

同濟大學

TONGJI UNIVERSITY

实验名称

姓 名

学院（系）

专 业

任课教师

日 期

一、实验原理

本次实验的内容为服务端的编程，是基于实验三的基础上完成的。下面将详细说明前后端的实验原理。

1. 前端

前端的开发我采用了 Vue.js 框架进行开发。与传统的 Javascript 开发方法不同，Vue.js 是一种数据驱动的开发框架。其主要优势是可以根据数据的变化进行实时渲染，并将视图各部分切分为功能组件分层实现。Vue.js 在前端框架中也较为轻量级，所需的学习成本较低。

2. 后端

后端部分我使用了 Express 进行开发，负责接收前端发来的请求并及时响应，同时对请求进行逻辑处理，将其转化为具体的数据库指令，相当于前端与数据库之间的中介。Express 和 VUE 都是基于 node.js 完成的，简单的说 Node.js 就是运行在服务端的 JavaScript，是一个事件驱动 I/O 服务端 JavaScript 环境，基于 Google 的 V8 引擎，V8 引擎执行 Javascript 的速度非常快，性能非常好

3. 数据库

数据库部分我使用的是 MySQL。MYSQL 是一款应用广泛的关系型数据库管理系统，对 SQL 标准支持较好，并且具有很强的通用性。还采用了 InnoDB 存储引擎，可以高效率地对事务进行执行。这一部分负责建立数据库和 table，并且我预存了一些数据进去，然后根据后端的请求对数据进行操作。

二、实验步骤

1. 框架搭建

Vue 框架的搭建非常的简单好操作，主要步骤如下：

(1) 下载安装 Node.js 之后，打开 cmd 输入：

Npm install vue-cli -g : 下载 vue-cli 脚手架

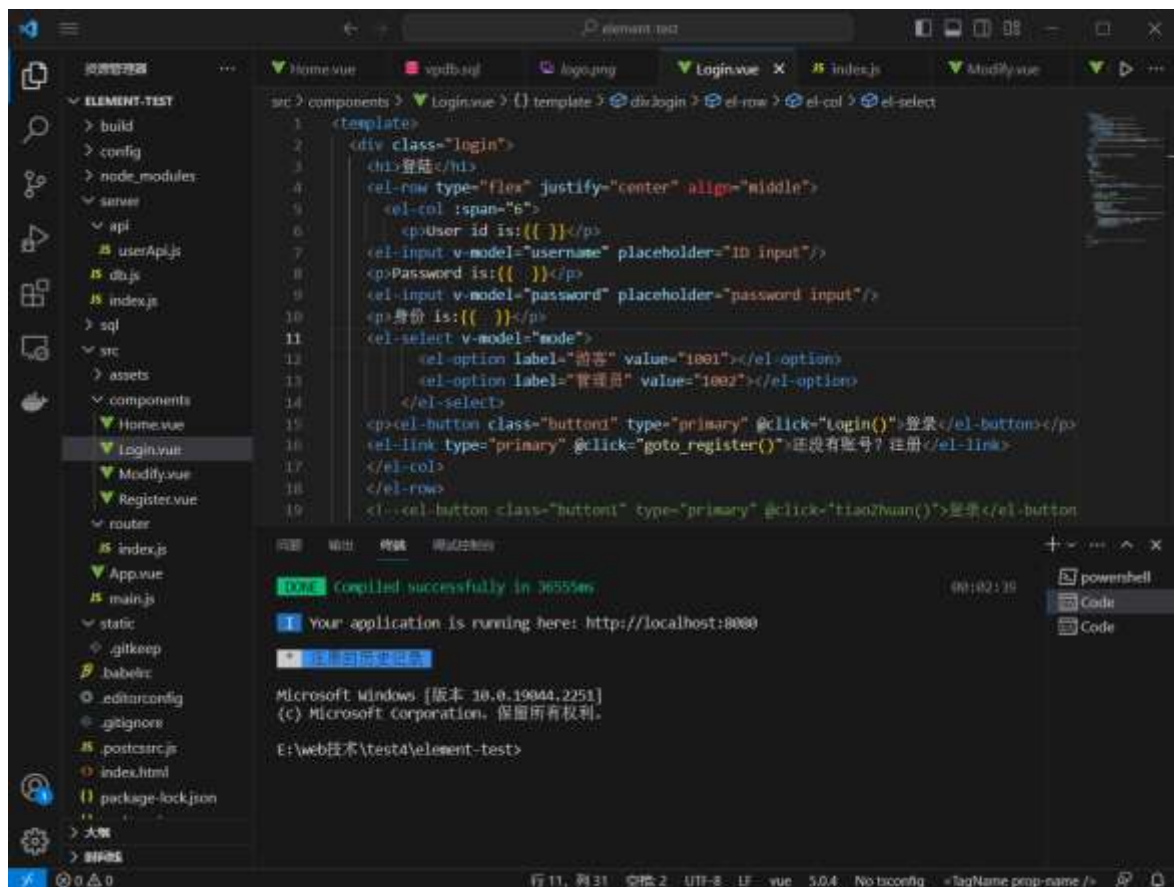
Vue init webpack xxx : 创建项目

Npm install vue-resource: 安装 vue-resource 依赖

Npm install express mysql body-parser: 安装模块和依赖

(2) 打开创建好的项目后，在项目根目录文件下新建 server 文件夹，在文件夹中新建配置后端服务器的文件和数据库连接配置的文件，并新建 api 文件夹，存储接口文件。

搭建好的框架如图所示：



2. 前端编程

前端编程的步骤很简单，首先先写<template>部分，这一部分和html基本没有差别。然后<script>部分和普通的JavaScript也没什么区别，但是方便了很多，因为很多变量它可以实时渲染，所以就不用写的很复杂。

本次实验中一共写了五个vue文件，App.vue里面没有实际的内容，Login.vue为登录页面，Register.vue为注册页面，Home.vue为主页，Modify.vue为管理员添加video到列表的页面。

前端的具体展示到下一部分介绍。下面简述一下前端一些细节的设计。

页面的跳转使用Router实现，需要首先在router/index.js里面声明这些页面，然后我采用的跳转的方式是直接this.\$router.push()的方法，非常便捷，但是我也觉得可能存在一些安全隐患。

以及在本次项目的设计中，要区分不同用户，不同用户有不同的操作权限。我采用的方式为直接在建表的时候就在user用户下加一个mode属性，为1001是为游客，为1002是为管理员。这样我在home页面的设计时，就无需设计两个页面，直接统一用一个页面，然后用v-if来判断是否放置一些管理员专属的按钮功能。

3. 后端编程

在完成了前端之后，可以看出前端都需要什么接口，然后在api文件夹下面直接写对应的接

口函数，对数据进行处理之后，就可以转化为数据库指令，将指令传给数据库，并且获取数据库传回的内容，再加工好返回给前端。

我一共设计了五个接口：

(1) 增加用户接口：

用于注册时，添加用户信息到数据库中的 user 表中。

```
// 增加用户接口
router.post('/addUser', (req, res) => {
  var params = req.body;
  var sql = 'insert into user(username, mode,password) values (?, ?, ?)';
  console.log(params);
  conn.query(sql, [params.username, params.mode,params.password], function(err,
result) {
    if (err) {
      console.log(err);
    }
    if (result) {
      jsonWrite(res, result);
    }
  })
});
```

(2) 查询用户接口：

用于登陆时，在 user 表中查看是否有相应的用户，登录密码是否正确。

```
// 登录判断
router.post('/queryUser', (req, res) => {
  var params = req.body;
  var sql = " select * from user where username = ? and mode = ? and password
= ?";
  console.log(params);
  conn.query(sql, [params.username,params.mode,params.password],
function(err, result) {
    if (err) {
      console.log(err);
    }
    if (result) {
      console.log(result.length);
      if(result.length === 0 ){
        console.log("不存在该用户");
        jsonWrite(res, result);
      }
      else {
        console.log("存在该用户");
        jsonWrite(res, result);
      }
    }
  })
});
```

```
}
})
});
```

(3) 删除视频接口:

用于管理员在播放列表中删除视频, 在 playlist 表中删除相应的信息。

```
router.post('/deletevideo', (req, res) => {
  var params = req.body;
  var sql = " delete from playlist where videoname = '" + params.videoname +
  "'";
  console.log(params);
  conn.query(sql, [params.videoname], function(err, result) {
    if (err) {
      console.log(err);
    }
    if (result) {
      console.log(res);
      jsonWrite(res, result);
    }
  })
});
```

(4) 获取视频列表的接口:

用于在 home 页面, 从数据库获得视频列表, 从 playlist 表中获取所有的 videoname 信息。

```
router.get('/getvlist',(req,res)=>{
  conn.query('select videoname from playlist',function(err,result){
    if(err)
    {
      console.log(err)
    }
    console.log(typeof row)
    jsonWrite(res,result)
  })
})
```

(5) 添加视频到列表的接口:

用于在 modify 页面, 管理员输入视频名称, 添加视频信息到 playlist。

```
// 增加视频接口
router.post('/addvideo', (req, res) => {
  var params = req.body;
  var sql = 'insert into playlist(VID, videoname) values (?, ?)';
  console.log(params);
  conn.query(sql, [params.VID, params.videoname], function(err, result) {
    if (err) {
      console.log(err);
    }
    if (result) {
```

```
    jsonWrite(res, result);
  }
})
});
```

4. 数据库设计

在此次实验设计中，建立了名为 testvue 的数据库，然后建立了两个表单：

(1) user 表：

名称	Type	说明
username	Varchar(10)	用户名称，注意不要超出 10 个字符
mode	Varchar(5)	管理员为 1002，普通用户为 1001
password	Varchar(10)	密码

为了方便测试，我已经注册了几个账号进去：

管理员账号：username: manager, password: 123

游客账号：username: youke, password: 123

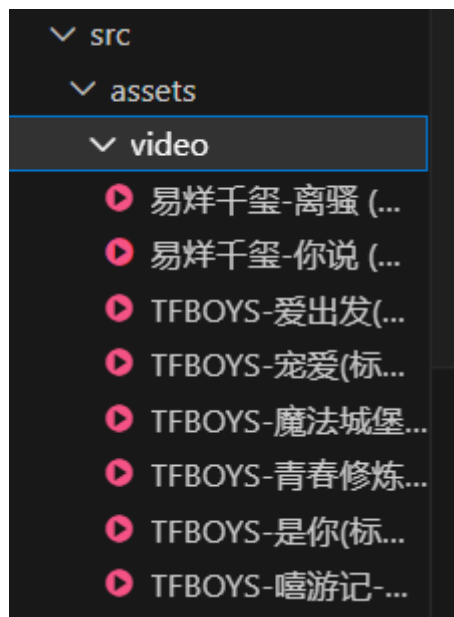
(2) playlist 表：

名称	Type	说明
VID	Int	用户名称，注意不要超出 10 个字符
videoname	Varchar(50)	Video 的名称，例如“易烱千玺-你说（饭制版）（标清）.mp4”

在建表时，我添加了一些 video 进来：

```
insert into playlist(VID,videoname) values(1,'TFBOYS-爱出发(标清).mp4');
insert into playlist(VID,videoname) values(2,"TFBOYS-魔法城堡（电影《洛克王国3》
主题曲）(标清).mp4");
insert into playlist(VID,videoname) values(3,"TFBOYS-宠爱(标清).mp4");
insert into playlist(VID,videoname) values(4,"TFBOYS-青春修炼手册(标清).mp4");
insert into playlist(VID,videoname) values(5,"TFBOYS-嘻游记-剩下的盛夏(标
清).mp4");
```

我一共在 assets/video 文件夹下放了这些视频：



所有视频的名称为（后 5 个已经添加进去了）:

易烊千玺-离骚（《思美人》电视剧片尾曲）(标清).mp4

易烊千玺-你说（饭制版）(标清).mp4

TFBOYS-是你(标清).mp4

TFBOYS-嘻游记-剩下的盛夏(标清).mp4

TFBOYS-宠爱(标清).mp4

TFBOYS-爱出发(标清).mp4

TFBOYS-青春修炼手册(标清).mp4

TFBOYS-魔法城堡（电影《洛克王国 3》主题曲）(标清).mp4

三、实验内容

（1）项目运行步骤：

运行 server 文件下的 index.js，运行后端服务器：

```

问题  输出  终端  调试控制台

Microsoft Windows [版本 10.0.19044.2251]
(c) Microsoft Corporation。保留所有权利。

E:\web技术\test4\element-test>node "e:\web技术\test4\element-test\server\index.js"
success listen at port:3000.....
    
```

运行 vue：

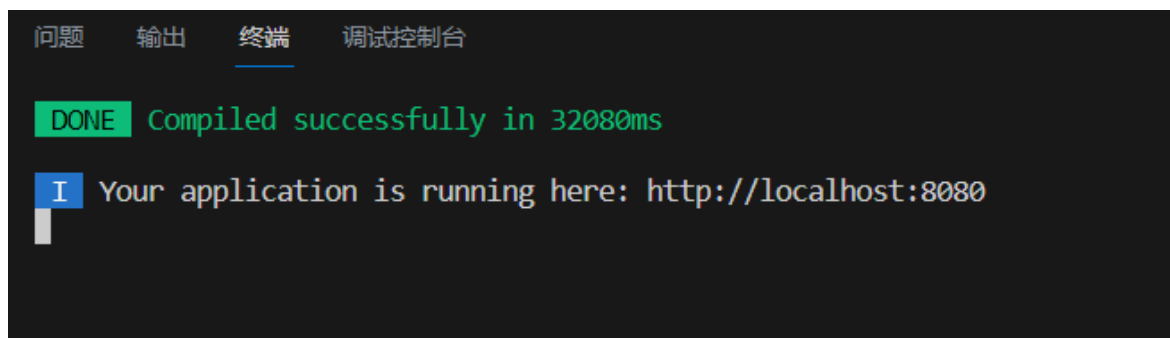
```
Microsoft Windows [版本 10.0.19044.2251]
(c) Microsoft Corporation. 保留所有权利。

E:\web技术\test4\element-test>npm run dev

> video-player@1.0.0 dev
> webpack-dev-server --inline --progress --config build/webpack.dev.conf.js

█
```

这个运行起来有点慢，运行成功后如图：



The screenshot shows a terminal window with tabs for '问题' (Issues), '输出' (Output), '终端' (Terminal), and '调试控制台' (Debug Console). The '终端' tab is active, displaying the following text:

```
DONE Compiled successfully in 32080ms

I Your application is running here: http://localhost:8080
```

访问 <https://localhost:8080> 即可进入。

(2) 项目展示：

本次实验实现的功能有：

- 1、 有用户注册和登录模块，不同的用户有不同的操作权限，分为 2 个角色，一个游客，一个管理员。游客只能播放视频。管理员可以修改播放列表。
- 2、 从后端数据库获取播放列表，进行播放。
- 3、 管理员可编辑列表。
- 4、 视频内容是预先存储好的。
- 5、 所有人都可以下载视频到本地。

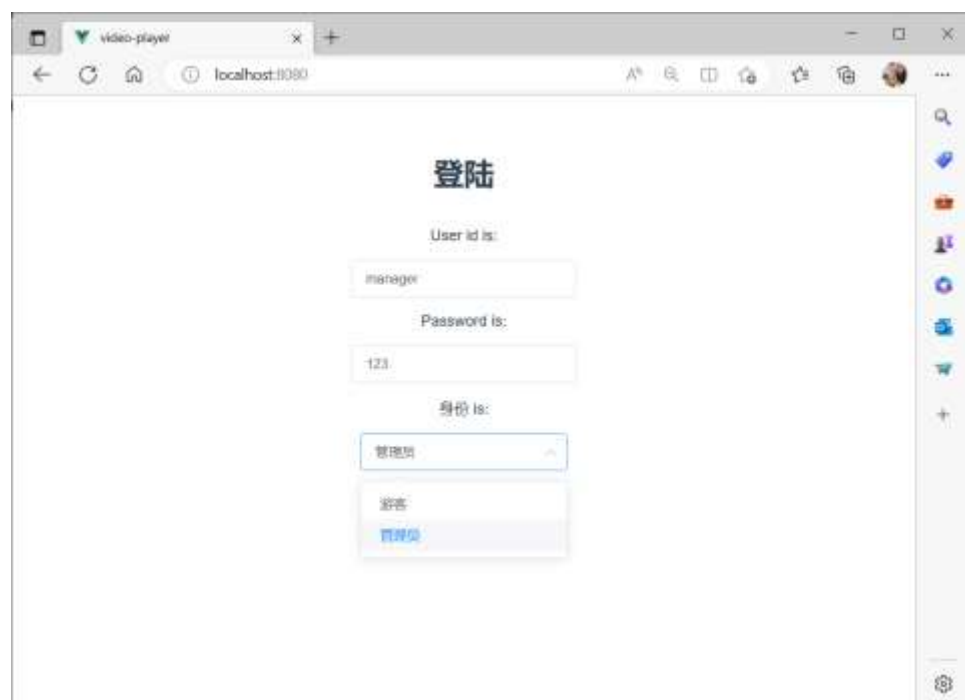
下面是页面设计及功能的详细介绍：

1. 登录页面

访问网址进入后的界面为登录界面：



如果有账号可以直接登录，输入信息，并选择身份：

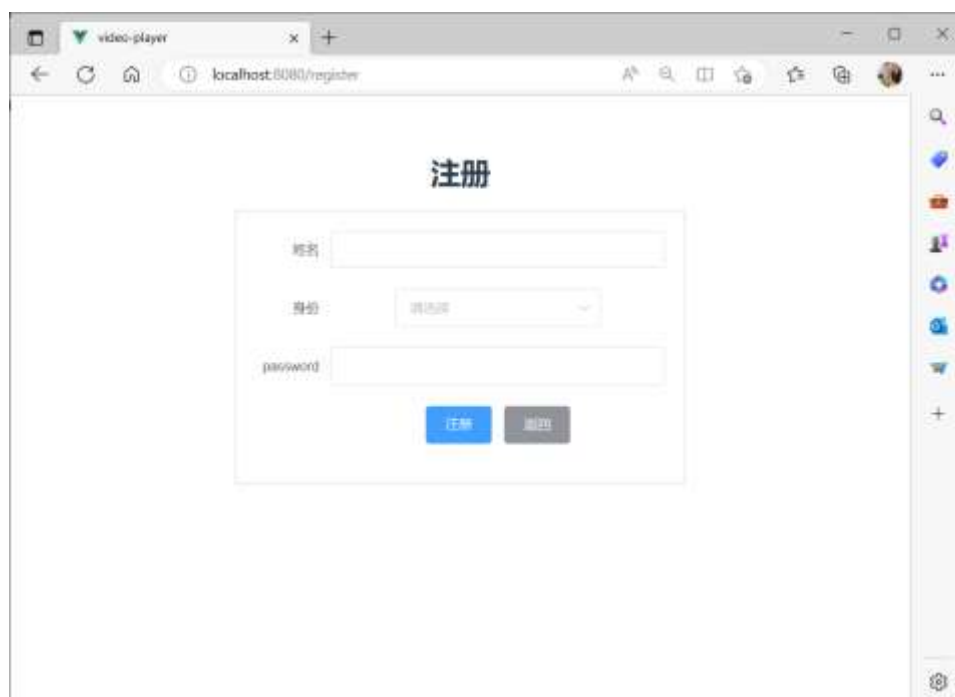


然后点击登录即可跳转到 home 页面。如果输入不正确会有提示信息：



2.注册页面

在登陆页面点击“还没有账号？注册”可进入到注册页面

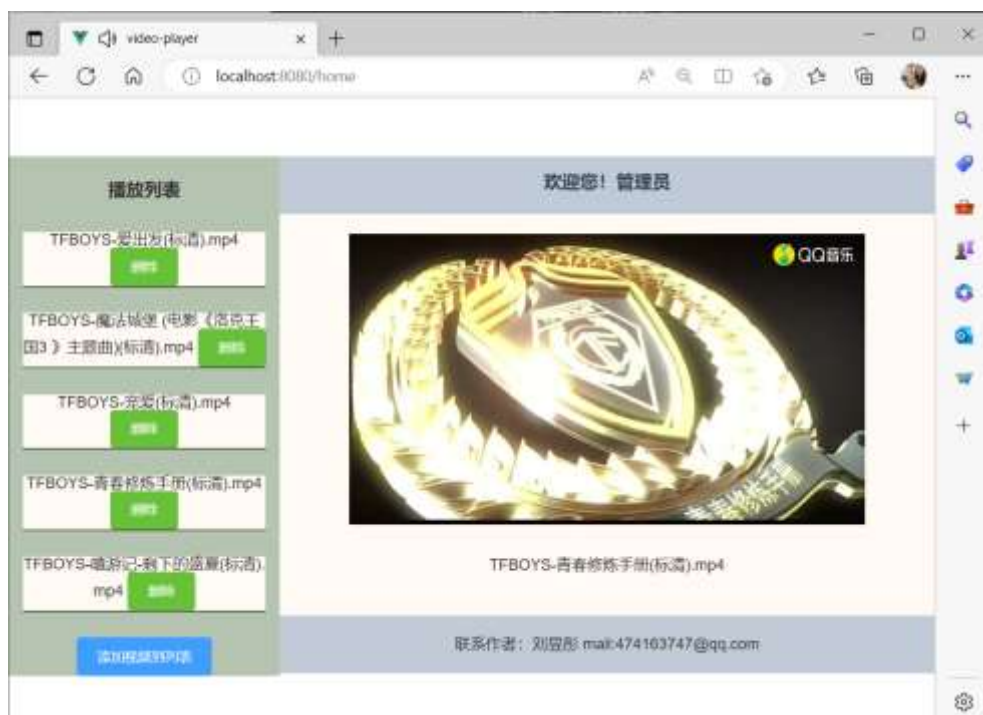


注册成功后直接跳转到主页，点击返回会回到登录页面。错误的信息提示和登陆页面类似。

3.主页页面

登录后进入主页页面，若以管理员方式登录，则左侧的播放列表含有删除按键和添加视频按键，点击删除按键可以删除视频，点击播放列表的歌曲可以切换到相应的视频：





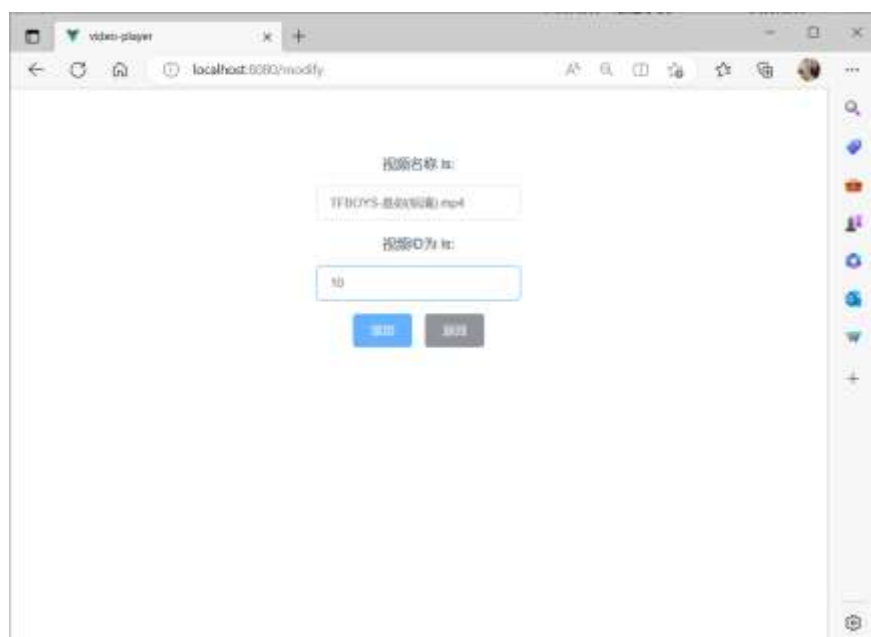
删除视频:



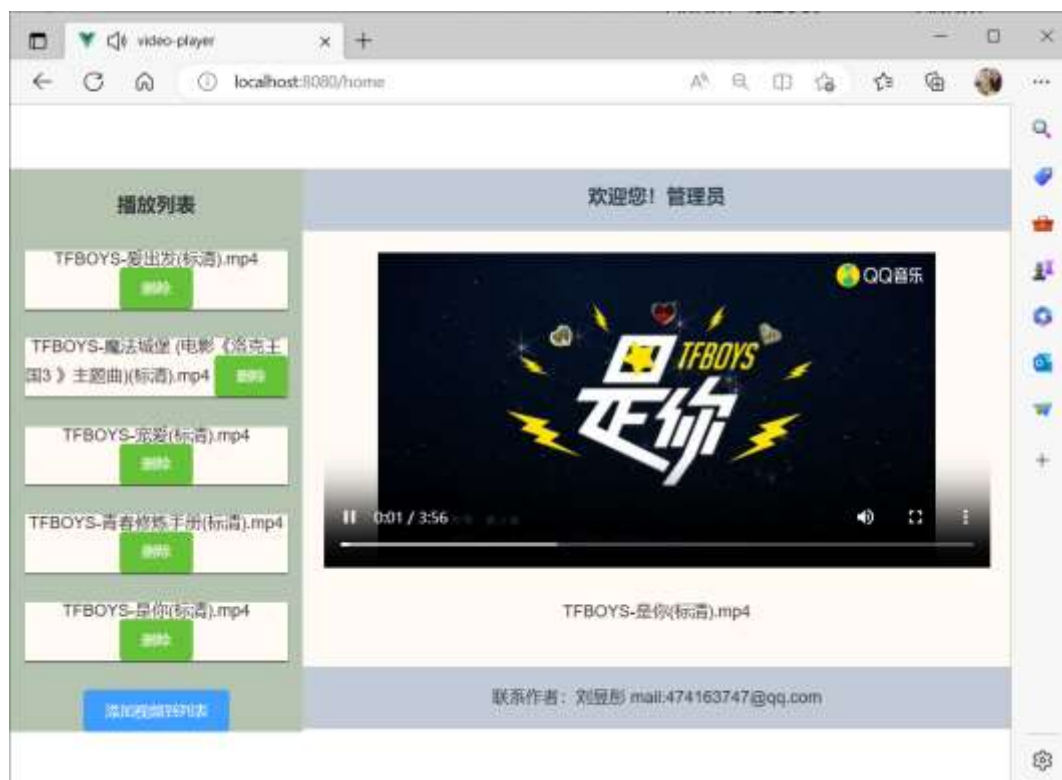
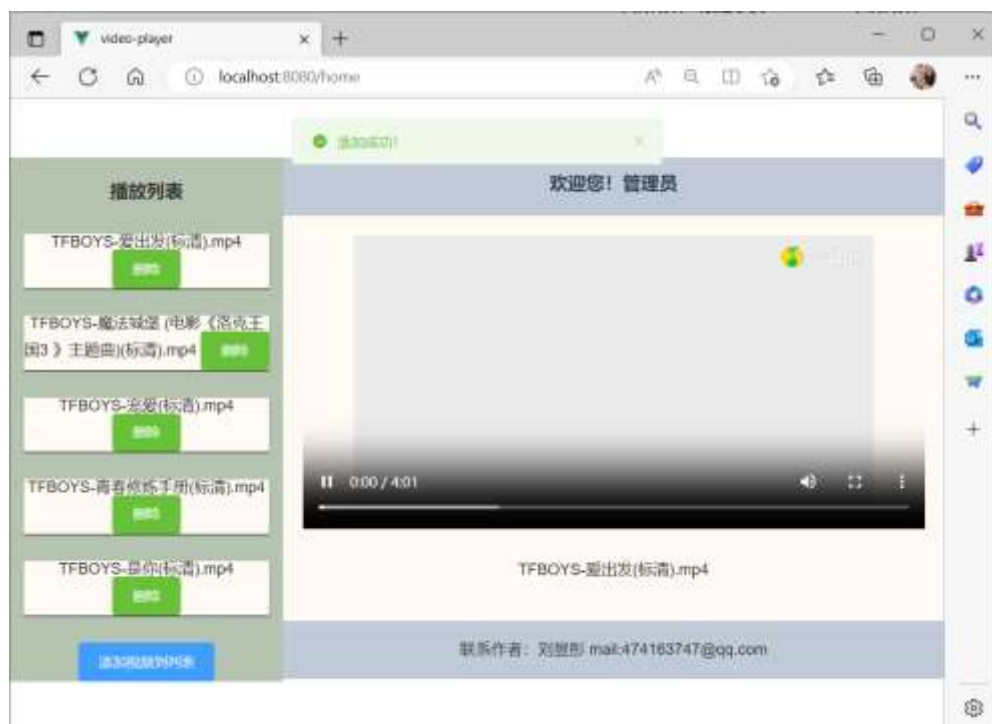
点击添加视频到列表进入 Modify 页面:



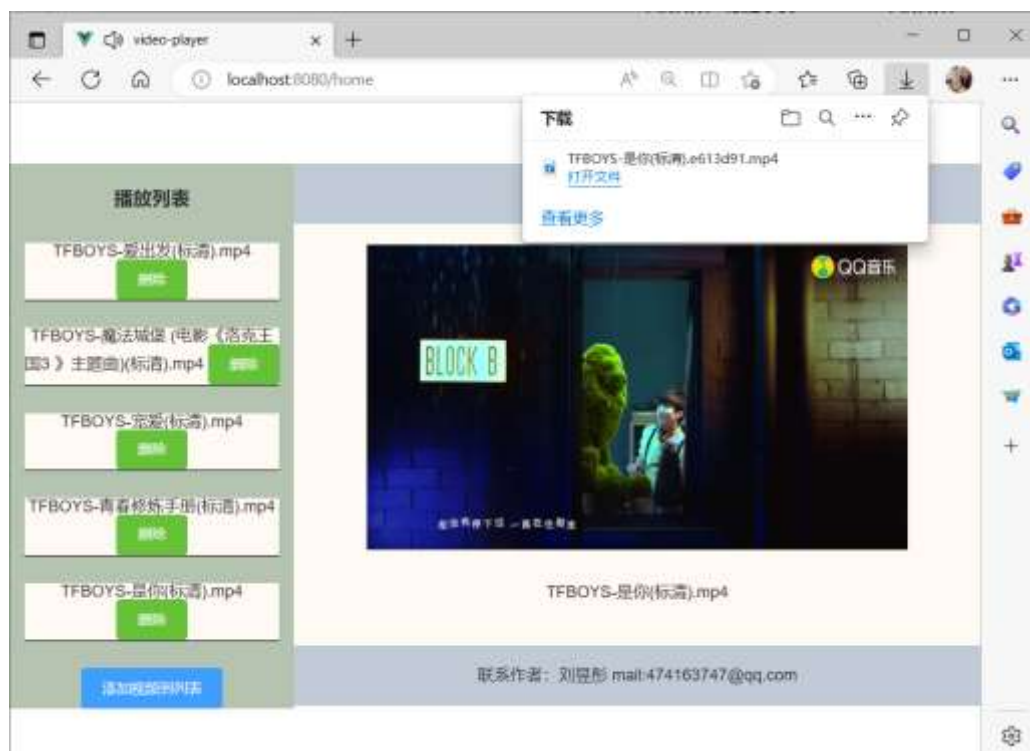
输入信息：



可以看到，新的视频已经添加进来了，点击就可以播放：



还有下载功能，但这个其实是 video 标签自带的。点击旁边那三个小点，就能下载。



下面我们介绍一下普通的用户界面，相比于管理员少了删除和添加功能，可以看到 tfboys-是你.mp4 在被管理员添加后，出现在了这里：



四、心得体会

之前做项目一般都是以小组为单位，然后大家分工负责前后端，但是这次的实验都是以个人为单位的，第一次自己一个人设计整个 web，虽然这次的作业很小，但是也很有成就感，开发起来又一次体会到了编程的乐趣。

但是在开发的时候也遇到了很多困难，尤其是我对 javascript 的语法并不是很熟悉，所以会碰到一些奇怪的问题，未来还是要多学习，多写，才能不断地成长。

总而言之，这次实验收获了很多，也给了我很多启发。

装

订

线