

杭州电子科技大学信息工程学院

课程设计报告

课 程	HTML5 移动开发
-----	------------

系	计算机系
---	------

专 业	计算机
-----	-----

班 级	180927101
-----	-----------

学 号	189050221
-----	-----------

学生姓名	林元澍
------	-----

任课教师	王欣
------	----

学 期	2021-2022 学年第一学期
-----	------------------

目 录

- 1 网站需求分析.....1
 - 1.1 用户需求分析.....1
 - 1.2 功能需求分析.....1
- 2 网站设计.....3
 - 2.1 网站总体设计.....3
 - 2.2.1 网站模块设计.....3
 - 2.2.2 网站流程设计.....3
- 3 网站实现.....7
 - 3.1 首页的实现.....7
 - 3.2 歌手详情页面的实现.....7
 - 3.3 个人信息页面的实现.....7
 - 3.4 歌词详情页面的实现.....7
 - 3.5 歌曲详情及评论页面的实现.....7
 - 3.6 个人收藏页面的实现.....7
 - 3.7 全部歌单页面的实现.....7
 - 3.8 全部歌手页面的实现.....7
- 4 结 论.....21

1 网站需求分析

1.1 用户需求分析

这个网站用于给好友、网友分享互联网流行的、受欢迎的歌曲。用户可以在这里进行注册登陆，编辑自己的信息，作为一个具有个性的用户在网站上冲浪。

为了跟上网络时代的热潮，这个在线音乐网站，这个应用程序就是受网易云音乐在线音乐网站(music.163.com)启发。该音乐网站可以满足网站所有注册用户的听歌、评论交流的基本需求。

1.2 功能需求分析

作为一个在线的、音频播放网站，本网站应该具有如下功能：

- 1.最基本的歌曲选择播放、暂停、上一首、下一首、播放列表的组件。
- 2.浏览歌曲的歌词、歌手的名字、歌曲所属的专辑。

为了充分发挥出网络的互联互动功能，还需要进一步具有如下功能：

- 1.注册、登陆功能。
- 2.首页提供歌曲推荐，显示最近的流行歌曲。
- 3.歌手及歌手分类页面。
- 4.查看修改我的收藏歌曲。
- 5.歌曲评价页面，发表评论、观看用户对歌曲的评论。
- 6.歌曲的歌词显示页面。
- 7.歌单推荐页面，显示用户的精选收藏歌单。

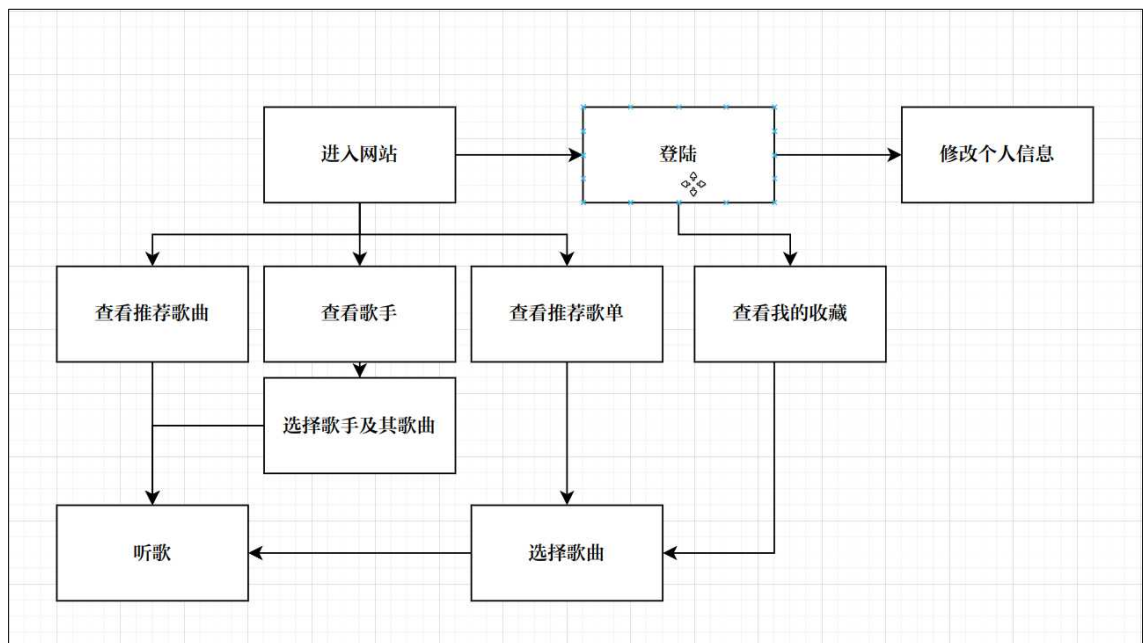
2 网站设计

2.1 网站总体设计

2.2.1 功能模块设计

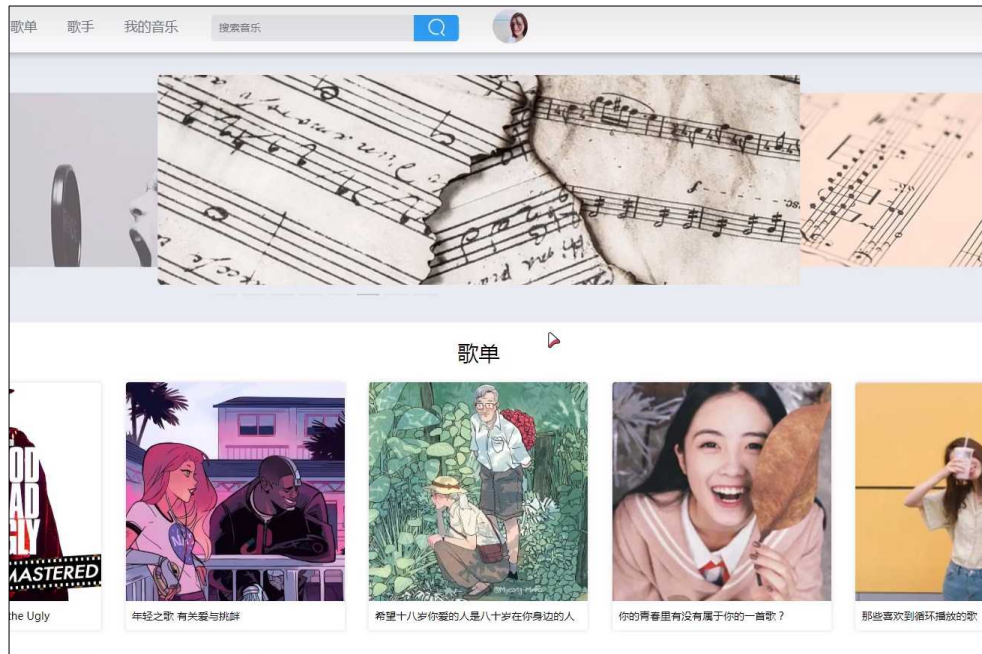
- 1.头像上传模块。
- 2.播放模块。
- 3.首页推荐滑动模块。
- 4.个人详情 hover 模块。
- 5.评论框模块。
- 6.（歌曲）专辑信息模块。

2.2.2 网站流程设计



3 网站实现（部分代码截图和页面截图）

3.1 首页的实现



首页

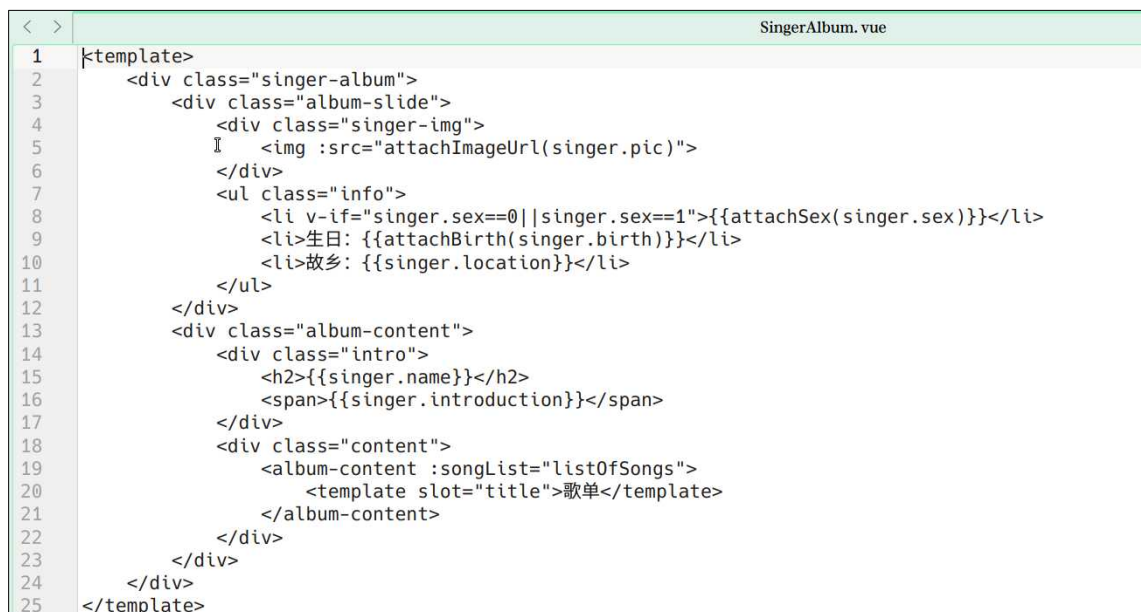
```
1 <template>|
2   <div class="home">
3     <swiper />
4     <div class="section" v-for="(item,index) in songsList" :key="index">
5       <div class="section-title">{{item.name}}</div>
6       <content-list :contentList="item.list"></content-list>
7     </div>
8   </div>
9 </template>
10 <script>
11 import Swiper from "../components/Swiper";
12 import contentList from '../components/ContentList';
13 import {getAllSinger,getAllSongList} from '../api/index';
14 export default {
15   name: 'home',
16   components: {
17     Swiper,
18     contentList
19   },
20   data () {
21     return {
22       songsList: [
23         {name:"歌单",list: []},
24         {name:"歌手",list: []}
25       ]
26     }
27   },
28   created () {
29     this.getSongList();
30     this.getSinger();
31   },
32   methods: {
33     getSongList(){ //获取前十条歌单
34       getAllSongList().then((res) => {
35         this.songsList[0].list = res.slice(0,10);
36       }).catch((err) => {
37         console.log(err);
38       })
39     }
40   }
41 }
```

首页代码展示

3.2 歌手详情页面的实现

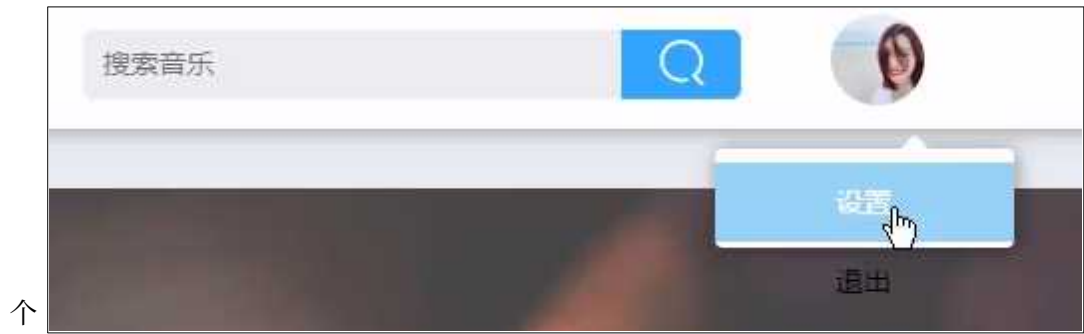


歌手详情页面



歌手详情页代码

3.3 个人信息页面的实现



个人资料实现 1

设置

个人信息

修改头像

编辑个人资料

用户名 wang

密码 ***

性别 ☒ 女 ☐ 男

手机 13776412237

邮箱 yoonn@qq.com

生日 2019-05-24

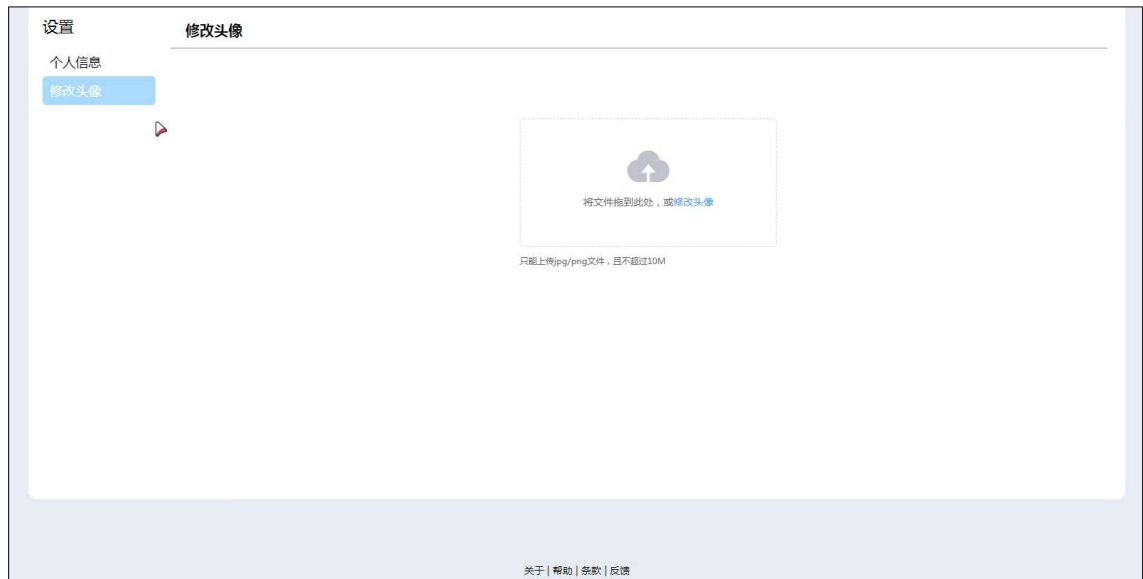
签名 好好吃饭

地区 山西

保存 取消

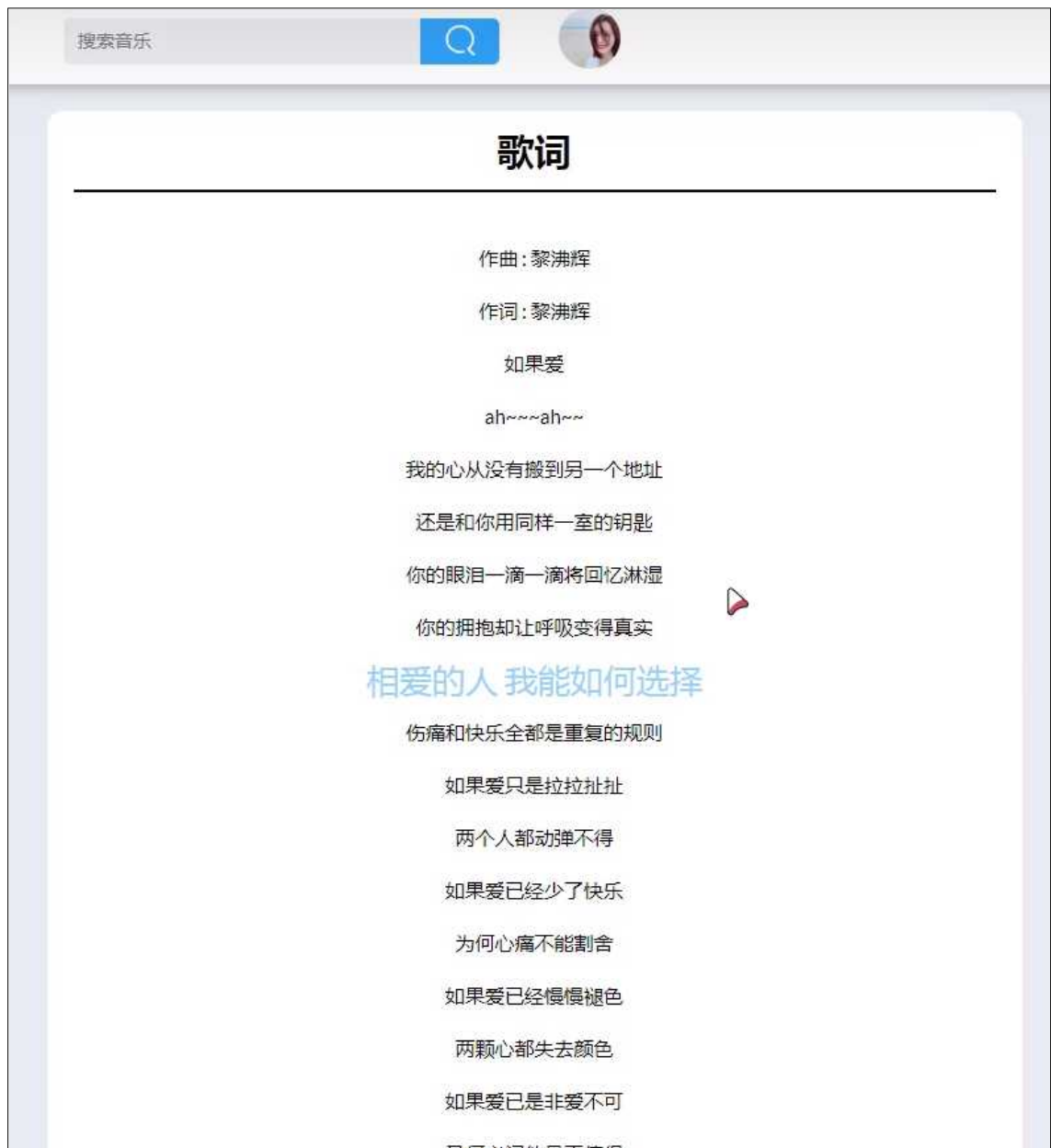
关于 | 帮助 | 条款 | 反馈

个人资料实现 2



个人资料实现 3

3.4 歌词详情面的实现



个歌词实现

```
1 <template>
2   <div class="song-lyric">
3     <h1 class="lyric-title">歌词</h1>
4     <!-- 有歌词 -->
5     <ul class="has-lyric" v-if="lyr.length" key="index">
6       <li v-for="(item,index) in lyr" v-bind:key="index">
7         {{item[1]}}
8       </li>
9     </ul>
10    <!-- 没有歌词 -->
11    <div v-else class="no-lyric" key="no-lyric">
12      <span>暂无歌词</span>
13    </div>
14  </div>
15 </template>
```

歌词展示部分代码

```
<script>
import {mixin} from '../mixins';
import {mapGetters} from 'vuex';

export default {
  name: 'lyric',
  mixins: [mixin],
  data(){
    return {
      lyr: [] //当前歌曲的歌词
    }
  },
}
```

歌词展示部分代码

3.5 歌曲详情及评论页面的实现



希望十八岁你爱的人是八十岁在你身边的人

歌单评分：**5** 评价：☆☆☆☆☆

简介：
让你怦然心动

歌单

歌曲名	艺人	专辑
1 夜的第七章	周杰伦&范儿	依然范特西
2 龙卷风	G.E.M.邓紫棋	龙卷风

评论



评论

歌曲展示

```

export default {
  name: 'singer-album',
  mixins: [mixin],
  components:{
    AlbumContent
  },
  data(){
    return {
      singerId: '',           //前面传来的歌手id
      singer: {},             //当前歌手信息
    }
  },
  computed:{
    ...mapGetters([
      'listOfSongs',          //当前播放列表
      'tempList',             //当前歌手对象
      'loginIn',              //用户是否已登录
      'userId',               //当前登录用户id
    ])
  },

```

歌曲展示部分代码

```

1 <template>
2   <div class="singer-album">
3     <div class="album-slide">
4       <div class="singer-img">
5         
6       </div>
7       <ul class="info">
8         <li v-if="singer.sex==0||singer.sex==1">{{attachSex(singer.sex)}}</li>
9         <li>生日: {{attachBirth(singer.birth)}}</li>
10        <li>故乡: {{singer.location}}</li>
11      </ul>
12    </div>
13    <div class="album-content">
14      <div class="intro">
15        <h2>{{singer.name}}</h2>
16        <span>{{singer.introduction}}</span>
17      </div>
18      <div class="content">
19        <album-content :songList="listOfSongs">
20          <template slot="title">歌单</template>
21        </album-content>
22      </div>
23    </div>
24  </div>
25 </template>

```

歌曲展示部分代码

3.6 个人收藏页面的实现



我的收藏展示

```
1 <template>
2   <div class="my-music">
3     <div class="album-slide">
4       <div class="album-img">
5         
6       </div>
7       <ul class="album-info">
8         <li>昵称:{{username}}</li>
9         <li>性别:{{userSex}}</li>
10        <li>生日: {{birth}}</li>
11        <li>故乡: {{location}}</li>
12      </ul>
13    </div>
14    <div class="album-content">
15      <div class="album-title">
16        个性签名: {{introduction}}
17      </div>
18      <div class="songs-body">
19        <album-content :songList="collectList">
20          <template slot="title">我的收藏</template>
21        </album-content>
22      </div>
23    </div>
24  </div>
25 </template>
```

我的收藏部分代码

3.7 全部歌单页面的实现



全部歌单展示

```

1 <template>
2   <div class="song-list">
3     <ul class="song-list-header">
4       <li v-for="(item,index) in songStyle" :key="index" @click="handleChangeView(item.name)"
5         :class="{active:item.name==activeName}">
6         {{item.name}}
7       </li>
8     </ul>
9     <div>
10      <content-list :contentList="data"></content-list>
11      <div class="pagination">
12        <el-pagination @current-change="handleCurrentChange" background layout="total,prev,pager,next"
13          :current-page="currentPage" :page-size="pageSize" :total="albumDatas.length">
14        </el-pagination>
15      </div>
16    </div>
17  </div>
18 </template>
19 <script>
20 import ContentList from '../components/ContentList';
21 import {getAllSongList,getSongListOfLikeStyle} from '../api/index';
22 import {mixin} from './mixins';
23 import {songStyle} from '../assets/data/songList';
24

```

全部歌单部分代码

3.8 全部歌手页面的实现



全部歌手展示

```
1 <template>
2   <div class="singer">
3     <ul class="singer-header">
4       <li v-for="(item,index) in singerStyle" :key="index" @click="handleChangeView(item)"
5         :class="{active:item.name==activeName}">
6         {{item.name}}
7       </li>
8     </ul>
9     <div>
10      <content-list :contentList="data"></content-list>
11      <div class="pagination">
12        <el-pagination @current-change="handleCurrentChange" background layout="total,prev,pager,next"
13          :current-page="currentPage" :page-size="pageSize" :total="albumDatas.length">
14        </el-pagination>
15      </div>
16    </div>
17  </div>
18 </template>
```

全部歌手部分代码展示

```
1 <template>
2   <div class="singer">
3     <ul class="singer-header">
4       <li v-for="(item,index) in singerStyle" :key="index" @click="handleChangeView(item)"
5         :class="{active:item.name==activeName}">
6         {{item.name}}
7       </li>
8     </ul>
9     <div>
10      <content-list :contentList="data"></content-list>
11      <div class="pagination">
12        <el-pagination @current-change="handleCurrentChange" background layout="total,prev,pager,next"
13          :current-page="currentPage" :page-size="pageSize" :total="albumDatas.length">
14        </el-pagination>
15      </div>
16    </div>
17  </div>
18 </template>
```

全部歌手部分代码展示

4 结 论

网站开发入门到熟练需要经历大量的学习与实践，特别是新兴的网页框架如 vue-x+elementUI 不断应用于程序的实际开发中。对与纯原生 HTML 的学习不应该停止不前，同时，对于新兴的网页框架 vue 学习使我对网页设计有了全新的认识。

这次简短的大作业编码，我基本认识了 vue 的资源文件的框架布局。初识 vue-x 的全局变量。理解网页的页面和组件的基本关系，对比原生 html 具有更轻的代码容量、更好的移植兼容性。学会使用 npm 并能快速初始化项目安装依赖包。