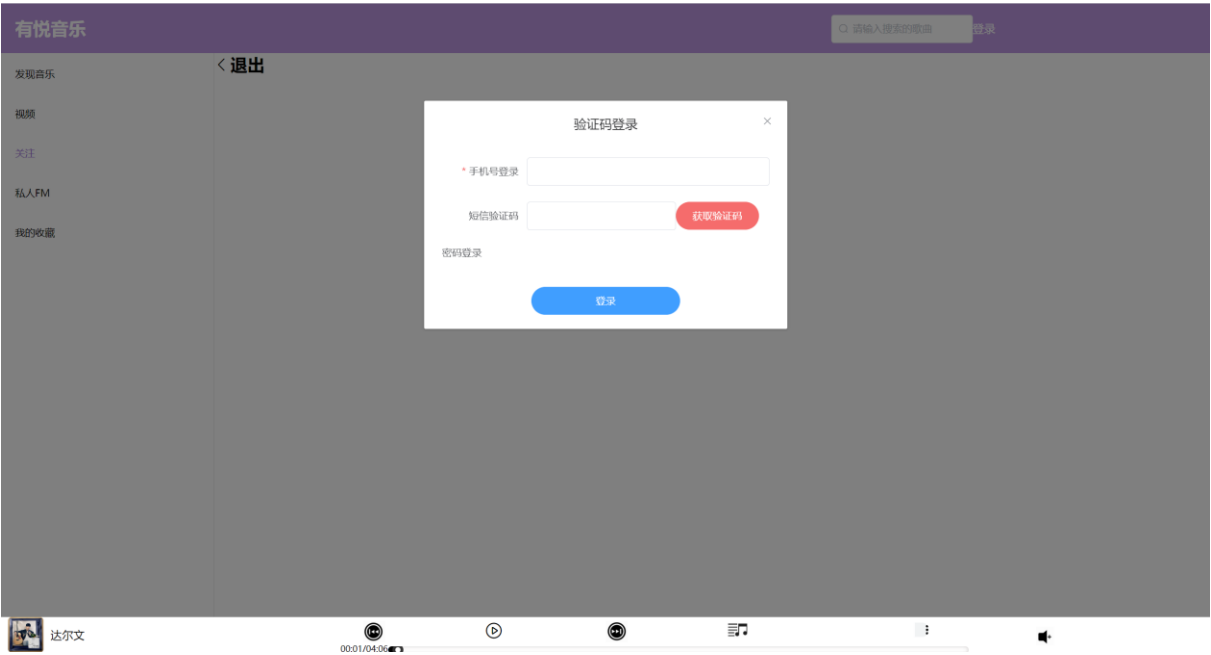


图 5-3 手机验证码登录

5.2 搜索

用户可在搜索框输入需要搜索的内容按回车键后界面把 input 搜索框中的信息通过 post 请求向后台服务器发送请求，服务器收到请求进行处理返回相应的数据，页面拿到服务器返回的数据渲染展示搜索到的单曲、歌手、视频、歌单、用户的信息数据。如图 5-4、图 5-5、图 5-6、图 5-7 和图 5-8 所示。



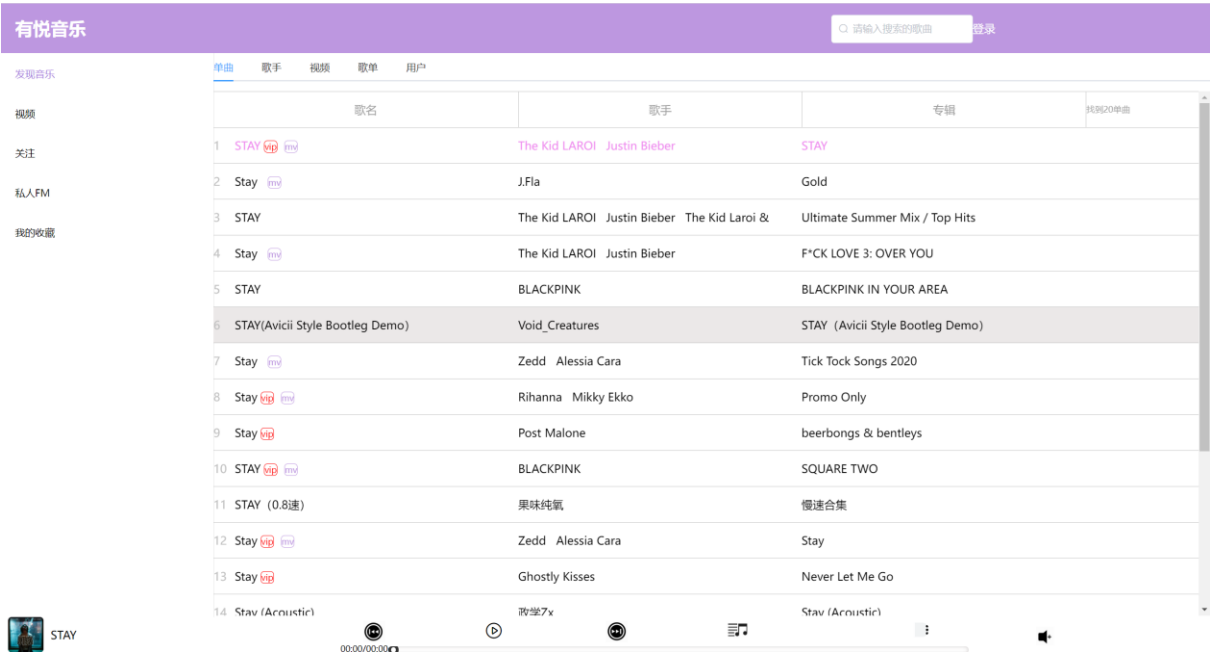


图 5-4 搜索功能的歌曲

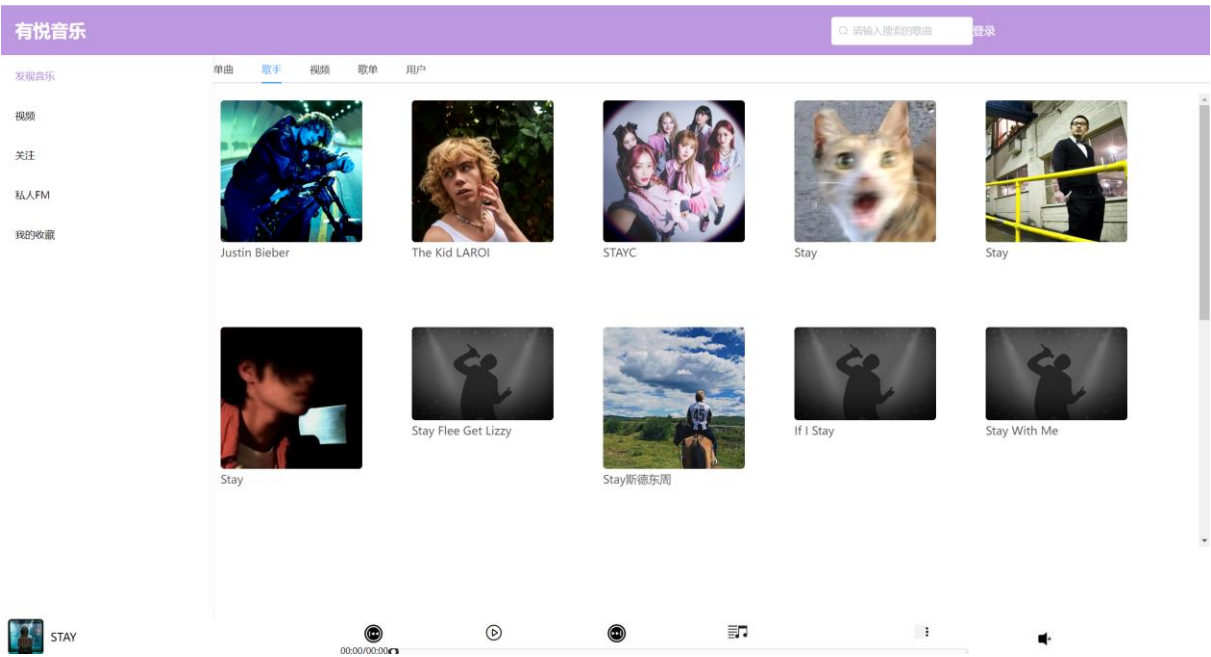


图 5-5 搜索功能的歌手

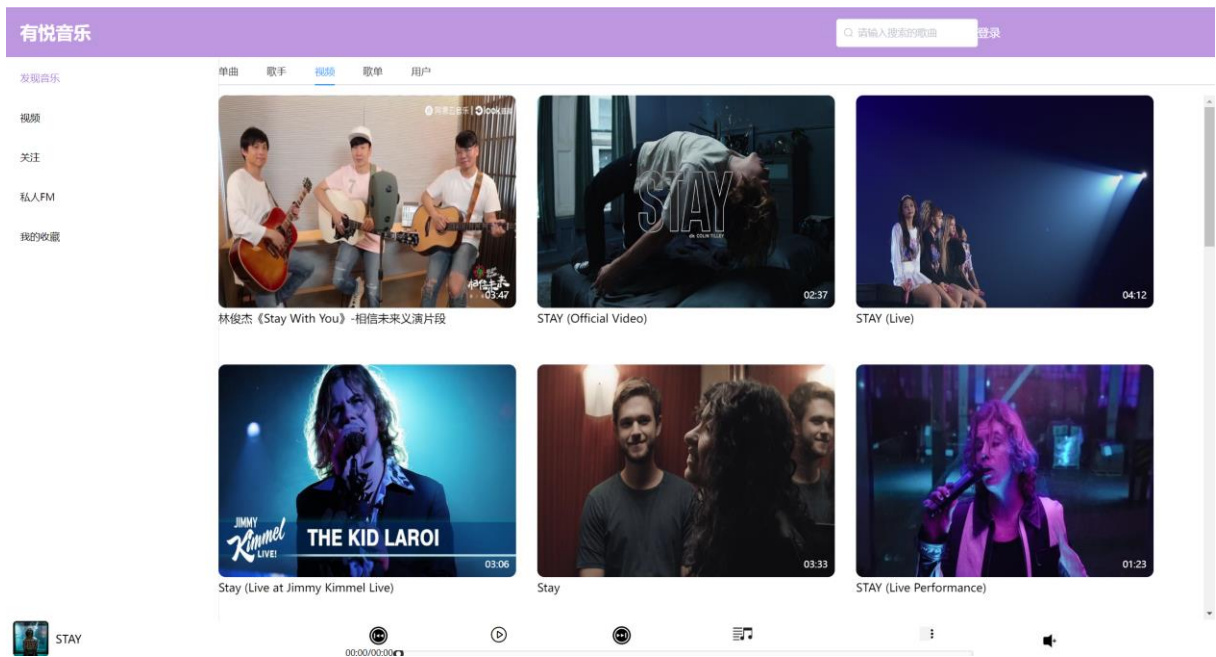


图 5-6 搜索功能的视频

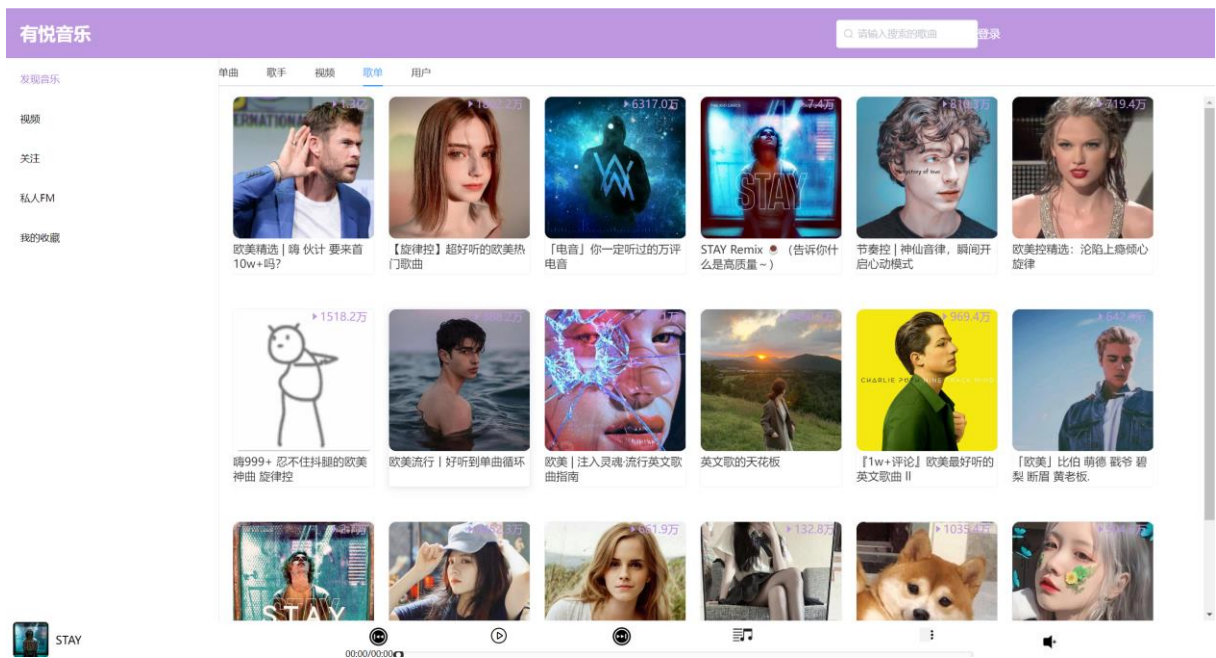


图 5-7 搜索功能的歌单

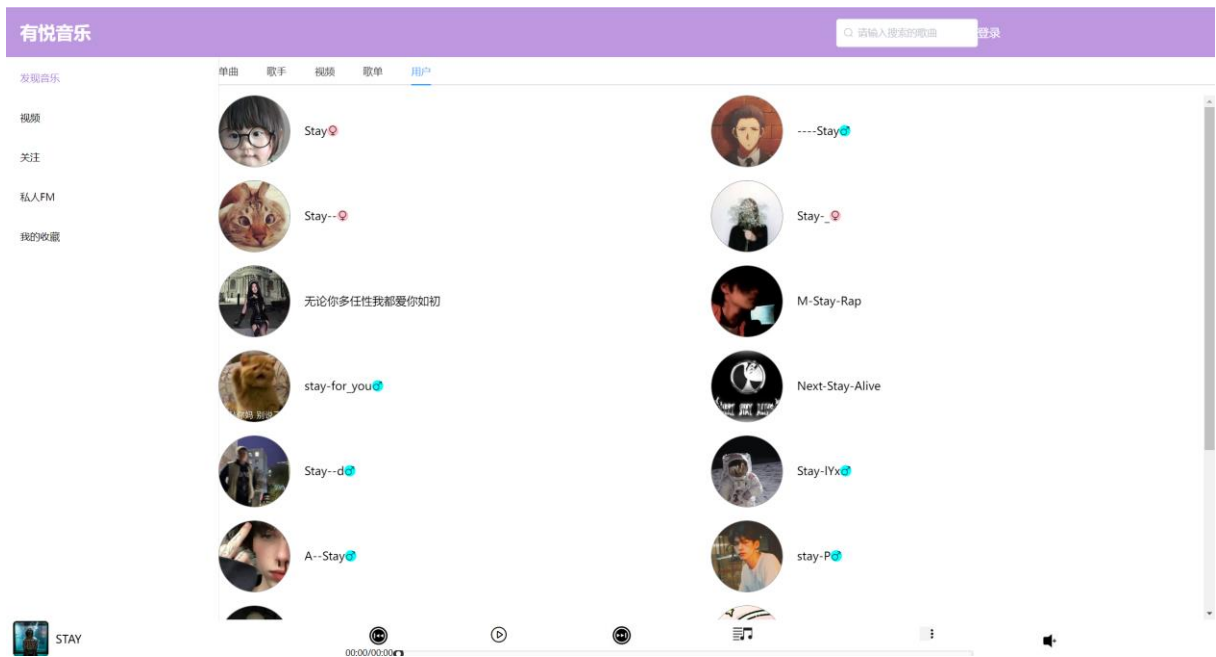


图 5-8 搜索功能的用户

### 5.3 发现音乐

后台服务器返回展示主要的推荐信息、新歌发布信息、以及推荐歌单数据，页面进行数据渲染如图 5-9 所示。

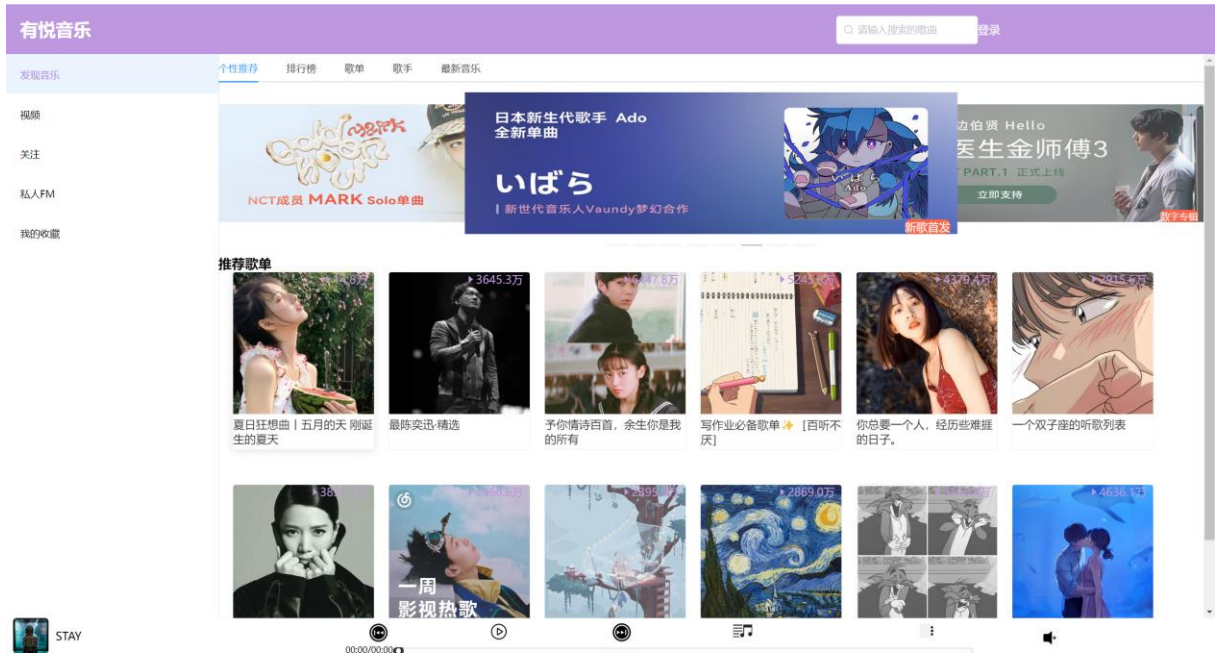


图 5-9 发现音乐模块

#### 5.3.1 排行榜

点击首页的排行榜向后台服务器发送请求，服务器收到请求进行处理返回相应的数据展示歌曲

的排行榜信息以及榜单的前六首音乐的排名，如图 5-10 所示。

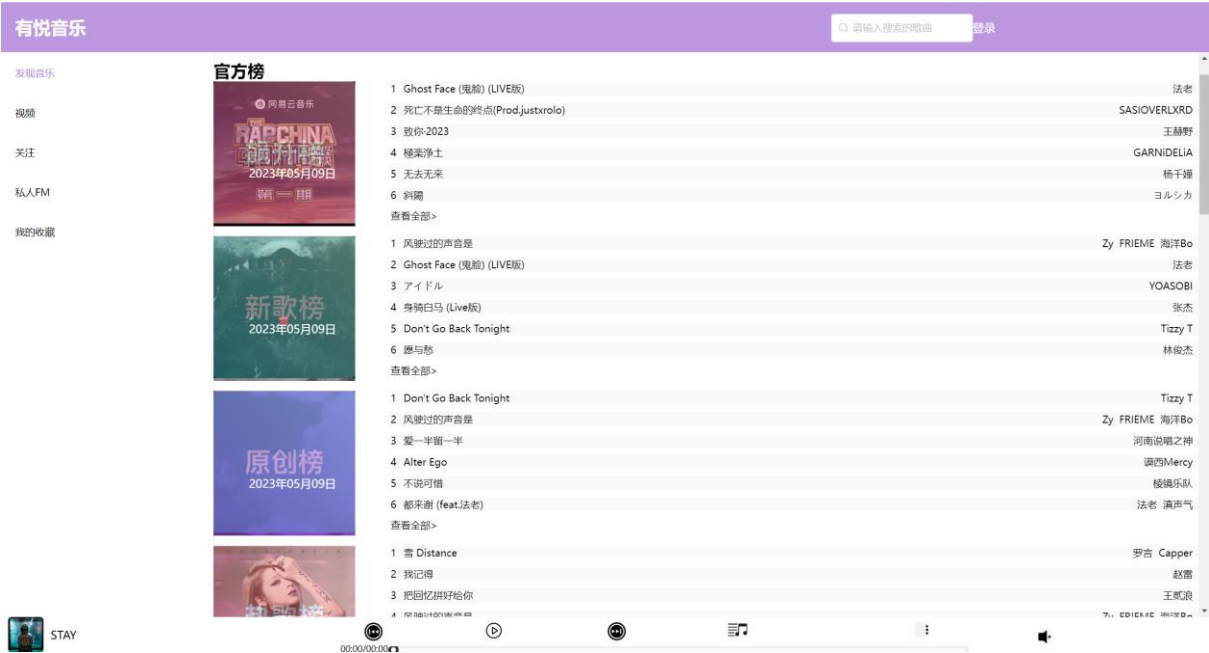


图 5-10 排行榜

5.3.2 歌单

首页的歌单向后台服务器发送请求，服务器收到请求进行处理返回相应的数据展示歌曲的排行榜信息，44%，用户可以点击选择不同的歌单的分类进行筛选歌单，如图 5-11 所示。

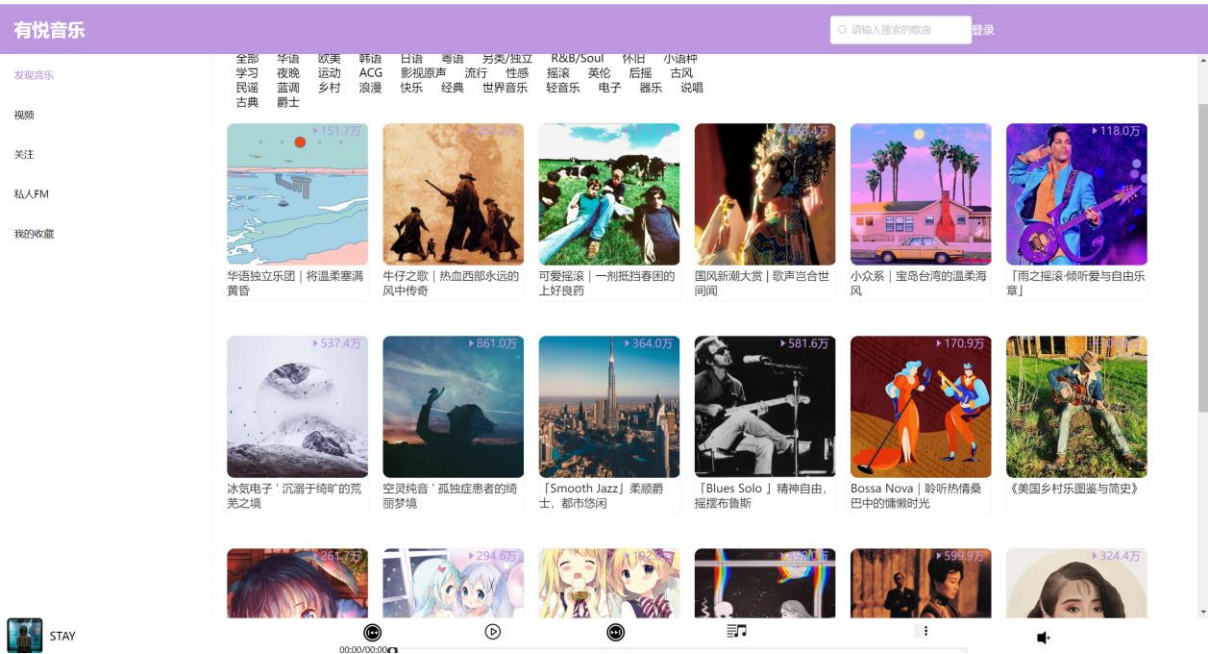


图 5-11 歌单

5.3.2.1 歌单歌曲

点击歌单后向后台服务器发送请求，服务器收到请求进行处理返回相应的数据展示歌单的歌曲、



评论、收藏者，歌单的创作者信息、创作时间、歌单的播放量、歌单歌曲数量、歌单的简介、收藏量、分享量，如图 5-12 所示。

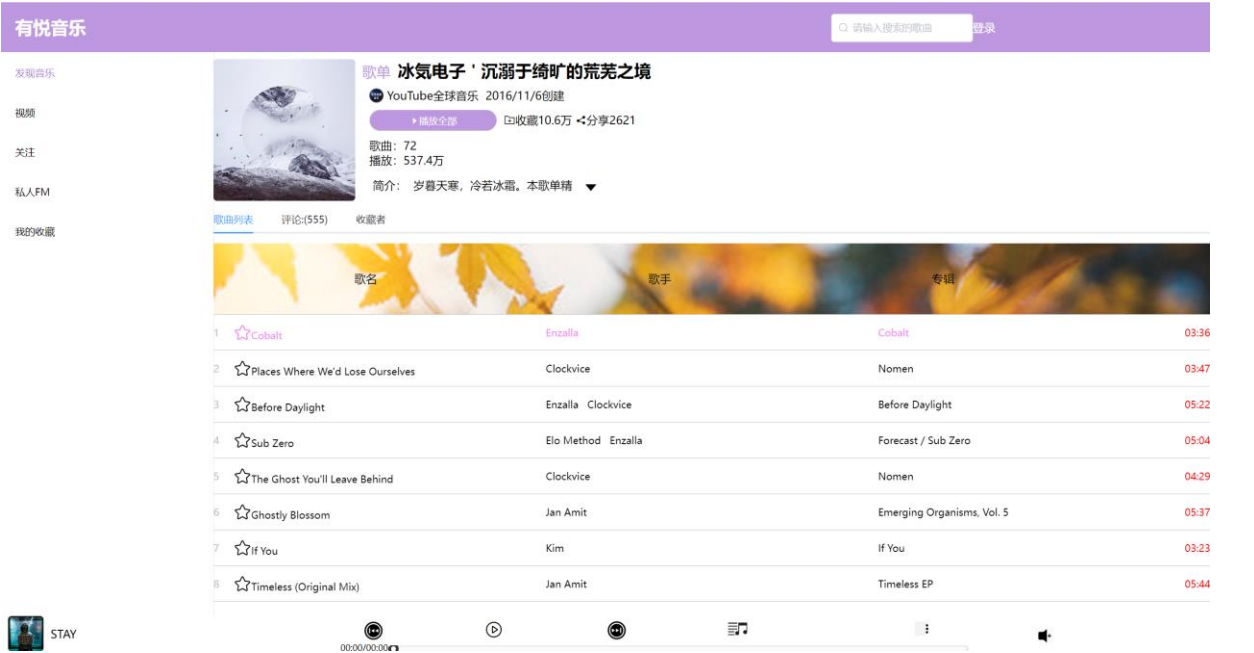


图 5-12 歌单详情

5.3.2.2 歌单评论

用户可以点击歌单评论获取评论信息，已登录用户可以对歌单发送评论，如图 5-13 所示。



图 5-13 歌单评论

5.3.2.3 歌单收藏者

用户可以点击歌单收藏者获取歌单的收藏者信息，如图 5-14 所示。

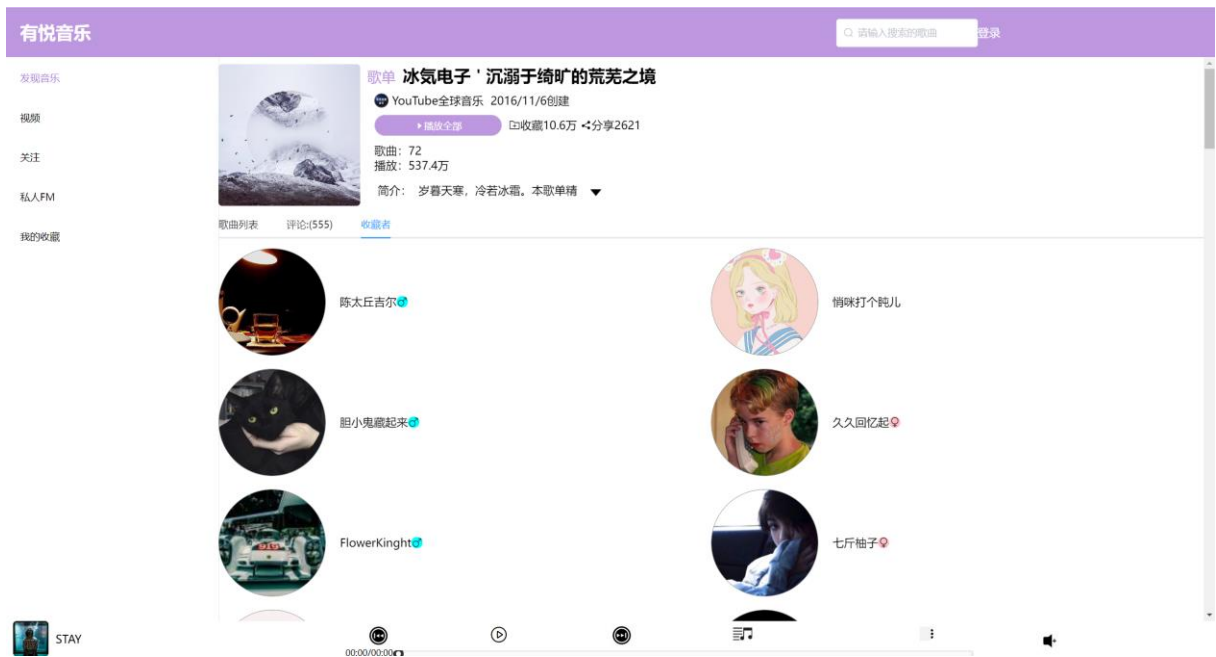


图 5-14 歌单收藏者

### 5.3.3 歌手

点击首页的歌手向后台服务器发送请求，服务器收到请求进行处理返回相应的数据展示歌手信息，获取歌手的个人简介，<sup>52%</sup> 视频，以及歌手的歌曲，如图 5-15 和图 5-16 所示。

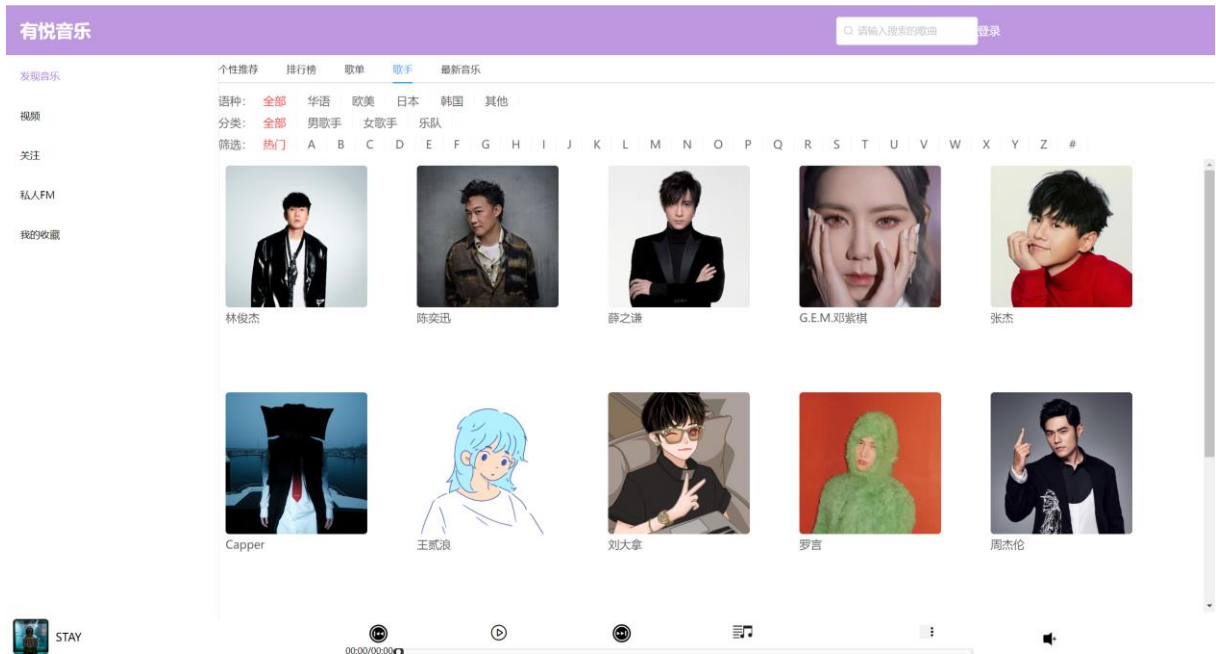


图 5-15 歌手

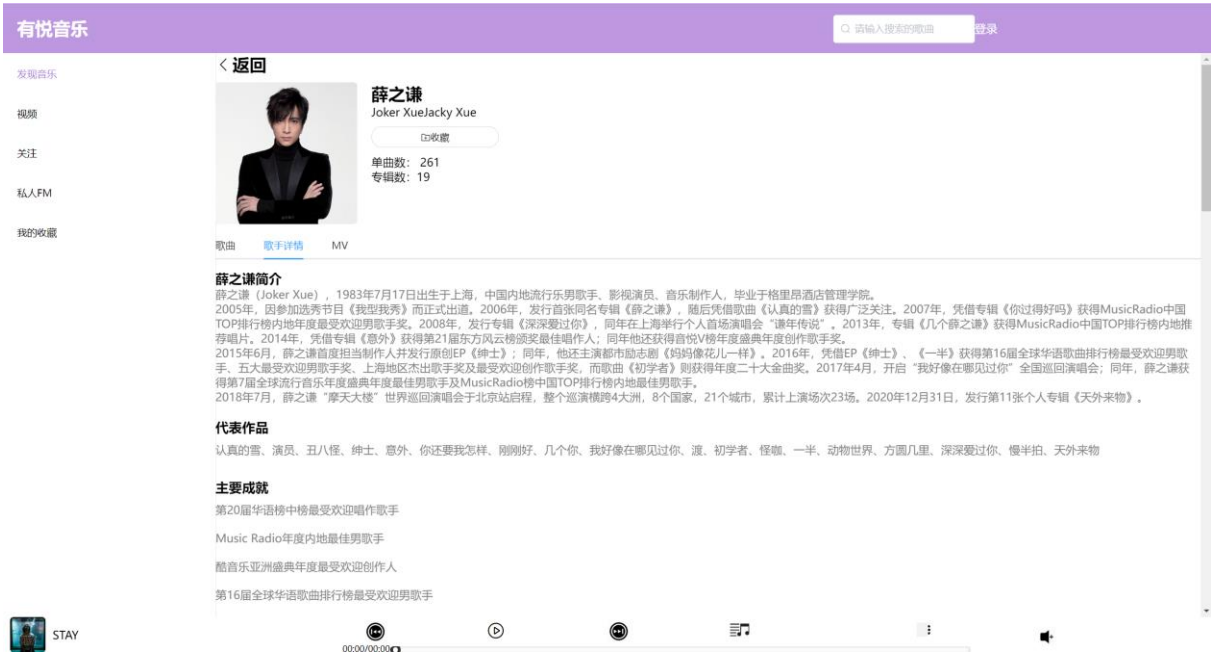


图 5-16 歌手信息

5.3.4 最新音乐

点击首页的最新音乐向后台服务器发送请求，服务器收到请求进行处理返回相应的数据展示最新音乐的具体信息，如图 5-17 所示。

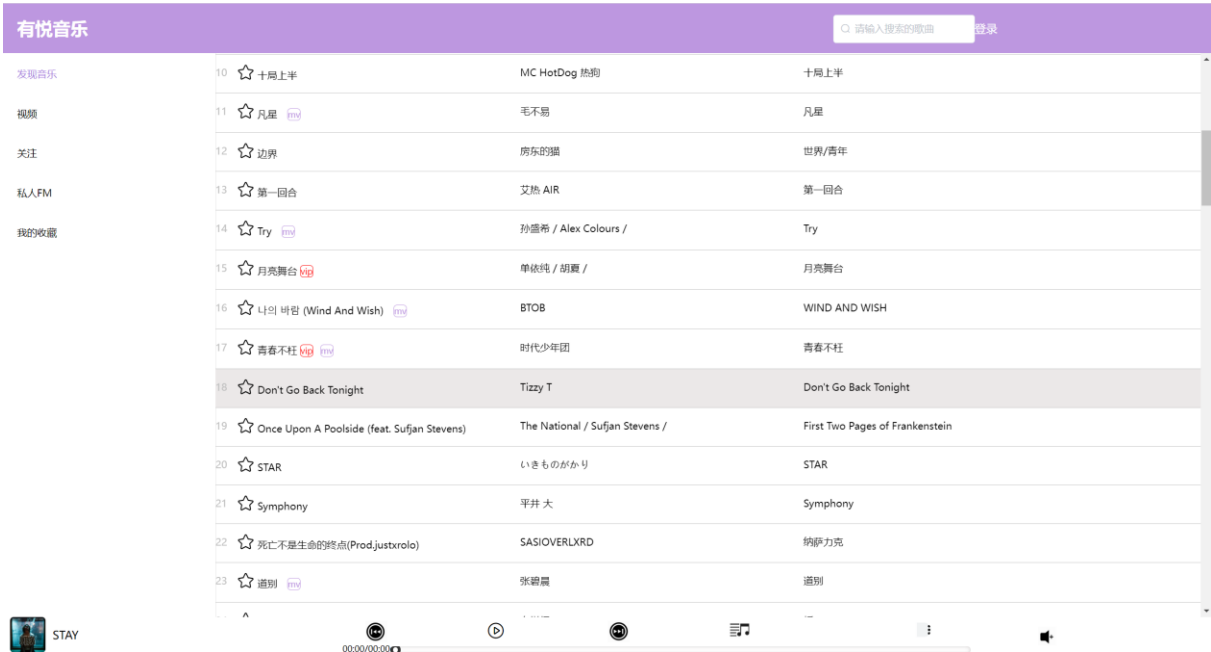


图 5-17 最新音乐

5.4 视频

点击首页视频也可以点击歌曲的 mv 按钮携带视频 id 跳转导航到视频界面向后台服务器发送请求，服务器收到请求进行处理返回相应的数据展示视频的信息，已登录的用户可以进行评论，如图

5-18 所示。

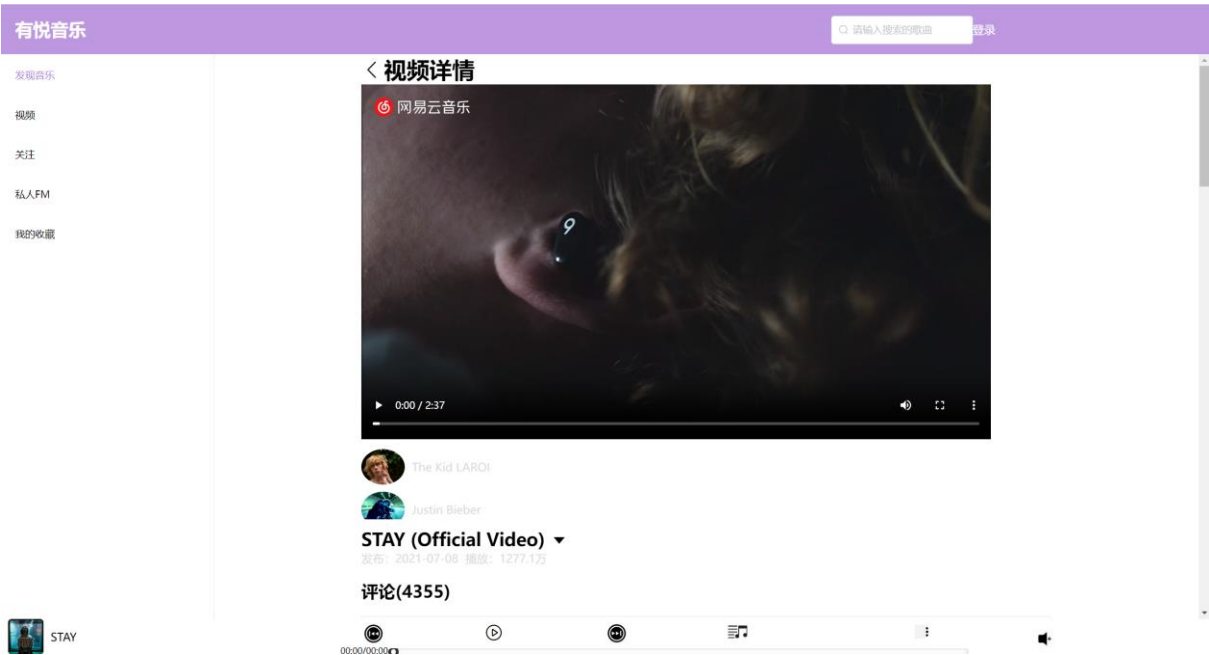


图 5-18 视频

5.5 关注

对于只对于登录状态的用户进行数据展示，未登录状态的用户无法获取数据展示，登录用户可以在线编辑分享动态信息，如图 5-19、图 5-20 和图 5-21 所示。

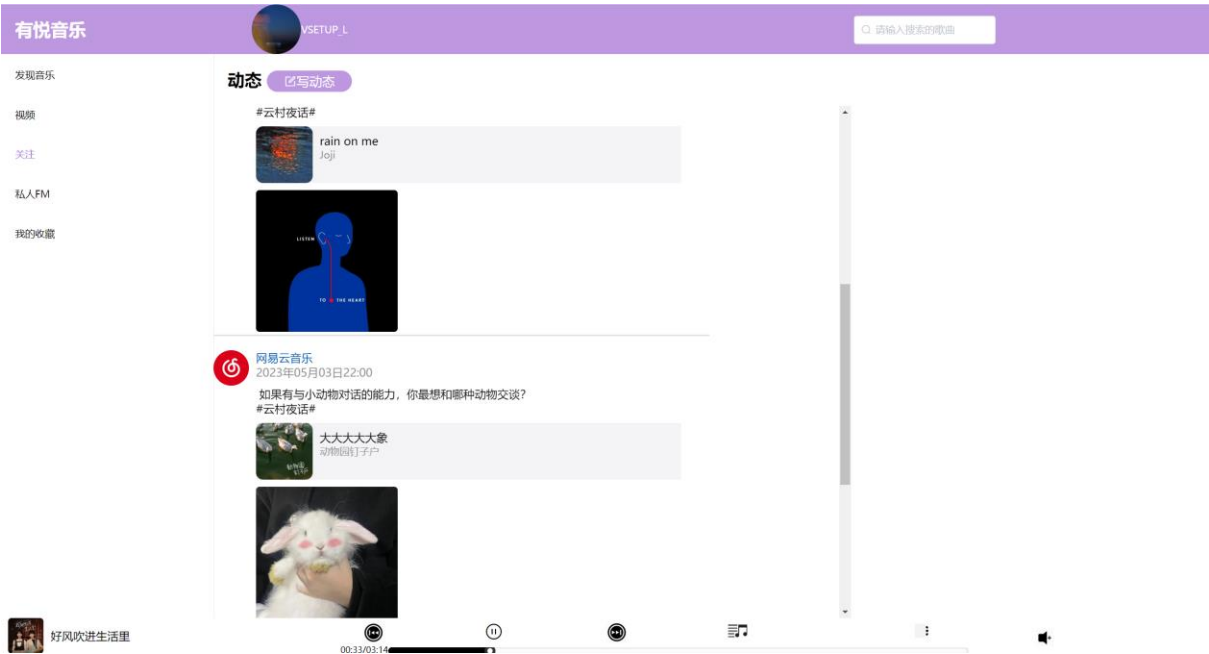


图 5-19 登录状态的关注



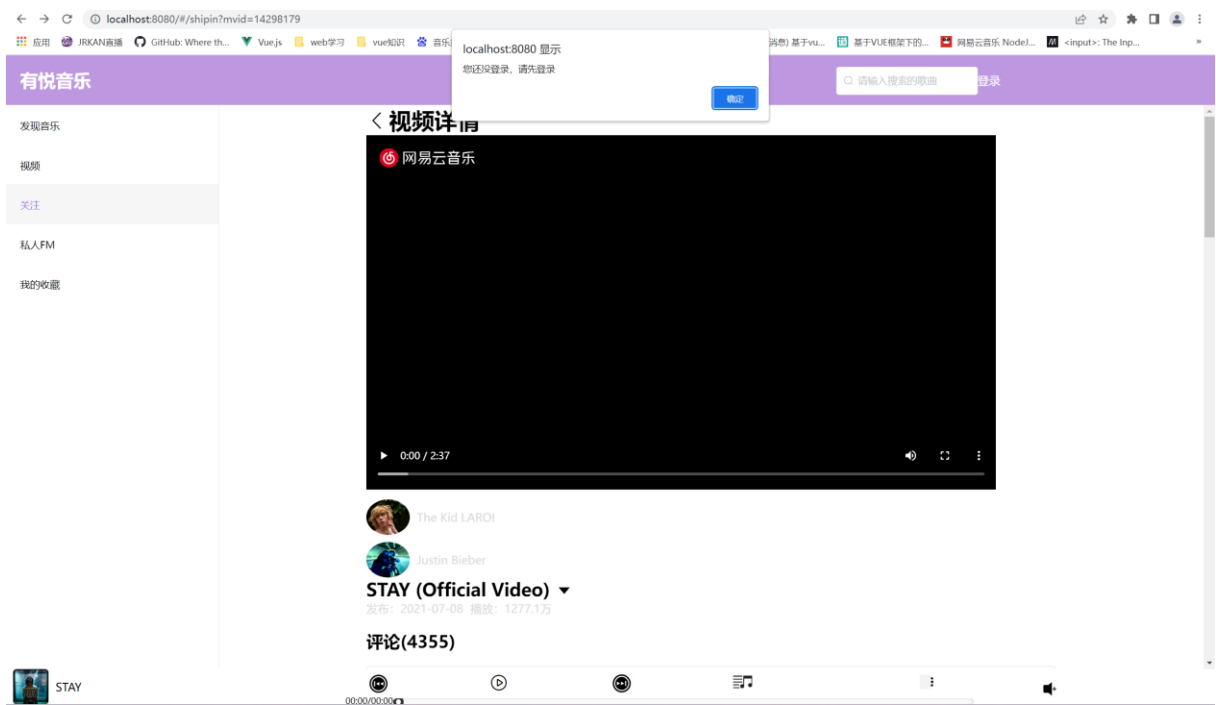


图 5-20 未登录状态的关注

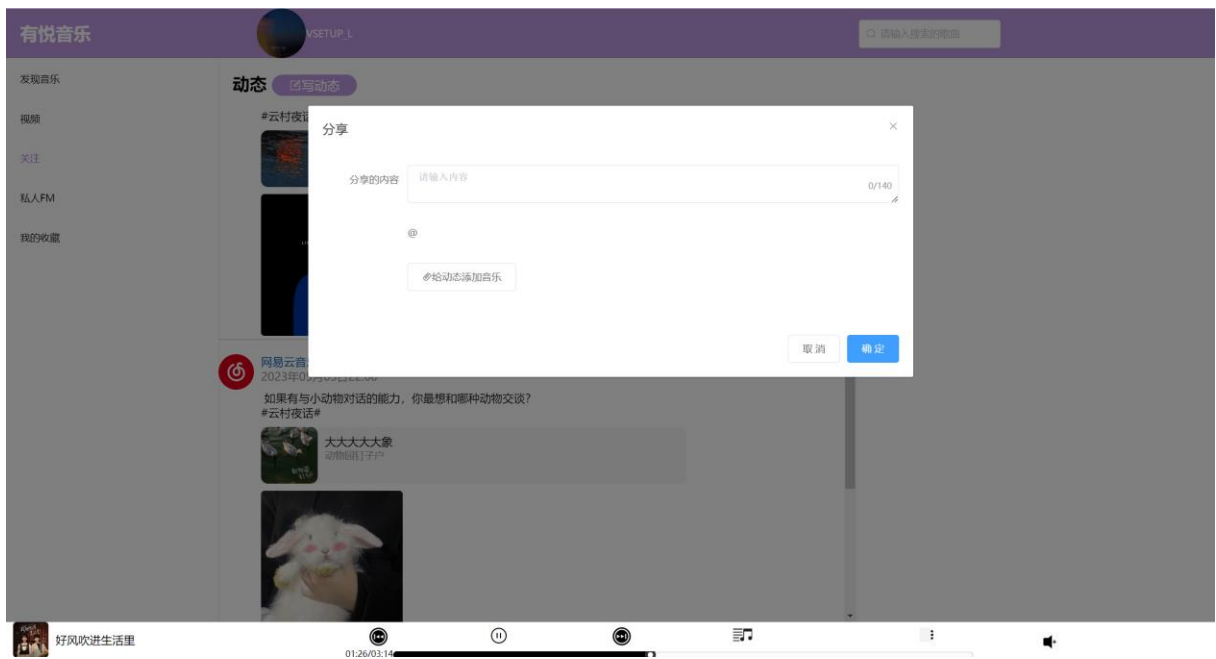


图 5-21 登录状态可编辑分享动态

### 5.6 私人 FM

点击首页私人 FM 跳转导航到 FM 界面向后台服务器发送请求，服务器收到请求进行处理返回相应的数据展示 FM 的信息，通过点击评论按钮可以发送评论限制于已登录用户，如图 5-22 所示。



图 5-22 FM 推荐音乐

5.7 底部音乐的控制

用户可以通过点击上下首进行歌曲的切换、点击播放列表进行展示音乐的曲目、点击歌曲头像获取歌曲的歌词以及评论信息，拉取音乐进度条可以控制音乐的播放时间，提供部分免费音乐的下载功能，调节音乐音量的大小，如图 5-23 和图 5-24 所示。



图 5-23 歌曲详情

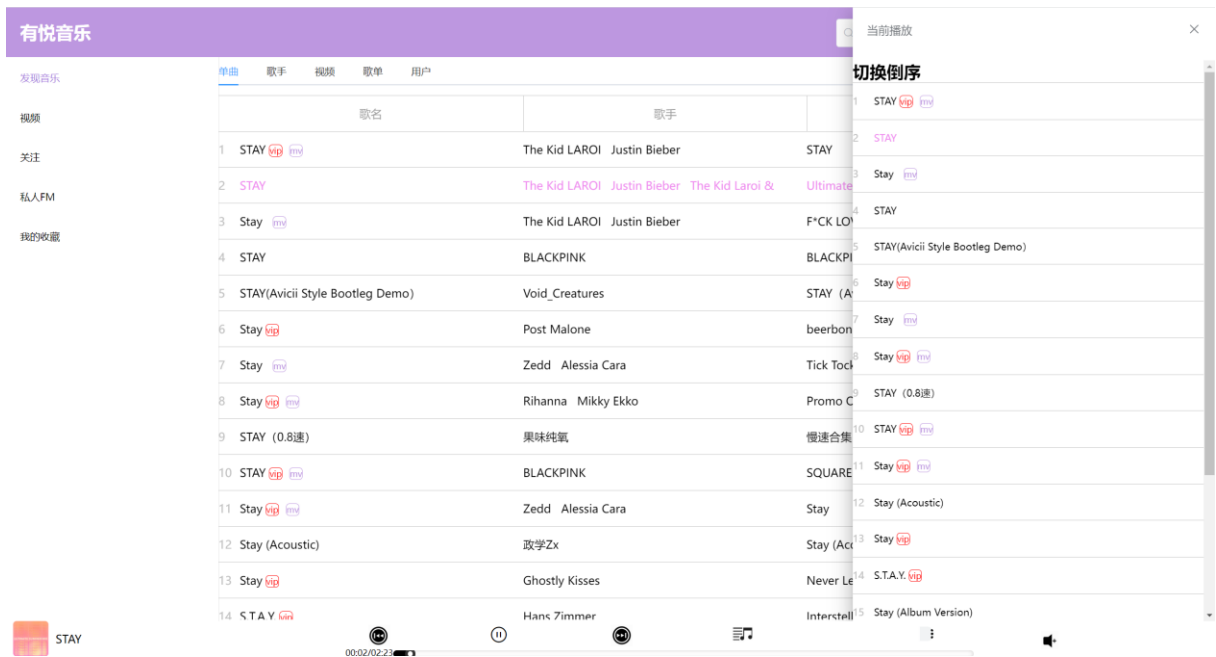


图 5-24 播放列表

5.8 用户主页信息

点击用户头像可获取用户主页信息，用户通过点击关注、粉丝、动态获取相应信息，根据用户的登录状态可以提供用户编辑个人信息，如图 5-25、图 5-26 和图 5-27 所示。

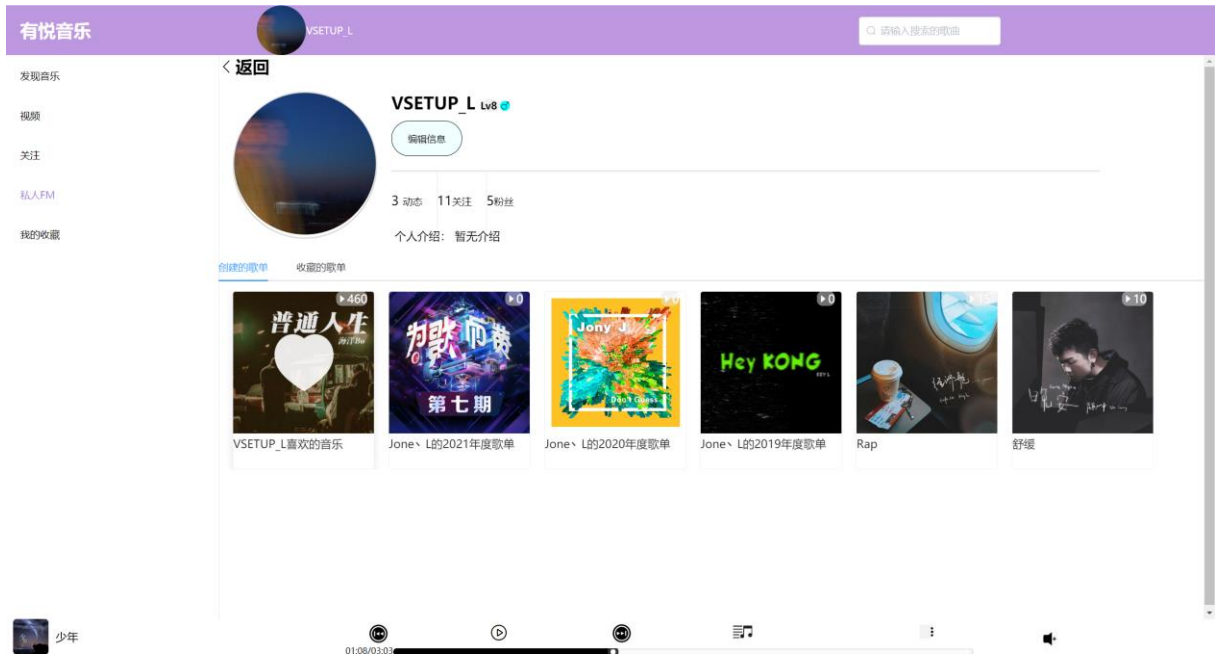


图 5-25 登录用户信息

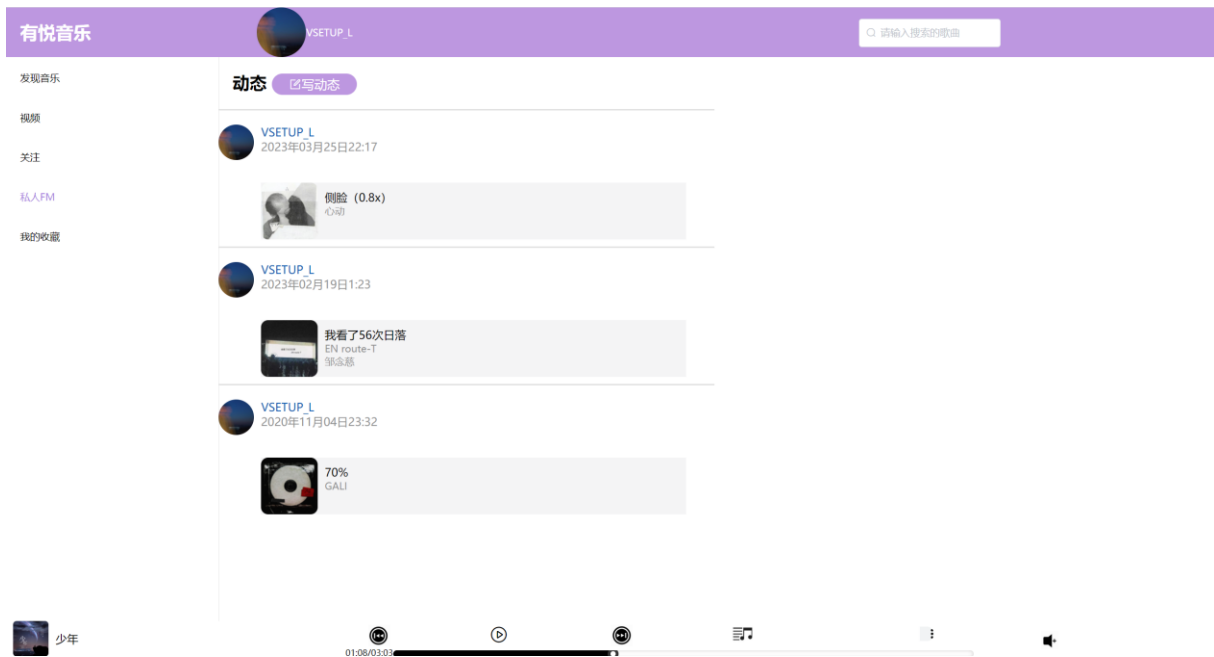


图 5-26 用户动态

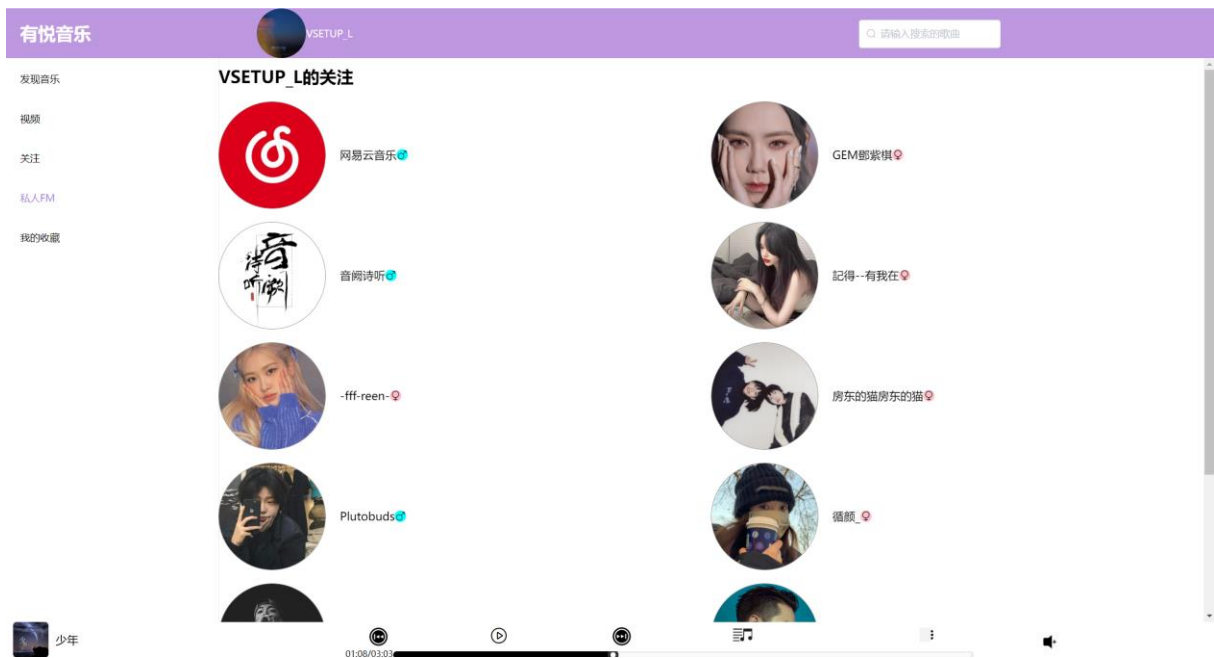


图 5-27 用户关注

### 5.9 用户收藏

根据用户的登录状态可以获取用户的歌曲收藏内容，未登录状态不提供收藏歌曲功能并提示用户需要登录才可以操作，如图 5-28 所示。

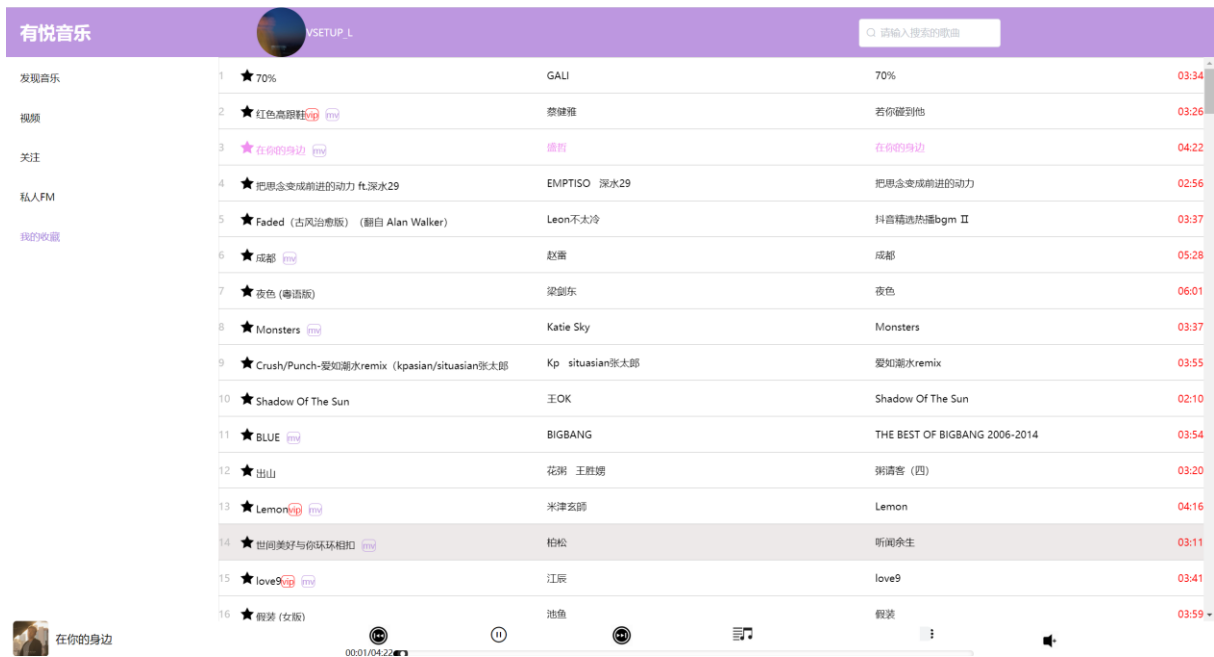


图 5-27 用户关注

6 系统测试

设计开发一款 web 应用完成开发的时候是需要进行系统测试的，测试的目的是为了发现该系统是否有代码缺陷、逻辑错误、以及性能缺陷。检测系统的功能配置是否有错，功能是否可以正常运行，对功能和性能需求等方面着重测试。一旦发现系统有漏洞的时候就需要从开发的代码阶段进行修复。一个好的应用上线是需要不断的测试不断的优化，这样才可以提前解决问题降低上线系统的时候出现的问题概率。根据上述分析，系统的测试是非常有必要的。以下是对本设计网页版音乐播放器的功能进行测试。对页面进行登录后的测试效果如表 6-1、表 6-2 所示。

表 6-1 登录状态测试表

| 功能名称    | 测试步骤   | 预期结果  | 测试结果    |
|---------|--|---|---------|
| 登录和注销登录 | (1).点击页面的登录字体通过扫码登录或者输入账号密码进行登录<br>(2).点击头像的退出登录按钮         | 扫码成功或者输入正确的账号密码可以跳转到主页面，点击头像注销可以退出登录                  | 与预期结果一致 |
| 搜索      | 点击页面的搜索框输入搜索内容按回车键   | 跳转到搜索页面返回搜索结果   | 与预期结果一致 |
| 评论      | (1).在歌曲或者歌单的评论输入框输入内容<br>(2).点击评论按钮                        | 输入内容后点击评论按钮对于登录的用户可以发送评论                              | 与预期结果一致 |
| 个人信息编辑  | (1).登录的用户点击头像<br>(2).点击编辑信息按钮<br>(3).输入修改的个人信息<br>(4).点击确定 | 用户点击头像会进入用户信息页面登录的用户会有编辑信息按钮输入修改的信息后点击提交可以修改登录用户的个人信息 | 与预期结果一致 |

|      |  |  |         |
|------|--|--|---------|
| 动态分享 | (1).点击关注<br>(2).点击写动态按钮<br>(3).输入需要发送的动态内容<br>(4).点击确定按钮 | 跳转到关注页面点击写动态按钮后弹出输入动态内容框输入完后点击确认可以发送动态页面展示动态内容 | 与预期结果一致 |
|------|--|--|---------|

70%  
测试未登录状态页面如表 6-2 所示。

表 6-2 未登录状态测试表

| 功能名称 | 测试步骤                                | 预期结果                                     | 测试结果    |
|------|-------------------------------------|--|---------|
| 搜索   | 55%<br>点击页面的搜索框输入搜索内容按回车键           | 跳转到搜索页面返回搜索结果                            | 与预期结果一致 |
| 评论   | (1).在歌曲或者歌单的评论输入框输入内容<br>(2).点击评论按钮 | 输入内容后点击评论按钮弹出提示用户未登录内容未发送成功              | 与预期结果一致 |
| 动态分享 | 点击关注                                | 55%<br>跳转到关注页面弹出提示用户未登录<br>用户点击确认跳转到登录页面 | 与预期结果一致 |

7 总结

对本次的网页版的音乐播放器开发设计对前后端分离式开发有了更深层次的了解，对前端和后端的数据交互浏览器的跨域问题有了更高水平的认识，对计算机网络知识、HTTP 协议、HTML、CSS、Javascript、Vue 框架、Axios、逻辑处理、node、网络安全等技术有了新的巩固。在完成本设计的开发也发现了自身的一些不足之处，对于一些 HTML 的标签属性还不够熟悉，CSS 的动画属性不够牢固，JS 的代码编写还不够效率和规范。通过在本次设计的过程中不断查询信息，不断学习内容来补充知识的薄弱部分，对前端的 vue 框架进行巩固知识，不断强化自身的专业知识。41%  
日渐熟悉了前后端分离式开发的流程，大大提高了开发的技术水平。也了解了互联网的音乐播放器的运行流程。

## 鸣 谢

大学的四年的美好时光已经悄然拉下了帷幕，也将迎来毕业的时间，在大学的四年时间里学会了很多也认识了很多朋友还有老师，在设计完成这次的网页版音乐播放器的开发，巩固了一些计算机网络知识，对专业知识的掌握得到了很大的效果，对前后端分离式开发的模式有了更加深入的了解，在此期间不断学习不断提高自身的知识水平，在完成设计的开发离不开老师的指导。很感谢有老师的耐心指导不然本设计开发也不会这么容易完成，感谢老师的在这次开发设计上的帮助。同时也感谢同学在本设计开发提出的问题解答帮助。接下来的时间我会努力前行，努力提高自身水平，克服更多的困难。

## 参考文献

- [1] Frain B . Responsive Web Design with HTML5 and CSS3[J]. Packt Publishing Limited, 2012.
- [2]（美）布拉德福（Bradford, A • ）（美）海涅（Haine, P • ）著；高京译.深入理解 HTML5 语义、标准与样式[M]. 北京：电子工业出版社,2013
- [3]Eric A.Meyer, Estelle Weyl 著.CSS 权威指南 第 4 版下[M].北京：中国电力出版社,2019
- [4]David Flanagan 著；张铭泽译.JavaScript 权威指南[M].北京：中国电力出版社,2001
- [5]梁灏编著.Vue.js 实战[M].北京：清华大学出版社,2017
- [6]Li M , Hu J , Lin X . The Development of Web Application Front-End of Intelligent Clinic Based on Vue.js[C]// Chinese intelligent automation conference. 2020.
- [7]Claudiu, Popescu. Learning Node.js: a hands-on guide to building web applications in JavaScript[J]. Computing Reviews, 2014.
- [8]史济民 顾春华 郑红. 软件工程:原理,方法与应用-第 3 版[M]. 高等教育出版社, 2009.
- [9]杜艳美.基于 web 前端的性能优化框架模型研究[D].西南科技大学,2018
- [10]孟祥双.前后端分离式 WEB 应用开发研究[J].电子元器件与信息技术,2019,第 3 卷(6): 40-43



## 附录

以下截取前端部分的主要一些逻辑处理。

(1) 函数防抖的封装，解决用户在短时间内连续多次点击按钮事件发送请求导致服务器资源消耗的问题。

```
//防抖函数
fangdo(fn, wait) {
  let time = null
  return function () {
    if (time) {
      clearTimeout(time);
    }
    let firstclick = !time
    if (firstclick) {
      fn.apply(this, arguments);
    }
    if (time !== null) {
      clearTimeout(time)
    }
    time = setTimeout(() => {
      time = null
    }, wait)
  }
},
```

图一 防抖函数的封装

(2) 对歌词部分进行分割实现歌词随音乐播放滚动的效果。

```
getarr(songlyric) {  
  //处理歌词  
  let arr = songlyric.split(/[(\r\n)\r\n]+)/).map((e, i) => {  
    let m = e.slice(1, 3); //分钟  
    let s = e.slice(4, 6); //秒  
    let ms = e.slice(7, 10); //毫秒  
    let lrc = e.slice(11, e.length); //歌词  
    let time = parseInt(m) * 60 * 1000 + parseInt(s) * 1000 + parseInt(ms); //时间  
    if (isNaN(Number(ms))) {  
      ms = e.slice(7, 9);  
      lrc = e.slice(10, e.length);  
      time = parseInt(m) * 60 * 1000 + parseInt(s) * 1000 + parseInt(ms)  
    }  
    return { m, s, ms, lrc, time };  
  });  
  arr.forEach((e, i) => { //e是数组元素, i是元素下标  
    if (i === arr.length - 1) {  
      e.next = 0  
    } else {  
      e.next = arr[i + 1].time  
    }  
  });  
  return arr  
},
```

图二 歌词的滚动处理

(3) 对一些重用函数的封装方便调用。

```

changetime(num) {
  var time = new Date(num);
  var y = time.getFullYear();
  var m = (time.getMonth() + 1 < 10
    ? "0" + (time.getMonth() + 1)
    : time.getMonth() + 1);
  var d = (time.getDate() < 10 ? "0" + time.getDate() : time.getDate());

  return y + "年" + m + "月" + d + "日";
},
changetime2(num) {
  var time = new Date(num);
  var y = time.getFullYear();
  var m = (time.getMonth() + 1 < 10
    ? "0" + (time.getMonth() + 1)
    : time.getMonth() + 1);
  var d = (time.getDate() < 10 ? "0" + time.getDate() : time.getDate());
  let h=time.getHours()
  let min=time.getMinutes()
  if(min<10){
    min='0'+min
  }
  return y + "年" + m + "月" + d + "日"+h+":"+min;
},
changesongtime(num) {
  let m, s;
  m = String(parseFloat(num) / (1000*60)).slice(0,1);

  s = String(num / 1000-parseInt(m)*60).slice(0,2);
  // console.log(parseInt(s),'@@@')
  if(parseInt(m)<10){
    m='0'+m.slice(0,1)
  }
  if(parseInt(s)<10){
    s='0'+s.slice(0,1)
  }
  return m + ":" + s;
},
changePlaycount(num) {
  if (num > 100000000) {
    return (num / 100000000).toFixed(1) + "亿";
  } else if (num > 10000) {
    return (num / 10000).toFixed(1) + "万";
  }
},

```

图三 常用的函数封装

57%

(4) 设置代理解决浏览器跨域的问题。

```
# 使用 VUE_APP_ 开头的变量会被 webpack 自动加载
# 定义请求的基础URL, 方便跨域请求时使用
VUE_APP_BASE_API = '/dev-api'
# 接口服务地址, 以你自己的为主
VUE_APP_SERVICE_URL = 'http://localhost:3000/'
#端口号
VUE_APP_POST=8080;
```

图四 设置代理

```
proxy: {
  // 匹配 /dev-api 开头的请求,
  [process.env.VUE_APP_BASE_API]: {
    // 目标服务器, 代理访问到 https://localhost:8001
    target: process.env.VUE_APP_SERVICE_URL,
    // 开启代理: 在本地会创建一个虚拟服务端, 然后发送请求的数据,
    // 并同时接收请求的数据, 这样服务端和服务端进行数据的交互就不会有跨域问题
    changOrigin: true, //开启代理
    pathRewrite: {
      // 会将 /dev-api 替换为 '', 也就是 /dev-api 会移除,
      // 如 /dev-api/db.json 代理到 https://localhost:8080/db.json
      ['^'+process.env.VUE_APP_BASE_API]: '/',
    }
  },
  '/weibo':{
    target:'https://weibo.com/',
    changOrigin:true,
    pathRewrite:{
      '^/weibo': '/'
    }
  }
}
```

62%

图五 设置代理解决跨域问题