

目 录

1. 绪论	
1.1 课题研究的目的	(- 1 -)
1.2 课题研究的意义	(- 1 -)
1.3 国内外现状	(- 2 -)
2. 开发环境及系统使用框架概述	
2.1 开发环境和技术介绍	(- 3 -)
2.2 开发工具介绍	(- 4 -)
3. 系统需求分析	
3.1 功能需求	(- 5 -)
3.2 性能需求	(- 5 -)
4. 系统详细设计	
4.1 数据库设计	(- 6 -)
4.2 系统结构设计	(- 8 -)
4.3 前台功能设计	(- 9 -)
4.4 管理员后台功能设计	(- 10 -)
5. 系统实现	
5.1 前台用户端实现	(- 12 -)
5.2 后台管理员端实现	(- 16 -)
5.3 后端关键代码展示	(- 19 -)
6. 系统测试	
6.1 测试目的	(- 20 -)
6.2 测试用例	(- 20 -)
6.3 测试结果截图	(- 21 -)
7. 总结	(- 26 -)
参考文献	(- 27 -)
致谢	(- 28 -)

基于 JavaWeb 的音乐分享平台的设计与实现

1. 绪论

1.1 课题研究的目

随着计算机科学技术的发展和社会的进步，人们也逐步进入信息化时代，现代人得益于网络和计算机的普及，生活也变得越来越丰富多彩，大家可以在互联网上尽情享受生活，让大家有一个休闲放松的场所。故而，各种各样的文娱网站逐渐出现在我们是生活中，在这样的大时代环境的背景下，音乐网站正是其中重要的一员。从宏观角度看，在高速发展的互联网中，各种技术更新换代越来越快，音乐网站想要立足于历史大潮的洪流之中而不倒，就必须要跟紧时代的步伐不断革新与进步，不断提升功能的强化和提高用户的好感度，增强后端系统的稳定，增强可维护性和健壮性。因为电脑的普及，越来越多的人使用电脑上网，带动了音乐网站产业的飞速发展。再者，在当前世界对音乐的热诚追逐的过程中，而导致音乐大潮的影响下，随着乐坛一次又一次的取得重大成绩，优秀的音乐家和歌手们引领世界风向，音乐已经与我们的日常生活紧密相连。在广大网民的需求下，音乐网站势必要从萌芽走向繁荣，成为一种新兴交流平台来替代传统交流方式，致使传统音乐产业的产业结构的调整。然而，现存的一些音乐网站技术已经非常落后，业务功能已经不能满足用户需要，后台系统也需要进一步的更新和优化。所以说设计并实现这样的一个满足需求的在线音乐网站就非常有必要了，其必要性和紧迫性不言而喻，我们的网站不仅要在使用的功能和系统的性能上做到与时俱进，保证在速度上不能慢的同时，还旨实现技术产业的更新^[1]。

1.2 课题研究的意义

当今世界，计算机软件技术的飞速发展和社会的进步，人们的生活水平大大提高，为满足人们日益增长的精神文明需要，意义网站系统因此也飞速成长壮大，音乐网站作因其普片大众、方便快捷而被广大网民所推崇接受，作为一种相对完善的平台而活跃在当今世界的互联网络上，给音乐爱好者提供了更多考虑问题的方向^[2]。但正如前文所说，互联网技术的快速的迭代更新，产品的不断升级，导

致我们必须要考虑的是新事物必将取代旧事物，新的功能更完善、设计更完善、性能更完美的音乐网站必将淘汰掉旧产品，不论是前台页面显示的设计，还是后台数据传输的速度，甚至是网站的安全性都已经更不是时代的步伐。故此，为了满足新生代用户的需要，也是为了软件技术的发展，设计并实现新的音乐网站系统就很有应验了，可以更加方便、更加快速、更加简单、更加安全地实现相关业务需求，并实现更多的功能来满足实际情况的需要。本系统采用了前端和后端分离的项目整体架构开发方式，前端采用 Vue 框架+ElementUI，而后端采用了 SpringBoot 微服务框架技术，不仅沿用了经典的 MVC 业务逻辑，也采用了新设计理念来实现，在经过资料调查、整理与分析之后，找到一种相对完善的解决方案来开发新的音乐网站^[3]。

1.3 国内外现状

步入新世纪，现代化信息化进程加快，音乐网站的蓬勃发展的时期也随之到来。然而，中国在音乐网站的发展都很落后，且内地市场在很多年来，都比较萧条，在十年前的很长一段时间都发展相对缓慢，皆是由于近年来来自国际社会上美国模式的影响和其对全世界市场的冲击，以及众多版权问题冲突不断。直至 2012 年，中国的音乐网站市场因为互联网的兴起和普及，才开始逐渐的发展起来，不断进步，用户和音乐爱好者们对音乐网站的功能的使用次数的已经超过了数亿次，并且在内地的很多现存的音乐网站，也得以快速发展壮大，它们已经逐步完成产业的优化，能够给予音乐爱好者提供较为完善的服务和功能。慢慢的，在国内多方互联网公司的努力下，中国的音乐网站产业也逐步追上时代的脚步，音乐网站市场正逐步展现出他强大的活力和潜力。现如今，中国音乐市场的发展已经成熟完善起来并一步步突破新的瓶颈。从网站的功能和内容这些方面来看，也逐步走向成熟，已经逐步开始形成了成体系的产业链，作品来源的渠道也拓宽了道路。在网络支持方面，各大运营商的移动网络和宽带网络飞速发展进步。并且在电子产品和消费服务方面也均赢得了大家的认可，不断的推出新的产品器件，一代代完善其设计，构建新消费理念并提供方案。渐渐的，中国音乐网站，也因此得到快速的发展，慢慢的开始赶超世界先进水平。

2. 开发环境及系统使用框架概述

2.1 开发环境和技术介绍

2.1.1 Java 语言

Java 是 Sun 公司在 1995 年推出的一种编程语言。Java 是当前互联网行业最流行的面向对象的编程语言，在软件系统的集成编写上有着巨大优势，并且由于应用广泛，使用的开发人员众多，java 也在逐步完善，常年位居编程语言排行榜首，受到业内人员的广泛推崇。且 Java 语言比较于其他编程语言而言又比较简单，又具有灵活性、安全性以及可移植性等特点^[4]。因此使用 java 语言来开发不仅在效率和性能上胜过其他语言，而且又易于部署和维护。由于它优秀的特性，程序开发者们只需要专注于写代码，就能做成优秀的成果^[5]。

2.1.2 Node.js、Vue 和 ElementUI

node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境。Vue 是一种渐进式的框架。Vue 核心库让开发人员只需关注其视图层，并且在其它技术整合方面非常的便利高效。另外，Vue 驱动采用单文件组件，Vue 生态完美支持开发的复杂单页应用^[6]。ElementUI 是一套为设计、开发人员准备的组件库，它是基于 Vue 的桌面端，这些 UI 库的差异都不是很大，基本上都是 Material Design 风格，ElementUI 方便我们做前端的开发设计^[7]。本音乐网站系统使用的是 Node12.9 和 Vue2.9 的版本。

2.1.3 JavaScript

JavaScript（简称“JS”）是一种解释型编程语言^[8]。JavaScript 是脚本语言，是一种解释性脚本语言，是一种轻量级的编程语言，是可插入 HTML 页面的编程代码，插入 HTML 页面后，可由所有的现代浏览器执行。为前端网页添丰富多彩的效果^[9]。

2.1.4 SpringBoot 框架

Spring Boot 是由 Pivotal 团队提供的全新的一种微服务框架，它旨在简化

Spring 应用的繁琐的各种配置。该框架仅仅需要简单的配置即可使用以前繁琐配置的所以功能,进而使得开发人员不再在那些繁琐重复的配置项上花费不必要的时间和精力^[10]。这样一来, SpringBoot 很快发展成为快速应用开发领域的先行者和领导者, Springboot 带来了全新的自动化配置解决方案^[11]。

2.1.5 MyBatis

MyBatis 本是 apache 的一个开源项目 iBatis, 2010 年这个项目由 apache software foundation 迁移到了 google code, 并且改名为 MyBatis, 是一个基于 Java 的持久层框架^[12]。它让开发人员可以不再在 JDBC 代码上耗费过多非必要的时间和精力, 同时避免了手动设置参数和获取结果集。MyBatis 只需要通过简单的配置和注解即可映射 java 对象, 让面向数据库的持久层开发变得更加方便, 同时也让代码维护更加方便^[13]。

2.2 开发工具介绍

2.2.1 IntelliJ idea

IntelliJ IDEA, 是 JetBrains 公司开发的一种面向企业级 java 开发的产品, 是当前市面上最优秀的 java 语言开发工具。它拥有丰富的插件和第三方插件, 为开发人员提供了极大的便利, 支持各类工具的整合, 大大提高了程序员的工作效率^[14]。

2.2.2 Visual Studio Code

Visual Studio Code (简称“VS Code”)是微软公司在 2015 年 4 月 30 日发布的一款开发工具, 针对于编写现代 Web 和云应用的跨平台源代码编辑器, 可在桌面上运行, 并且可用于 Windows, macOS 和 Linux^[15]。

2.2.3 Navicat

“Navicat”是一款非常优秀的数据库管理工具, 它可以让开发测试人员很方便的查看与管理各种不同类型的数据库, 并且可以通过 Navicat 来操作云数据库。Navicat 拥有强大丰富的功能, 能过满足开发人员需要, 极大的提高了开发人员的工作效率, 界面设计也非常友好^[16]。

3. 系统需求分析

3.1 功能需求

本系统的前台界面需要满足用户的相关需要，总体上要实现以下功能使得系统更加完善：

- (1) 注册新用户的模块：新用户需要输入包括用户的用户名、用户对应的密码、联系电话、性别、年龄、email 等信息将其保存在数据库然后完成注册；
- (2) 用户登陆模块：已注册的用户输入自己的用户名和密码，经验证通过才能登陆；
- (3) 用户评论音乐功能，登录后方能评论歌单，未登录不能使用评论功能；
- (4) 用户收藏功能，登录后方能使用收藏功能，未登录不能使用收藏功能；
- (5) 用户播放音乐功能等。

本系统的后台管理员的功能需求，要实现以下基本模块：

- (1) 管理员后台登陆功能：管理员使用正确的用户名和密码经验证通过，方能成功登陆进入管理员界面；
- (2) 管理用户信息：管理员权限应有查看信息功能按钮、删除信息功能按钮；
- (3) 管理歌单信息、歌手信息：管理员应有权限查看、删除和修改歌单歌手信息；
- (4) 歌曲、歌手和用户的可视化数据图表。

3.2 性能需求

本系统要保证性能良好以便系统的正常运行，总体上要满足以下性能需求：

- (1) 前端显示的界面设计的合理性：系统的界面设计要简洁大方，符合当代广大网民大众的审美标准，方便操作，布局合理，各功能按钮简明清晰、一目了然，对于初级用户容易上手，功能完善，实现需求分析的全部功能；
- (2) 信息在系统前端展示的完整性和安全性：系统要保证信息的准确性、安全性、有效性，必要时对用户给出及时的反馈，并且响应速度要快速；
- (3) 系统在主流的浏览器上都能流畅运行，便于用户使用；
- (4) 系统开发要符合一般的开发标准，便于系统的维护，方便在后续增加一些新的功能需求和二次开发。

4. 系统详细设计

4.1 数据库设计

一个好的软件系统的设计是从设计数据库系统开始进行的,数据库要存储系统中各数据对象的值、属性以及数据直接的关系。数据库的逻辑结构和存储结构的设计与完成,是本系统网站搭建的开始,也是其中的核心,设计的数据库必须要合理、完整,且易于维护。不仅要展现系统中需要的所有数据,还要让他们之间的联系组成一个整体,进而完成一个关系模型完善的数据库来为系统软件开发服务,使得系统可以准确、方便地操作数据库,获取所需的对应的信息。同样的,数据库合理、准确、完整的设计是软件工程开发与设计的起点与基石,在开发的过程中,从始至终都发挥着极为重要作用。

根据系统需求分析,为满足系统的数据层的需要,经过设计与整理,本系统一共设计了管理员表(admin)、收藏信息表(collect)、评论信息表(comment)、用户信息表(consumer)、歌单歌曲表(list_song)、评分表(rank)、歌手信息表(singer)、歌曲信息表(song)和歌单信息表(song_list)。它们相互协同,存储了本音乐网站系统所需要的全部数据信息,各数据表之间相互独立又紧密联系,共同完成数据支持任务,使得系统能够按照需求分析的相关要求得以正常运行,为系统设计打下了坚实的基础。

表结构如下:

表: 4-1 管理员表

admin					
字段名	数据类型	长度	可否为空	约束	备注
id	int	11	NOT NULL	主键	主键
name	varchar	255	NOT NULL		账号
password	varchar	255	NOT NULL		密码

表: 4-2 收藏信息表

collect					
字段名	数据类型	长度	可否为空	约束	备注
id	int	11	NOT NULL	主键	主键
user_id	varchar	255	NOT NULL		用户 id
type	tinyint	1	NULL		收藏类型 (0: 歌单; 1: 歌曲)
song_id	int	11	NULL		歌曲 id
song_list_id	int	11	NULL		歌单 id
create_time	datetime	0	NULL		收藏时间

表：4-3 歌单歌曲表

list_song					
字段名	数据类型	长度	可否为空	约束	备注
id	int	11	NOT NULL	主键	主键
song_id	int	11	NULL		歌曲 id
song_list_id	int	11	NULL		歌单 id

表：4-4 评论信息表

comment					
字段名	数据类型	长度	可否为空	约束	备注
id	int	11	NOT NULL	主键	主键
user_id	varchar	255	NULL		用户 id
type	tinyint	1	NULL		评论类型 (0: 歌单; 1: 歌曲)
song_id	int	11	NULL		歌曲 id
song_list_id	int	11	NULL		歌单 id
content	varchar	255	NULL		评论内容
create_time	datetime	0	NULL		收藏时间
up	int	11	NULL		评论点赞数

表：4-5 用户信息表

consumer					
字段名	数据类型	长度	可否为空	约束	备注
id	int	11	NOT NULL	主键	主键
username	varchar	255	NOT NULL		账号
password	varchar	255	NOT NULL		密码
sex	tinyint	1	NULL		性别 (1: 男; 0: 女)
phone_num	char	15	NULL		电话
email	char	30	NULL		邮箱
birth	datetime	0	NULL		生日
introduction	varchar	255	NULL		个性签名
location	varchar	255	NULL		地理位置
avator	varchar	255	NULL		头像
create_time	datetime	0	NULL		创建时间
update_time	datetime	0	NULL		更新时间

表：4-6 评分表

rank					
字段名	数据类型	长度	可否为空	约束	备注
id	int	11	NOT NULL	主键	主键
song_list_id	int	11	NULL		歌单 id
consumer_id	int	11	NULL		用户 id

表：4-7 歌手信息表

singer					
字段名	数据类型	长度	可否为空	约束	备注
id	int	11	NOT NULL	主键	主键
name	varchar	255	NOTNULL		姓名
sex	tinyint	1	NULL		性别（1：男；0：女）
pic	varchar	255	NULL		头像
birth	datetime	0	NULL		生日
location	varchar	255	NULL		地理位置
introduction	varchar	255	NULL		简介

表：4-8 歌曲信息表

song					
字段名	数据类型	长度	可否为空	约束	备注
id	int	11	NOT NULL	主键	主键
singer_id	int	11	NULL		歌手 id
name	varchar	255	NULL		歌名
introduction	varchar	255	NULL		简介
create_time	datetime	0	NULL		创建时间
update_time	datetime	0	NULL		更新时间
pic	varchar	255	NULL		歌曲图片
lyric	text	65535	NULL		歌词
url	varchar	255	NULL		歌曲地址

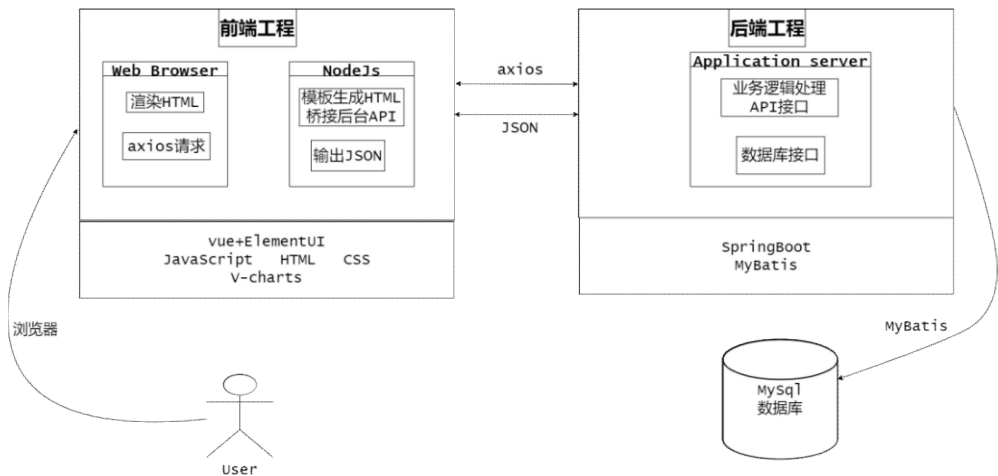
表：4-9 歌单信息表

song_list					
字段名	数据类型	长度	可否为空	约束	备注
id	int	11	NOTNULL	主键	主键
title	varchar	255	NULL		标题
pic	varchar	255	NULL		歌单图片
introduction	varchar	255	NULL		简介
style	varchar	255	NULL		歌单风格

4.2 系统结构设计

本音乐网站的系统使用了最新的前后端分离的 MVVM 框架(Model、View、View Model)，MVVM 实现了前后端更好的分离。ViewModel 是前端的视图模型，前端开发者对从后端获取的 Model 数据进行转换处理，做二次封装，以生成符合 View 层使用预期的视图数据模型^[17]。MVVM 是前后端分离方案实施的重要一环，使数据和界面得到很好的分离效果。数据库采用 MySQL 数据库，后端采用当下比较流

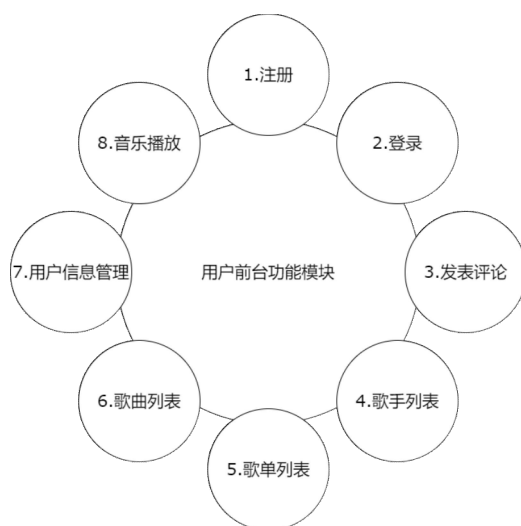
行的 SpringBoot 框架，通过 MyBatis 访问数据库，前端开发采用的框架是 Vue，Vue 是一套用于构建用户界面的渐进式 JavaScript 框架，体积小，在开发过程中，拥有更高的运行效率，其框架基于虚拟 DOM（Document Object Model），可以预先通过 JavaScript 进行各种计算，把最终的 DOM 操作计算出来并优化，同时开发者可以依赖于其框架的双向数据绑定特点，不用再去操作 DOM 对象，从而有更多精力投入到业务逻辑中去^[18]。引用一些可视化组件，如 V-charts，让数据更加直观地展示给用户，在开发过程中，后端主要负责业务逻辑处理和对数据进行处理^[19]，相对而言更加抽象，前端主要负责和用户进行交互，设计页面功能，调用后台 API 结构，实现相应功能，做出良好的页面，让用户更加高效的使用系统^[20]。本音乐网站系统架构图如下：



图：4-1 系统架构图

4.3 前台功能设计

本音乐网站系统的用户端要实现用户注册、登录、发表评论、歌手列表、歌单列表、歌曲列表、用户信息维护、音乐播放、收藏音乐、收藏列表等功能，包括增删改查等操作。如下图，图 4-2 所示即为系统前台用户功能模块图，他们包括八个大的功能板块，且契合需求分析所需功能要求，下面一一介绍。



图：4-2 系统前台用户功能模块图

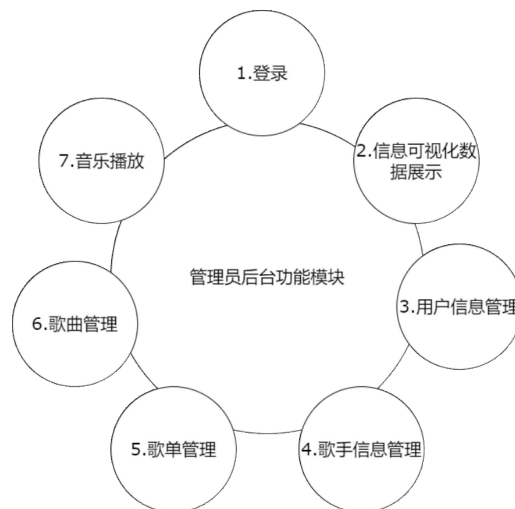
系统前台用户的主要功能是用户权限可以使用的全部功能，主要包括以下方面的内容：

- （1）注册：新用户需要输入包括用户的用户名、用户对应的密码、联系电话、性别、年龄、email 等信息然后完成注册，且用户名和密码是必填内容，其他信息为选填内容；
- （2）登录：用户输入正确的账号密码，经后端进行数据库验证成功方可登录，否则登录失败；
- （3）发表评论：用户登录后，可在歌单下面发表评论，若未登录则不能使用评论功能；
- （4）歌手列表：点击歌手按钮可查看歌手列表，点击歌手可查看歌手的歌曲作品，并进行音乐播放；
- （5）歌单列表：点击歌单按钮可查看歌单列表，点击歌单可查看歌单里的歌曲列表，并进行音乐播放；
- （6）歌曲列表：点击歌单可以查看歌单列表，即可进行音乐播放；
- （7）用户信息管理：用户登录后，点击设置可修改用户个人信息，填写完成后点击确定保存即可保存成功；
- （8）音乐播放：点击播放按钮或者点击歌曲列表里的歌曲可以播放。

4.4 管理员后台功能设计

本音乐网站系统的后台管理系统，要对前台用户系统的相关信息进行管理

维护，管理员端要实现管理员的登录、展示可视化信息图表、用户信息管理功能页面、歌手信息管理功能页面、歌单管理功能页面、歌曲管理功能页面、音乐播放功能页面等，其中主要包括一些增删改查的操作。如下图，图 4-3 所示即为系统后台的管理员权限所需要的主要功能，它们大致包括七个板块，完成的功能模块符合需求分析的总要求，下面一一概述。



图：4-3 系统后台管理员功能模块图

系统后台管理员功能主要是是管理员对前台用户的信息管理和网站资源文件的管理，主要包括以下几点：

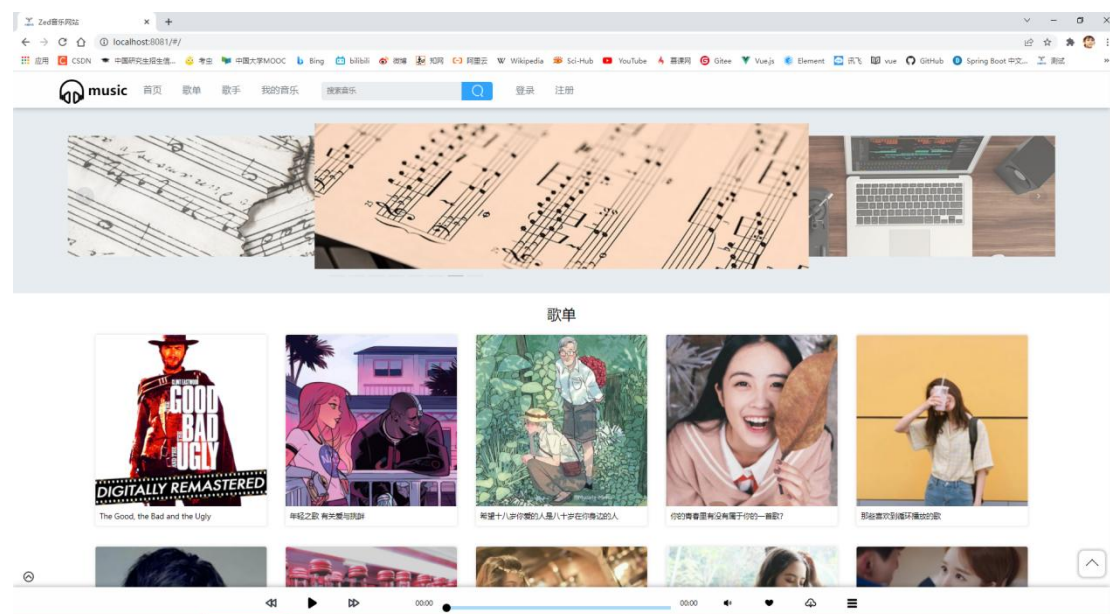
- （1）登录：只有输入的账号密码正确且一一对应，后端通过检测数据库比对正确，方可登陆成功，获取管理员权限，否则登录失败；
- （2）信息可视化数据展示：后台系统首页以图表的方式展示用户、歌手、歌曲等相关信息；
- （3）用户信息管理：管理员可修改用户个人信息，并保存到数据库中，进而改变前台用户的数据；
- （4）歌手信息管理：管理员可以添加、删除、修改歌手相关信息；
- （5）歌单管理：管理员可以添加、删除、修改歌单相关信息；
- （6）歌曲管理：管理员可以添加、删除歌曲；
- （7）音乐播放：管理员可以试听音乐。

5. 系统实现

5.1 前台用户端实现

5.1.1 音乐网站首页

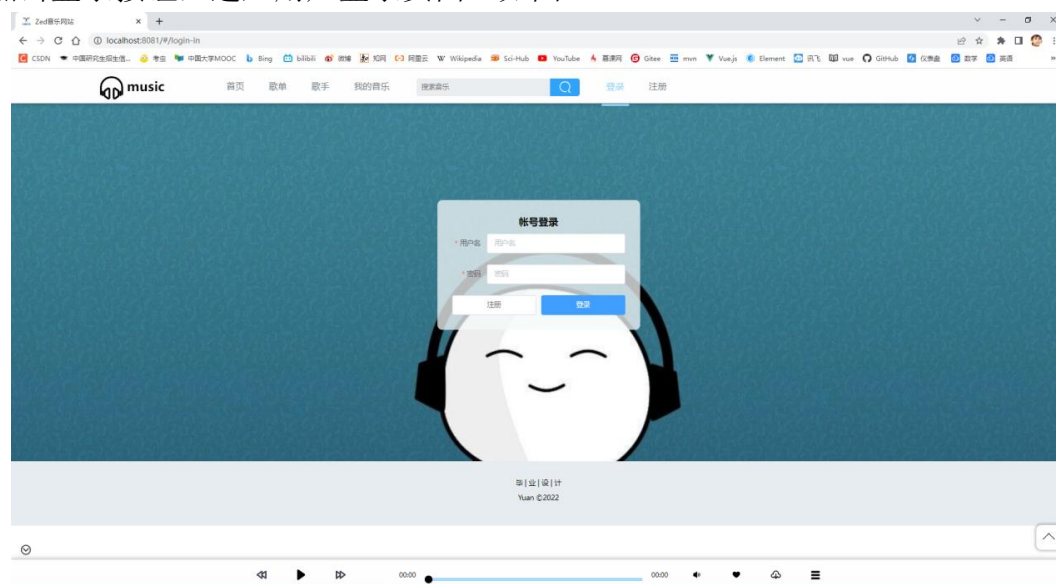
浏览器输入：localhost:8081 敲回车进入本音乐网站系统的首页。如图：5-1。



图：5-1 前台用户端首页

5.1.2 用户登录

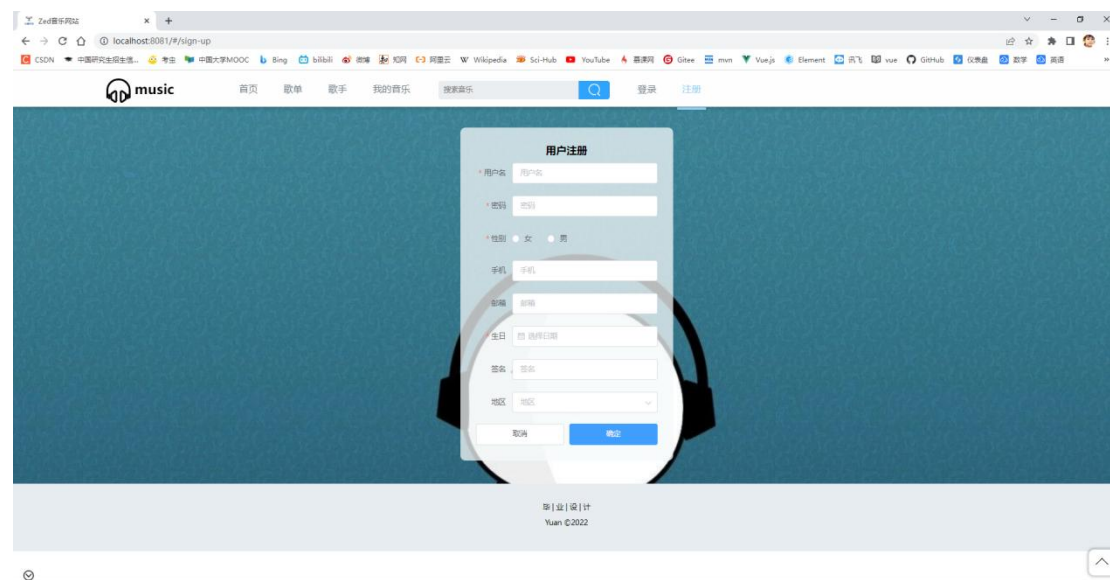
点击登录按钮，进入用户登录页面。如图：5-2。



图：5-2 用户登录页面

5.1.3 用户注册

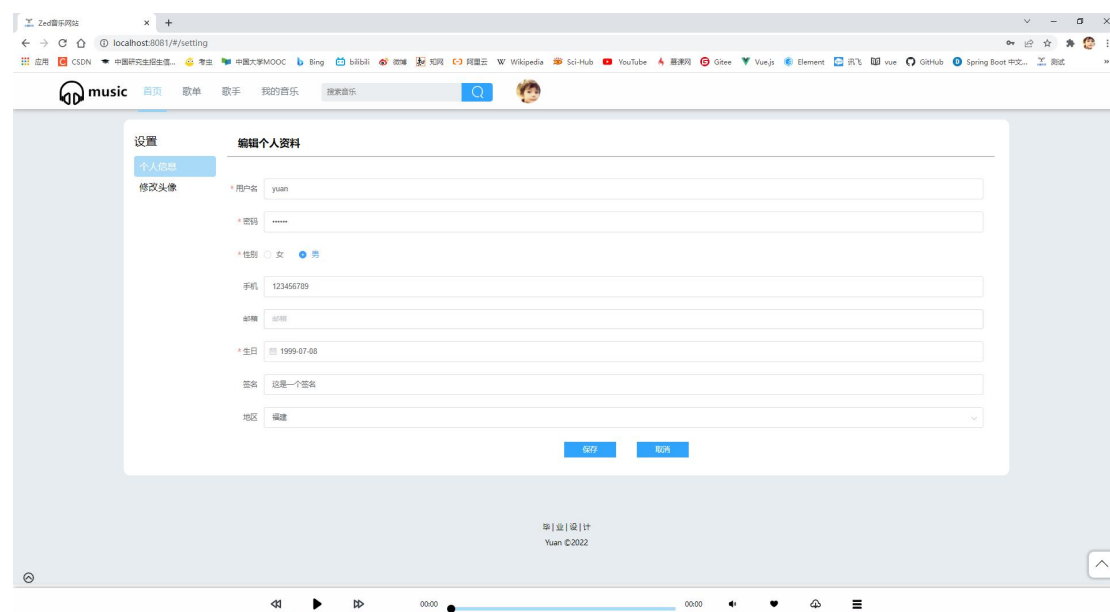
在页面上方的按钮栏，有汉字“注册”，点击该按钮，本系统的新用户即可进入用户注册页面填写相关信息进行注册。如图：5-3。



图：5-3 用户注册页面

5.1.4 用户信息维护

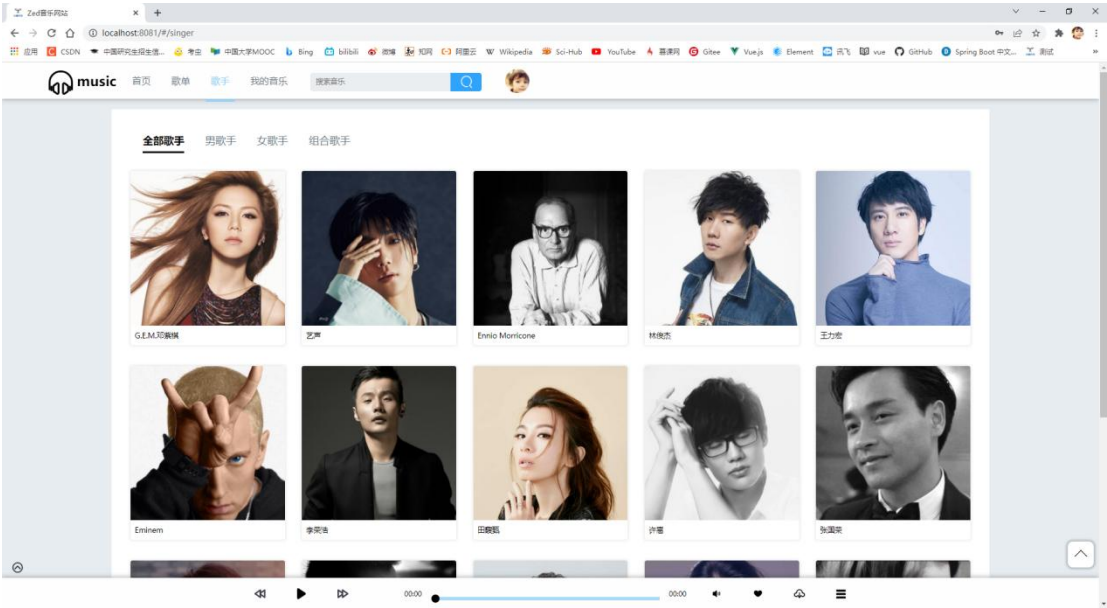
点击头像然后点击设置按钮，进入用户个人信息维护页面。如图：5-4。



图：5-4 用户个人信息维护

5.1.5 歌手列表

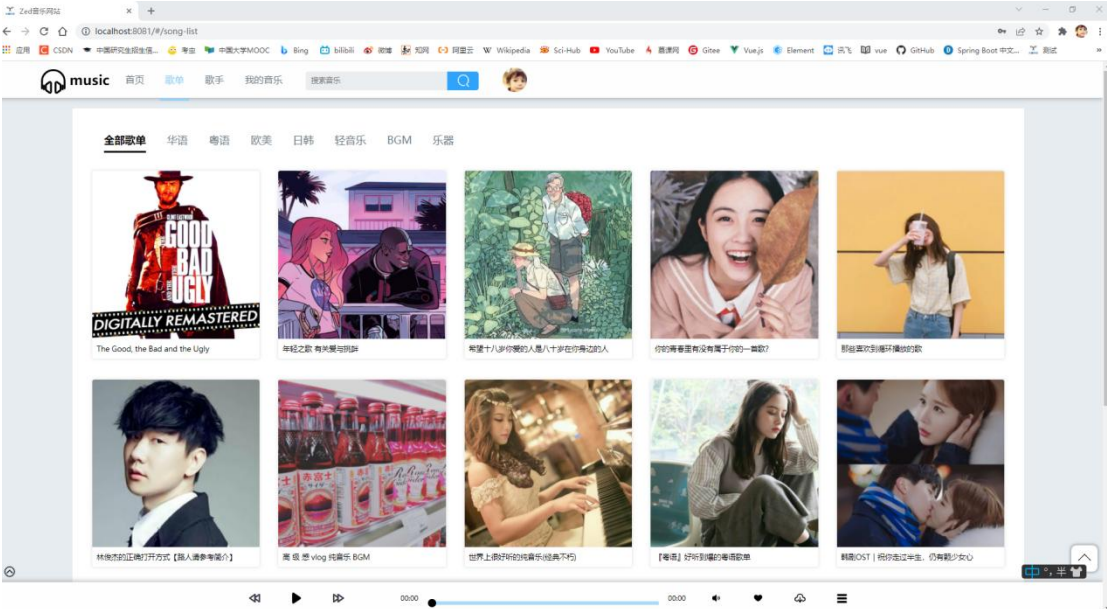
点击歌手或者滚动鼠标可以切换到歌手列表。如图：5-5。



图：4-5 前台歌手列表界面

5.1.6 歌单列表

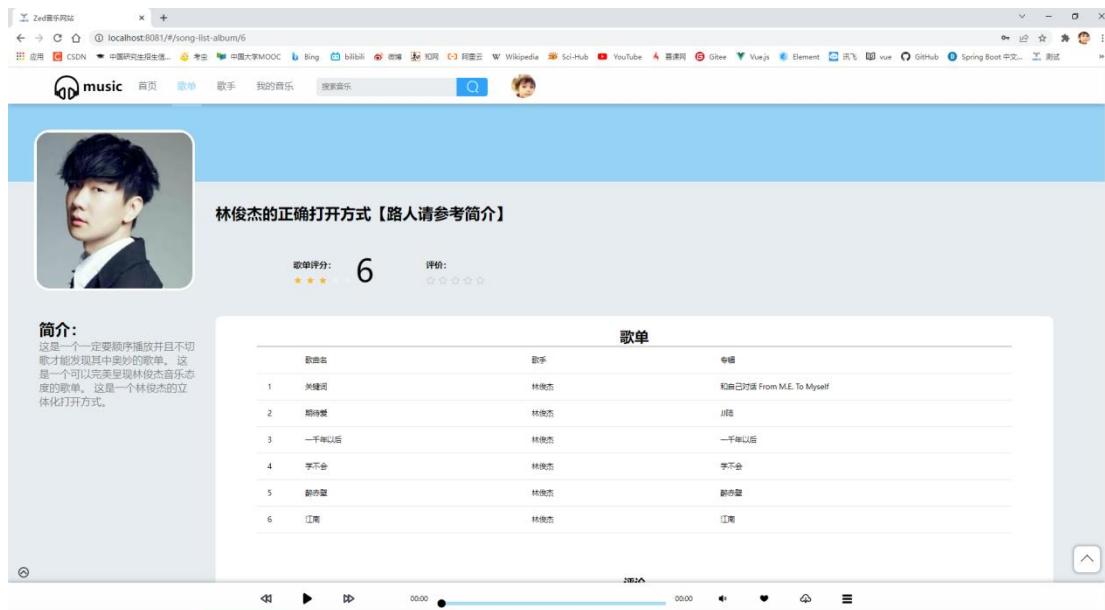
点击歌单按钮进入歌单列表界面。如图：5-6。



图：5-6 前台歌单界面

5.1.7 歌曲列表

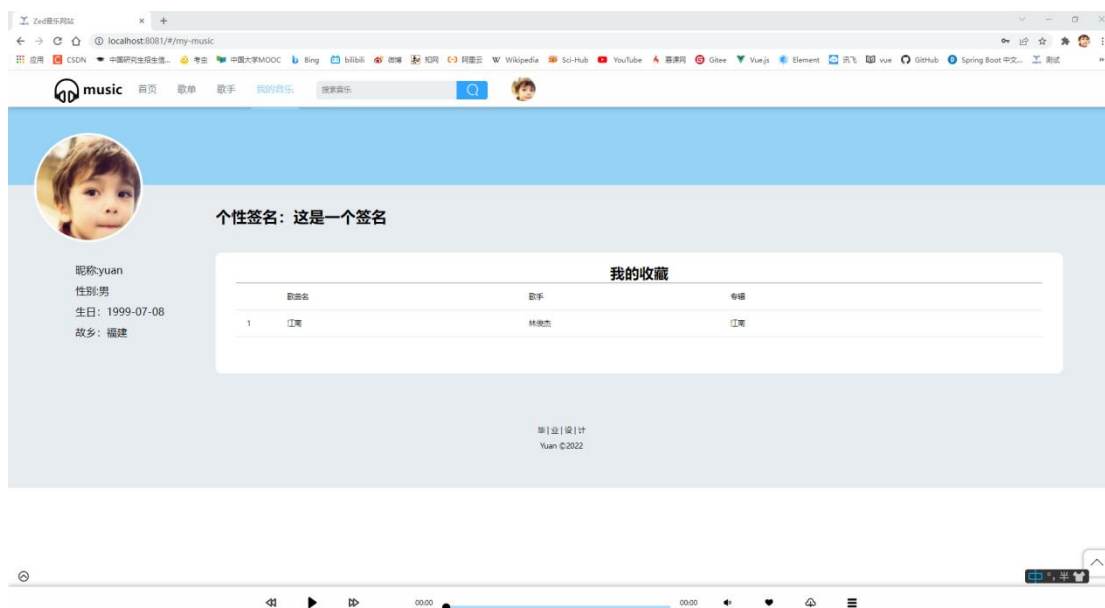
点击任意歌单进入对应的歌曲列表。如图：5-7。



图：5-7 前台歌曲列表

5.1.8 收藏

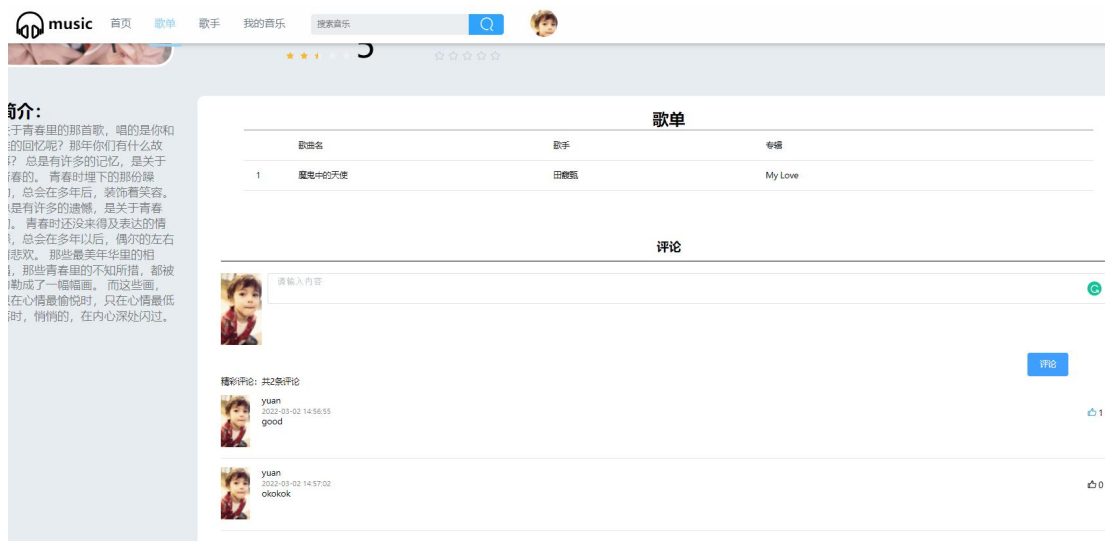
点击我的音乐按钮进入我的收藏界面。如图：5-8。



图：5-8 用户收藏夹

5.1.9 评论

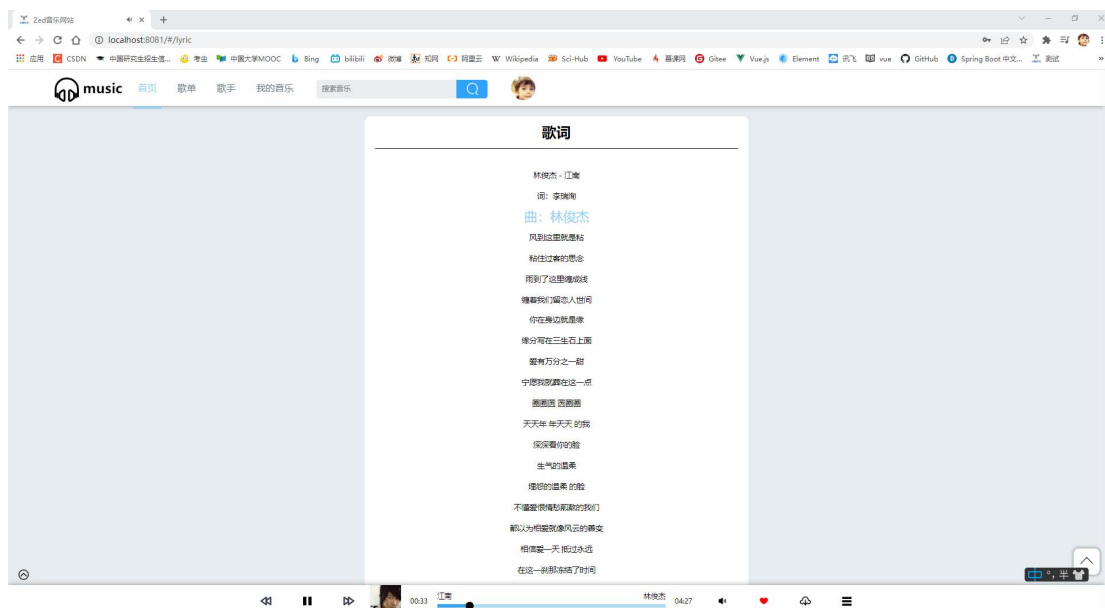
查看评论，如图：5-9。



图：5-9 用户评论

5.1.10 音乐播放

点击播放按钮可播放音乐，并可查看歌词。如图：4-10。

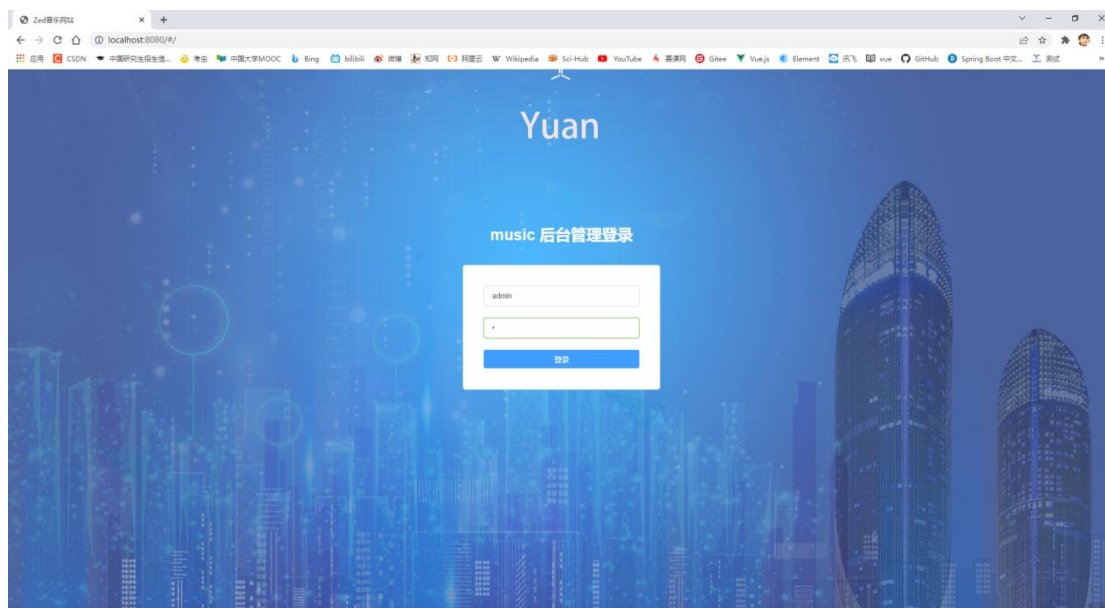


图：5-10 前台用户播放音乐

5.2 后台管理员端实现

5.2.1 后台管理员登录

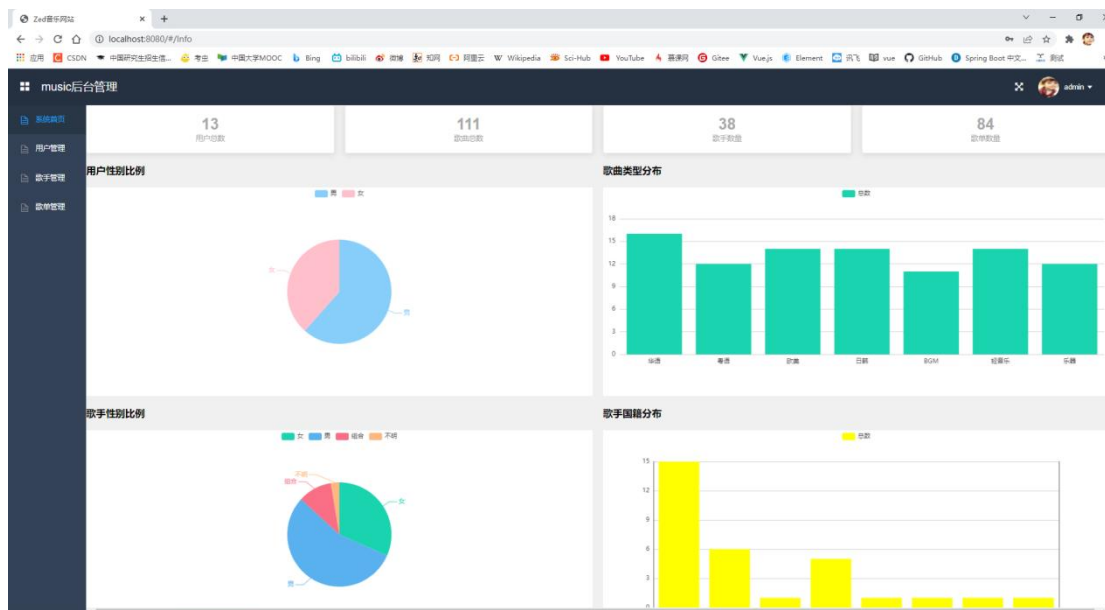
浏览器输入：localhost:8080 敲回车进入音乐网站首页。如图：5-11。



图：5-11 后台管理员登录页面

5.2.2 管理员首页和可视化数据图表

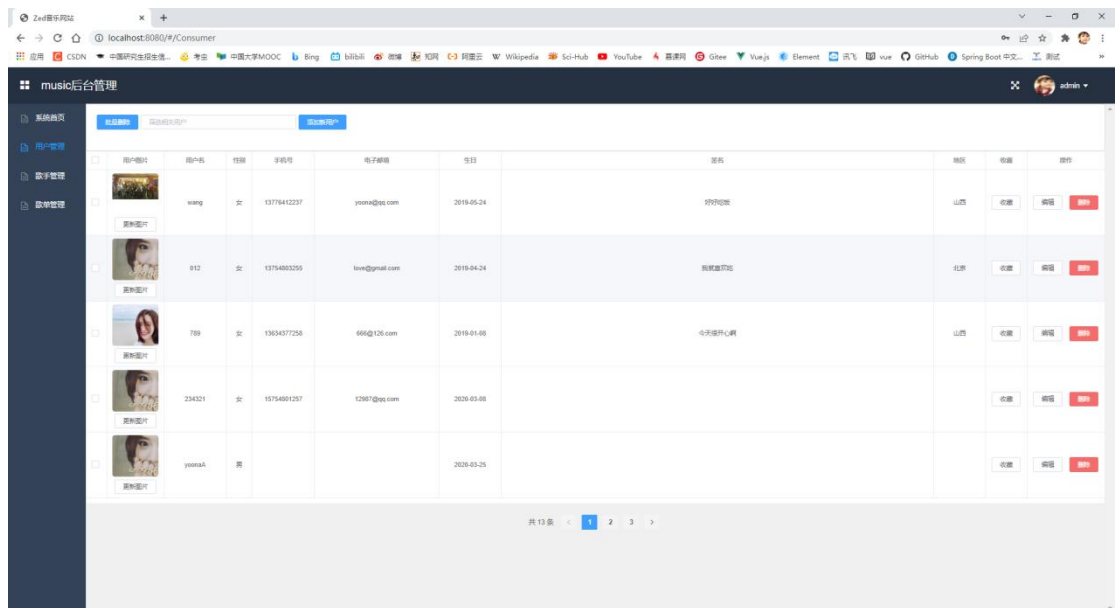
登录成功后，自动跳转到管理员首页，查看数据图表。如图：5-12。



图：5-12 管理员首页可视化数据图表

5.2.3 用户管理界面

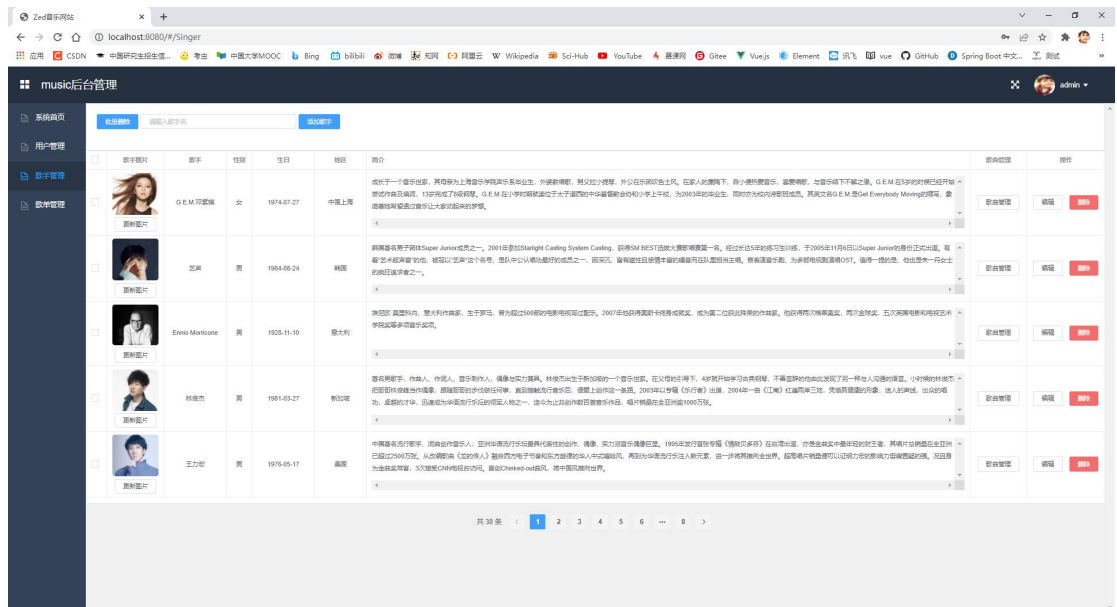
单击左侧用户管理选项，就会为管理员展现用户管理的相关操作按钮的用户信息，管理员即可对用户信息进行管理。如图：5-13。



图：5-13 用户信息管理

5.2.4 歌手管理界面

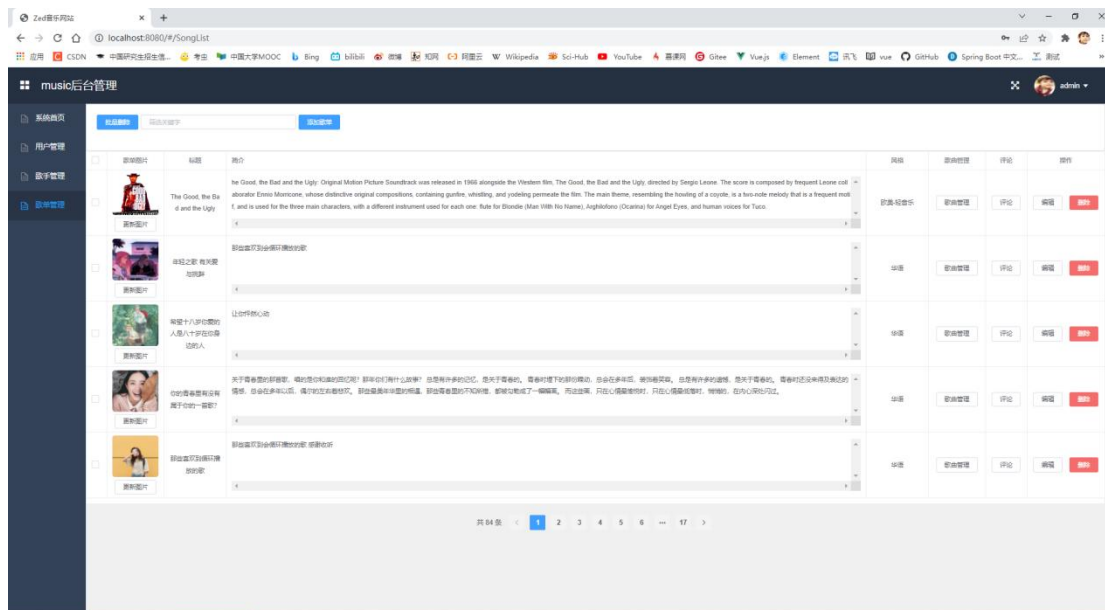
单击左侧歌手管理选项，就会为管理员展示歌手信息及相關操作按钮，管理员即可对歌手信息进行管理。如图：5-14。



图：5-14 歌手信息管理

5.2.5 歌单管理界面

点击歌单管理按钮进入歌单管理界面。如图：5-15。



图：5-15 歌单信息管理

5.3 后端关键代码展示

(1) 整合 Knife4j 相关代码如下：

```
@Configuration
@EnableSwagger2WebMvc
@EnableKnife4j
@ConfigurationProperties(prefix = "swagger.config")
public class Knife4jConfig {
    @Bean(value = "Zed 音乐网站 API 接口文档")
    public Docket defaultApi2() {
        return new Docket(DocumentationType.SWAGGER_2)
            .apiInfo(new ApiInfoBuilder()
                .title("Zed 音乐网站 API 接口文档")
                .description("# swagger-bootstrap-ui-demo RESTful APIs")
                .termsOfServiceUrl("http://localhost:8888/swagger-ui.html")
                .version("v2.0")
                .build())
            .groupName("2.X 版本") .select()
            .apis(RequestHandlerSelectors.basePackage("com.zky.music.controller"))
            .paths(PathSelectors.any())
            .build();
    }
}
```

6. 系统测试

6.1 测试目的

系统测试的种类很多，它应包括一系列的测试，如网络测试、硬件测试和软件测试等等。但我们在完成系统后所做的测试是指软件测试^[21]。系统测试是检验系统是否符合我们需求分析的初衷、是否完成功能模块的必要步骤，只有经过系统测试并且测试成功通过，一个完整的系统才算真正的完成，是一个系统开发最后进行的工作，是对整个系统最后的查漏补缺，帮助我们及时的发现问题、并解决问题。我们必须要进行系统测试，原因在于只要通过测试，才能找出系统中存在的未知的错误和程序在执行的过程中存在的问题与不足。测试的目的旨在能够用最小的代价发现系统中肉眼不可见的一系列的 bug。可根据开发的各个阶段的不同需求和不同情况来设计测试用例，并根据我们拟定好的这些测试用例来进行系统且全面的测试，以此来找出错误的内容。下面对系统的主要功能设计的测试用例，并完成测试。

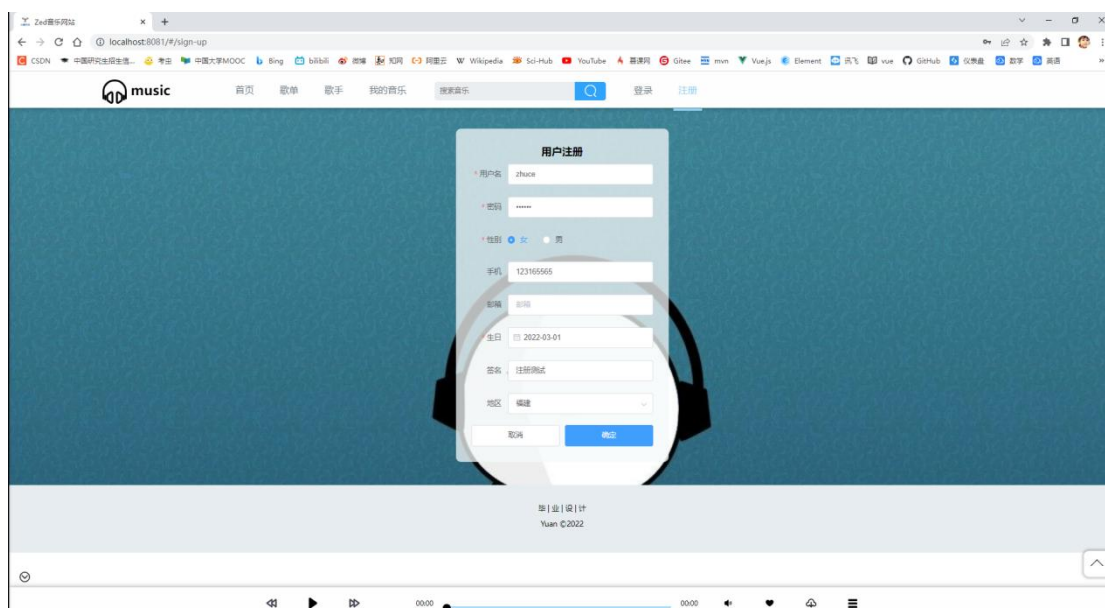
6.2 测试用例

表：6-1 测试用例表

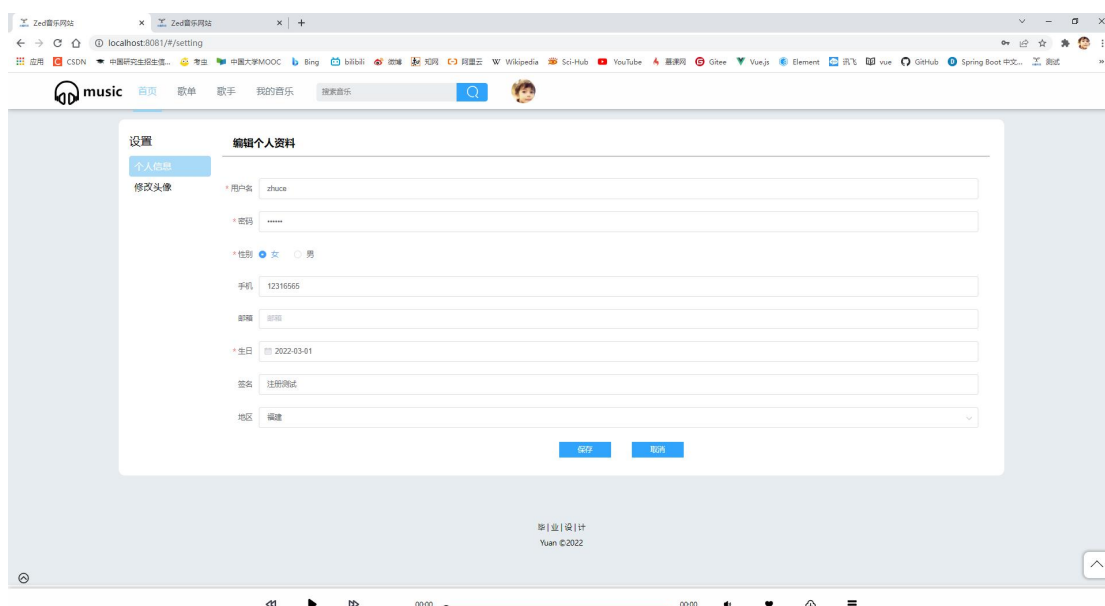
用例编号	测试功能模块	执行步骤	预期结果	实际结果
1	验证用户注册功能	浏览器打开前台网站，点击注册按钮，输入相关信息点击确定。	注册成功并且能通过注册的账号密码登录。	注册成功并且能通过注册的账号密码登录。
2	验证评论功能	登录后点击歌单，在页面底部发表评论并提交。	评论成功并显示评论。	评论成功并显示评论。
3	验证管理员修改用户信息功能	登录管理员后台系统，点击用户管理，点击编辑进行修改。	修改成功并显示修改信息。	修改成功并显示修改信息。
4	验证管理员修改歌手信息功能	登录管理员后台系统，点击歌手管理，点击编辑进行修改。	修改成功并显示修改信息。	修改成功并显示修改信息。
5	验证添加歌单功能	登录管理员后台系统，点击歌单管理，点击添加歌单进行添加。	系统页面上弹窗提示添加成功，并会在页面上更新添加的信息。	系统页面上弹窗提示添加成功，并会在页面上更新添加的信息。

6.3 测试结果截图

(1) 验证用户注册功能，如下图：6-1 和图：6-2。浏览器打开前台网站，点击注册按钮，输入相关信息点击确定，页面弹窗提示注册成功。图：6-2 查看刚注册的用户信息与填写信息无误。

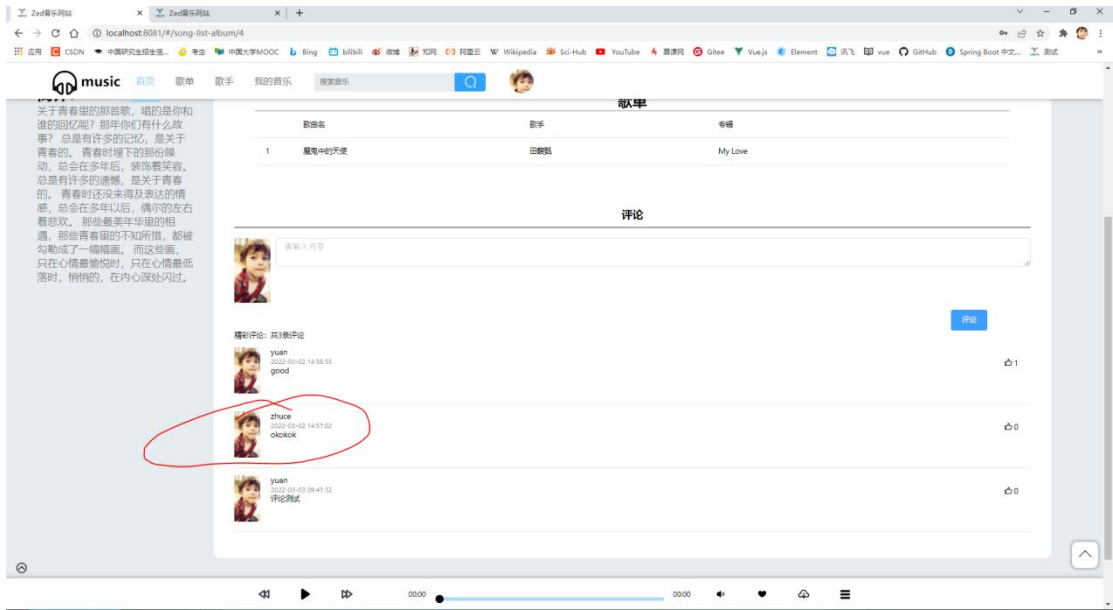


图：6-1 用户注册



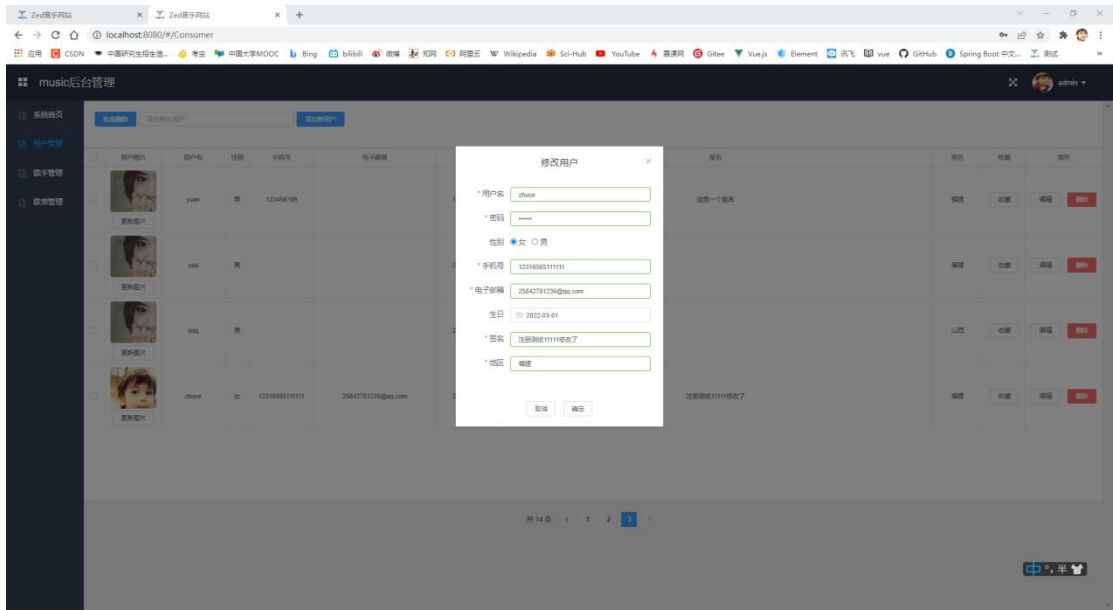
图：6-2 注册成功并登录

(2) 验证评论功能，如下图：6-3。登录后点击歌单，在页面底部发表评论并提交。提交之后看到评论区多了一条评论，即评论成功。

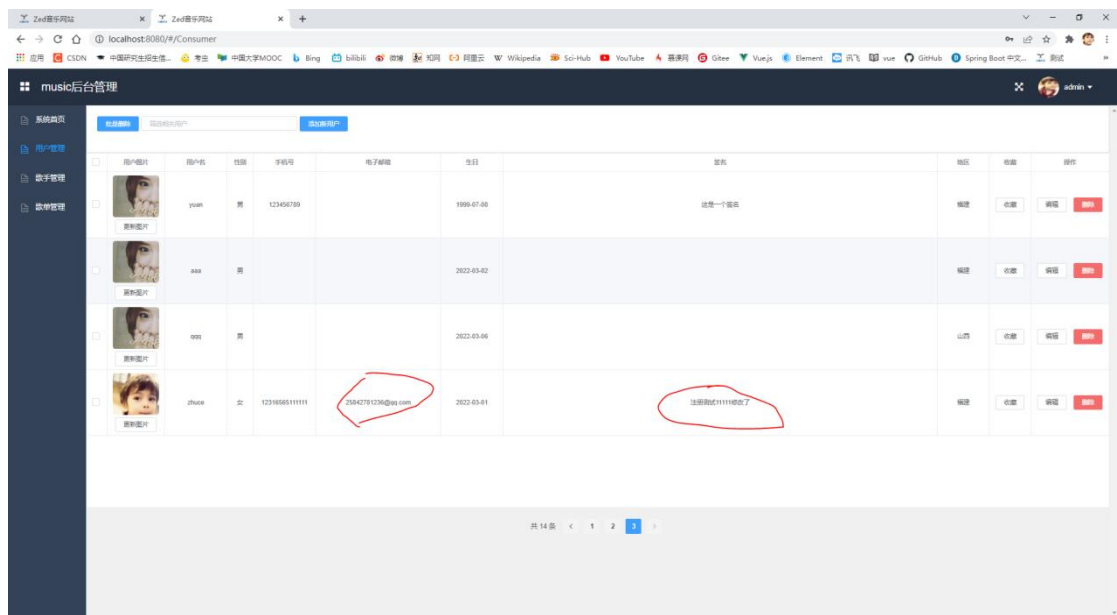


图：6-3 评论功能验证

(3) 验证管理员修改用户信息功能，如下图：6-4 和图：6-5。登录管理员后台系统，点击用户管理，点击编辑进行修改，完成相关信息的修改填写后点击确定，页面弹窗提示修改完成，并在页面中显示修改后的信息。图：6-5

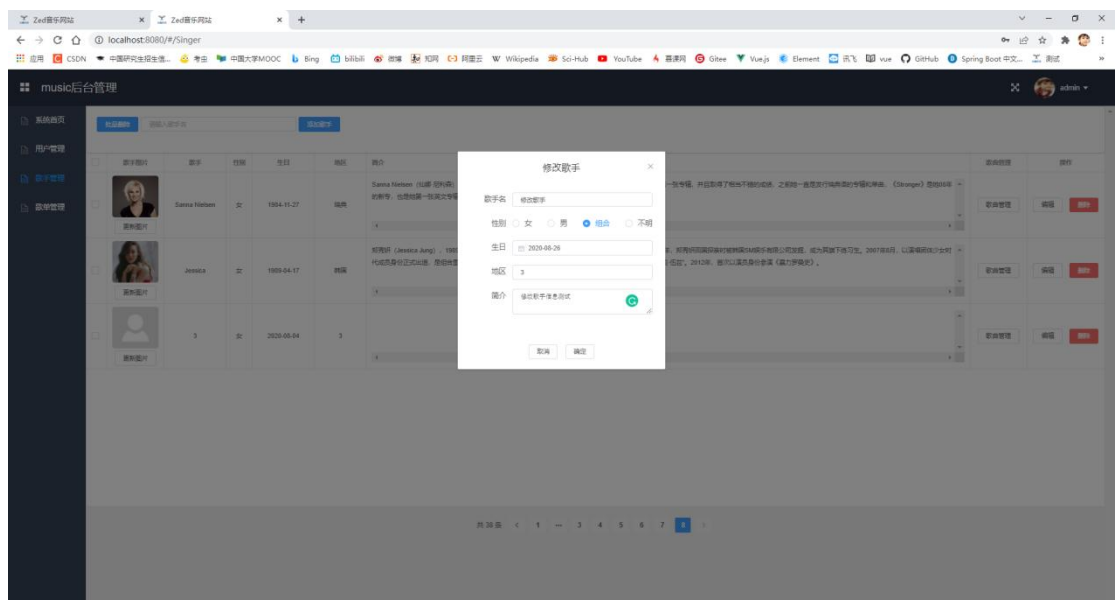


图：6-4 修改用户信息

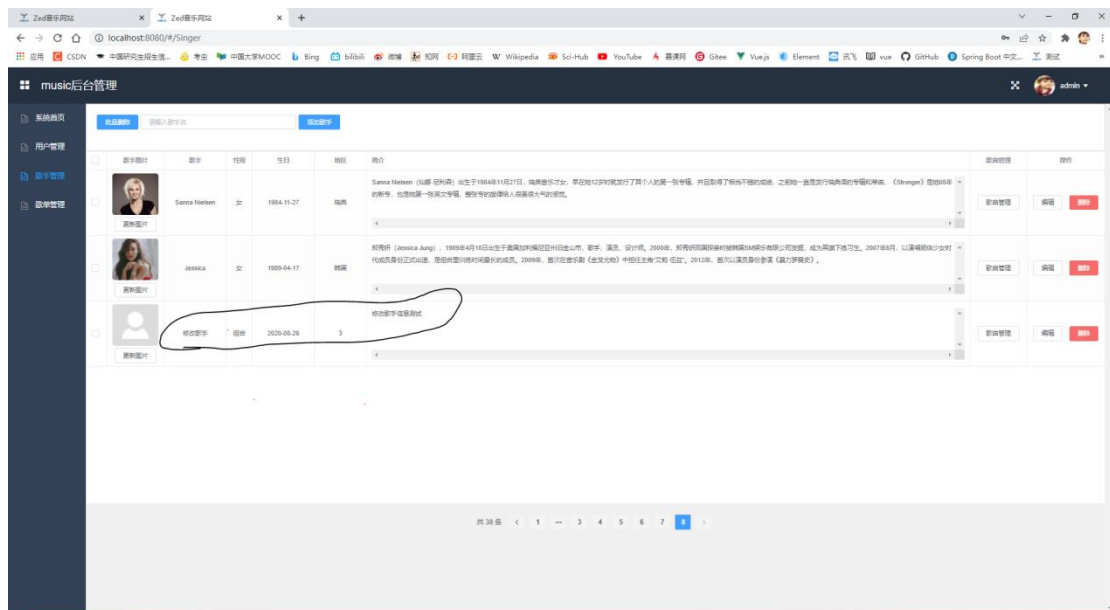


图：6-5 修改用户信息成功

(4) 验证管理员修改歌手信息功能，如下图：6-6 和图：6-7。登录管理员后台系统，点击歌手管理，点击编辑进行修改。填写相关信息完成后点击确定，页面弹窗修改成功，并在页面显示修改后的信息。

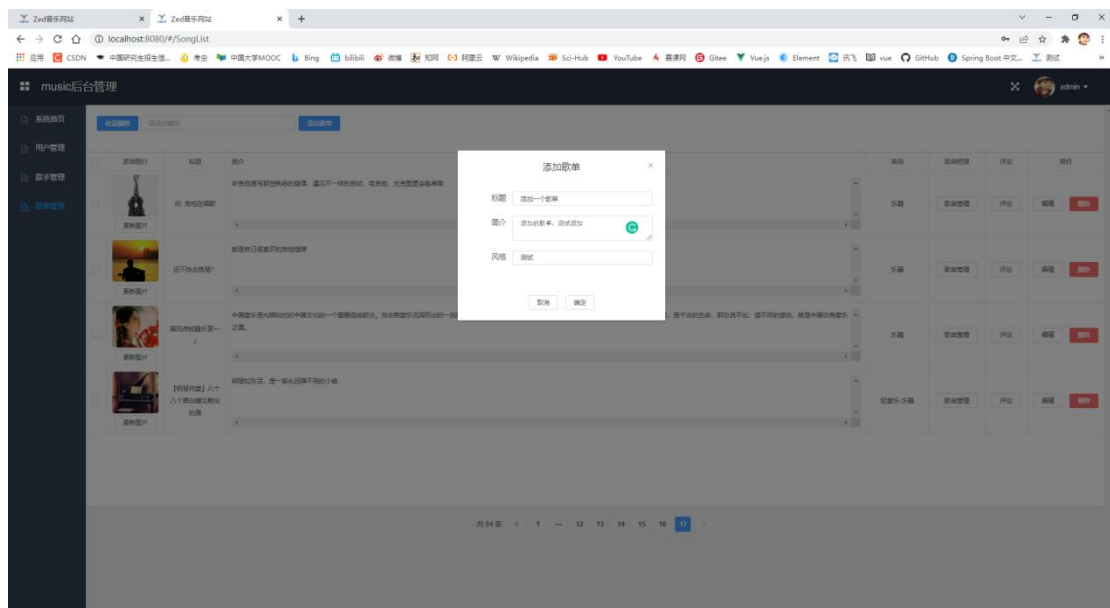


图：6-6 修改歌手信息

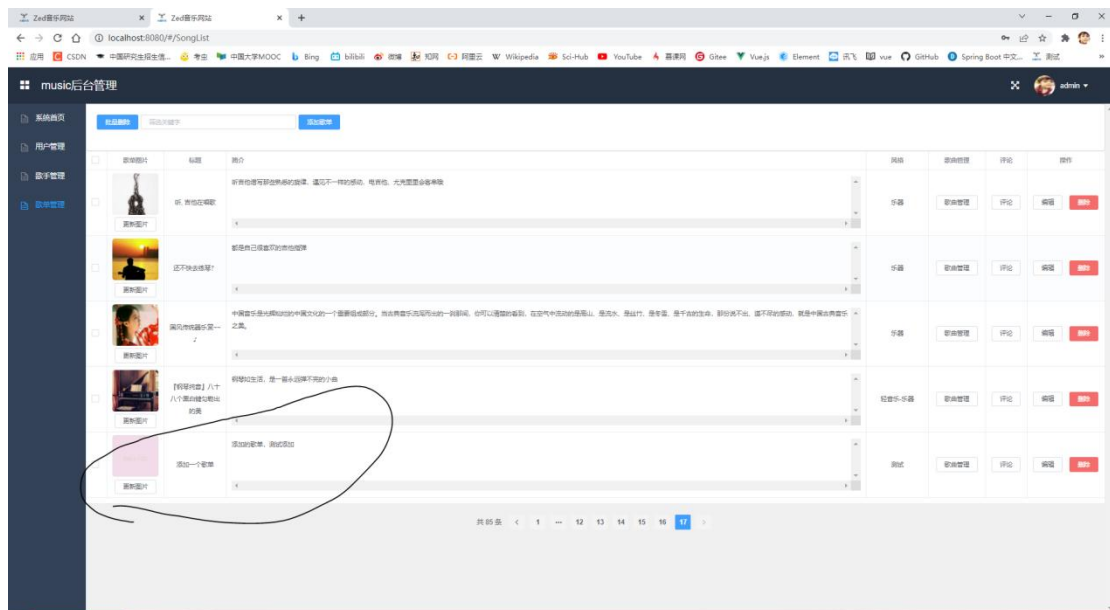


图：6-7 修改歌手信息成功

(5) 验证添加歌单功能，如下图：6-8 和图：6-9。登录管理员后台系统，点击歌单管理，点击添加歌单进行添加。



图：6-8 添加歌单



图：6-9 添加歌单成功

7. 总结

历时三个月，在导师的指导和我的共同努力下，完成了毕业设计。在走完了整个系统设计与实现的全过程后，深感自己的不足与欠缺不仅仅是专业技术上的代码能力的欠缺，还有在业务与产业上的认识与分析能力的不足，在资料收集和市场调查的能力还有待提升。因为我从始至终，亲自完成了资料收集整理、文献阅读、需求分析、系统设计、系统实现与测试的全过程，这对我自己来说，本身就是一个重大挑战。通过这三个月的学习与努力，我虽面临着种种困难，但都一一克服，我自己的能力也在逐步提高，特别是解决问题的思维和方法，让我在整个过程中不断的积累这方面的经验。虽然困难种种，但我还是完成了音乐网站系统的实现和全部功能，并且根据系统的相关资料完成了毕业论文的撰写。在专业能力上，我已实现了新的专业技术的突破；在业务分析方面，考虑和解决问题的思维也逐步成熟；在学业上，完成毕业设计及论文意味着我已经成为一名优秀的本科毕业生了；在思想上，经过查阅的资料也深感幸福生活来之不易，国家的逐步强大、社会逐步进步，我们青年的历史使命也要承担起来；在生活上，完成毕设期间，因任务重多，每天都是规律的作息，保持适当运动和良好的身体状况，已经养成了良好的生活习惯……在整个音乐网站系统的设计与实现和毕业论文的完成中，我阅读了大量的文章，他们给我介绍了很多新的互联网技术，例如 SpringBoot 框架、Vue 框架+ElementUI、V-charts 等，还有 MVVM 架构、前后端分离的设计模式、IntelliJ IDEA 开发工具等，在尝试新的技术的同时，也对开发更加感兴趣，对 java 开发更为向往。

参考文献

- [1] 马艳夕. 基于 SpringBoot 与 Vue 技术的企业电商平台的设计与实现[J]. 信息与电脑, 2021, 33(3):99-100.
- [2] 刘小英, 刘强. 在线学习交流平台的设计与实现[J]. 攀枝花学院学报: 综合版, 2021, 38(2):62-67.
- [3] 张雷, 王悦. 基于 SpringBoot 微服务架构下的 MVC 模型研究[J]. 安徽电子信息职业技术学院学报, 2018, 17(4):1-9.
- [4] 张剑飞. JavaEE 开发技术[M]. 哈尔滨工业大学出版社. 2013.
- [5] 王映龙. JavaEE 实用教程[M]. 清华大学出版社. 2011.
- [6] 汪云飞. JavaEE 开发的颠覆者[M]. 电子工业出版社. 2016.
- [7] MetskerSJ. Java 框架设计. 第一版[M]. 电子工业出版社. 2012.
- [8] 明日科技. Java 从入门到精通. 第五版[M]. 清华大学出版社. 2019.
- [9] 刘京华. JavaWeb 整合开发王者归来[M]. 清华大学出版社, 2010.
- [10] 梁灏. Vue.js 实战. 第一版[M]. 清华大学出版社. 2017.
- [11] 石正喜. MySQL 数据库实用教程[M]. 北京师范大学出版社. 2014.
- [12] 孙鑫. JavaWeb 开发详解[M]. 电子工业出版社. 2010.
- [13] 吕宇琛. SpringBoot 框架在 web 应用开发中的探讨[J]. 科技创新导报, 2018, 15(8):168-168.
- [14] 贾黎, 刘忠超, 李英玲. 基于 SpringBoot+Vue 新冠疫苗接种全过程管理系统[J]. 西南民族大学学报(自然科学版). 2022, 48(02):181-189.
- [15] 李唯. 基于 SpringBoot+Mybatis 的驾校预约系统设计与实现[J]. 电脑编程技巧与维护. 2022, (03):10-12.
- [16] 吴瑞. 服务器虚拟化系统 Web 管理平台的设计与实现[D]. 山东大学. 2020.
- [17] The_Beacon. 前端小白的第一个 Vue 程序[EB/OL]. https://blog.51cto.com/u_15175871/723303, (2021-04-21) [2022-4-26].
- [18] 魏明俊, 杨庆. 基于 SpringBoot 的评价预警系统设计与实现[J]. 电脑编程技巧与维护. 2022, (02):12-14.
- [19] 向福川, 方玉, 刘浪等. 基于 SpringBoot+Vue 框架的协同育才系统设计与开发[J]. 现代信息科技. 2021, 5(14). 5-7+12.
- [20] 何豪. 基于 Vue.js 的国家语委申报管理系统的设计与实现[D]. 武汉: 华中师范大学. 2021.
- [21] 李凯, 李士成, 何子旋等. 基于安卓系统手机应用的研究[J]. 新商务周刊. 2016-1-1.