**摘要：**靠谱新闻APP是包含海量资讯的新闻服务平台,真实反映每时每刻的新闻热点。您可以搜索新闻事件、热点话题、人物动态、产品资讯等,快速了解它们的最新进展，24小时不间断提供最快捷,最全面,最丰富的新闻视听与互动服务。靠谱新闻APP将真实、贴实际的新闻内容展示给浏览者，同时实现了“登录注册”、“新闻简介”、“新闻内容”、“新闻视频”、“网友评论”、“评分订阅”、“收藏点赞”等功能模块。

**关键词：新闻、APP、页面设计**

目录

[引言 1](#_Toc91270980)

[1.1 项目背景 1](#_Toc91270981)

[1.2项目目的和意义 1](#_Toc91270982)

[1.2.1 项目目的 1](#_Toc91270983)

[1.2.2 项目意义 1](#_Toc91270984)

[1.3 设计思路 2](#_Toc91270985)

[第二章 可行性分析 3](#_Toc91270986)

[第三章 需求分析 4](#_Toc91270987)

[3.1功能要求 4](#_Toc91270988)

[3.2 APP要求 4](#_Toc91270989)

[第四章 系统设计 5](#_Toc91270990)

[4.1系统前端和后端设计流程 5](#_Toc91270991)

[4.2数据库设计 6](#_Toc91270992)

[第五章 系统详细设计 10](#_Toc91270993)

[5.1 APP前端功能模块 10](#_Toc91270994)

[结论 43](#_Toc91270995)

[6.1 总结 43](#_Toc91270996)

[6.2 展望 43](#_Toc91270997)

[参考文献 45](#_Toc91270998)

[致谢 46](#_Toc91270999)

# 引言

## 1.1 项目背景

在设计作品开始时，我想做是一款关于新闻的APP，并准备作为毕业设计的题目进行完善，但在对同类型现有的新闻APP进行调研的过程中我发现，APP的基本功能分为，“登录模块”，“查看新闻模块”，“编辑新闻模块”，“添加新闻模块”，但在添加发布新闻上有些困难，我们没有完全的掌握，在与指导老师沟通后，转换思路，向新闻详情里添加功能。

靠谱新闻的主要的目的是向浏览者展示真实、可靠的信息，提供24小时滚动报道国内、国际及社会新闻。靠谱新闻APP的开发主要包括后台数据库的建立以及前端页面的搭建两个方面。主要采用 SQL Server 作为后台数据库，VScode为前端开发工具，对APP开发进行了深入学习，完成了“登录”、“注册”、“订阅”、“收藏”、“评论”、“评分”、“历史记录”、“搜索”、“积分”等功能模块。

## 1.2项目目的和意义

### 1.2.1 项目目的

开发靠谱新闻APP，为促进社会发展，提供最新、真实的消息功能：

1、该APP浏览真实内容、社会活动、国家政治。

2、查看新闻事件，热点话题，人物动态，产品咨询。

3、新闻报道，新闻评论，新闻副刊。

### 1.2.2 项目意义

　近年来，随着新媒体的崛起，纸媒受到了重大打击，纸媒发展的道路越来越窄，面临前所未有的挑战。纸媒用户大量流失，广告切换、广告收入下降、经营压力增加、纸媒转换成为大势所趋。在激增的新信息交互时代，以纸媒为代表的传统媒体的光荣消失了，迈出了数字转换和媒体间发展的步伐，媒体融合的新时代到来了。对纸媒体来说，“这是最好的时代，也是最坏的时代。”面对“寒冬”的困境，传统纸媒积极拥抱变化，在未来的新环境中谋求新的发展。

### 1.3 设计思路

本文主要使用Vue框架和Mysql数据库，设计实现靠谱新闻APP。

用户可以通过注册登录来访问此APP，登陆后用户可以根据导航栏选择自己感兴趣的选项。其中包括“新闻文章”、“新闻视频”、“好友”、“个人”功能模块

首页：首页包含了新闻文章详情浏览，最新的新闻资讯以及经典的资讯展示。

好友：用户之间分享。

新闻视频：包含视频和文章详情，浏览者的评论和评分、收藏、订阅、评分。

个人：登录、注册、退出登录、修改个人资料、历史记录、订阅记录、收藏记录、学习积分。

# 第二章 可行性分析

该系统的可行性分析包括以下几个方面的内容。

**技术可行性：**

技术上的可行性分析主要分析技术条件能否顺利完成开发工作，硬、软件能否满足开发者的需要等。

该系统采用Vue框架，使用了node.js、JavaScript、HTML、CSS3等语言进行开发，并在项目开发中使用了Mysql Server、Vant等技术，通过Vant组件来实现APP的页面设计、功能的实现；通过Mysql Server技术来实现数据库的访问，数据库表可以通过面向对象的方式来创建；通过Node.js技术来整合前两项技术，给前两项技术提供支持，以达到最佳的效果，使你能够编写更干净、更可管理、并且更易于测试的代码。因此，系统的软件开发平台已经成熟可行。

**管理可行性**：

APP使用简单方便，使用对象是年轻人，他们基本上都可以正确添加资料。APP使用的素材基本都是真实，完整的，而且发布资料是有专门的人员添加，为APP发布的信息提供保障。

**运行可行性**：

使用目前流行的技术，对软硬的要求并不高，运行投入也相对较少，使用人员素质较高，系统运行一直比较稳定。因此，本系统在运行上是可行的。

**法律可行性**：

APP使用资料都是经过实地取材和当地的允许，APP整体的都页面布局是原创，不会涉及抄袭的问题，更无任何与法律相抵触的方面。因此，本APP在法律上是可行的。

综上所述，APP开发目标已明确，在技术和经济等方面都可行，并且投入少、效益好。因此APP的开发是完全可行。

# 第三章 需求分析

## 3.1功能要求

APP包括：首页、好友、新闻视频、个人。

首页：新闻列表、轮播、搜索、点赞、阅读量、评论。

消息：查看用户信息。

新闻视频：包含视频和文章详情，浏览者的评论和评分、收藏、订阅、评分。

个人：登录、注册、退出登录、修改个人资料、历史记录、订阅记录、收藏记录、学习积分。

## 3.2 APP要求

由于靠谱新闻APP都是真实的，再经过深度挖掘，所以靠谱新闻APP的素材要真实详细，资料收集比较繁琐：

（1）真实性：APP上的每一张新闻照片都是真实存在的，不能弄虚假信息。

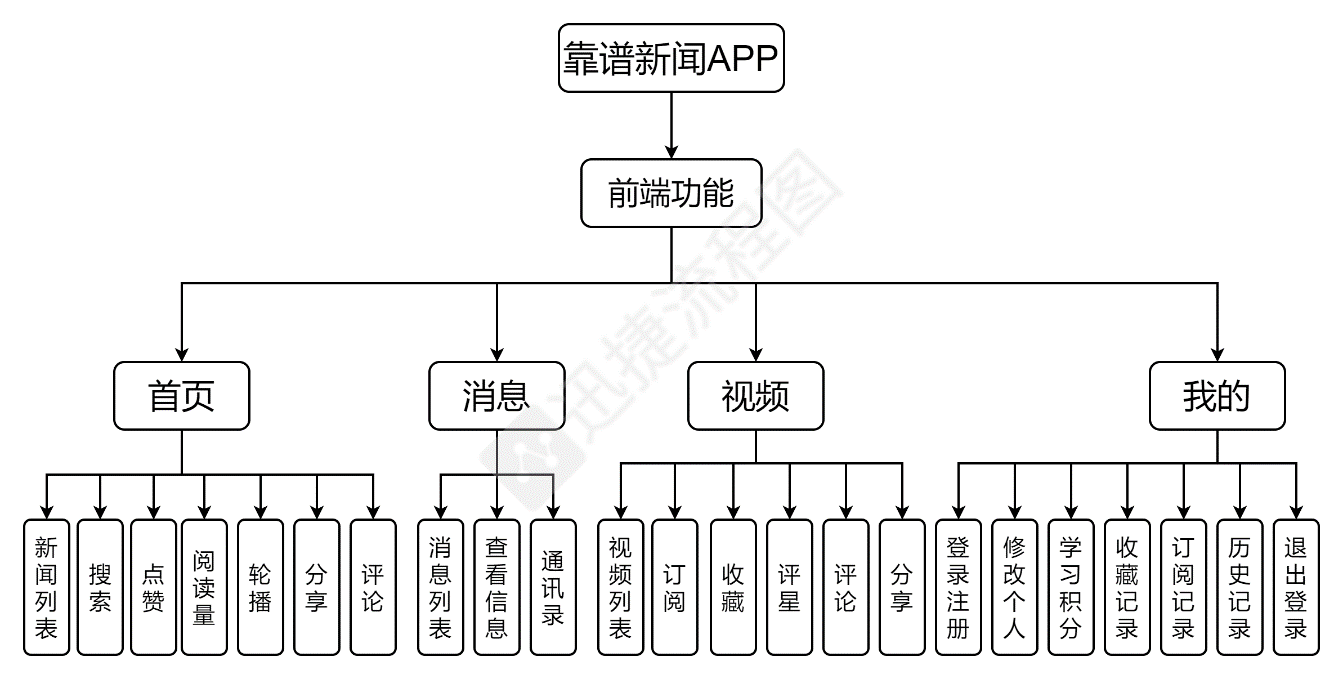
（2）一致性：APP的数据要保证一致性、准确性，当某一数据表中记录改变，与之相关联的数据表也随之变化。

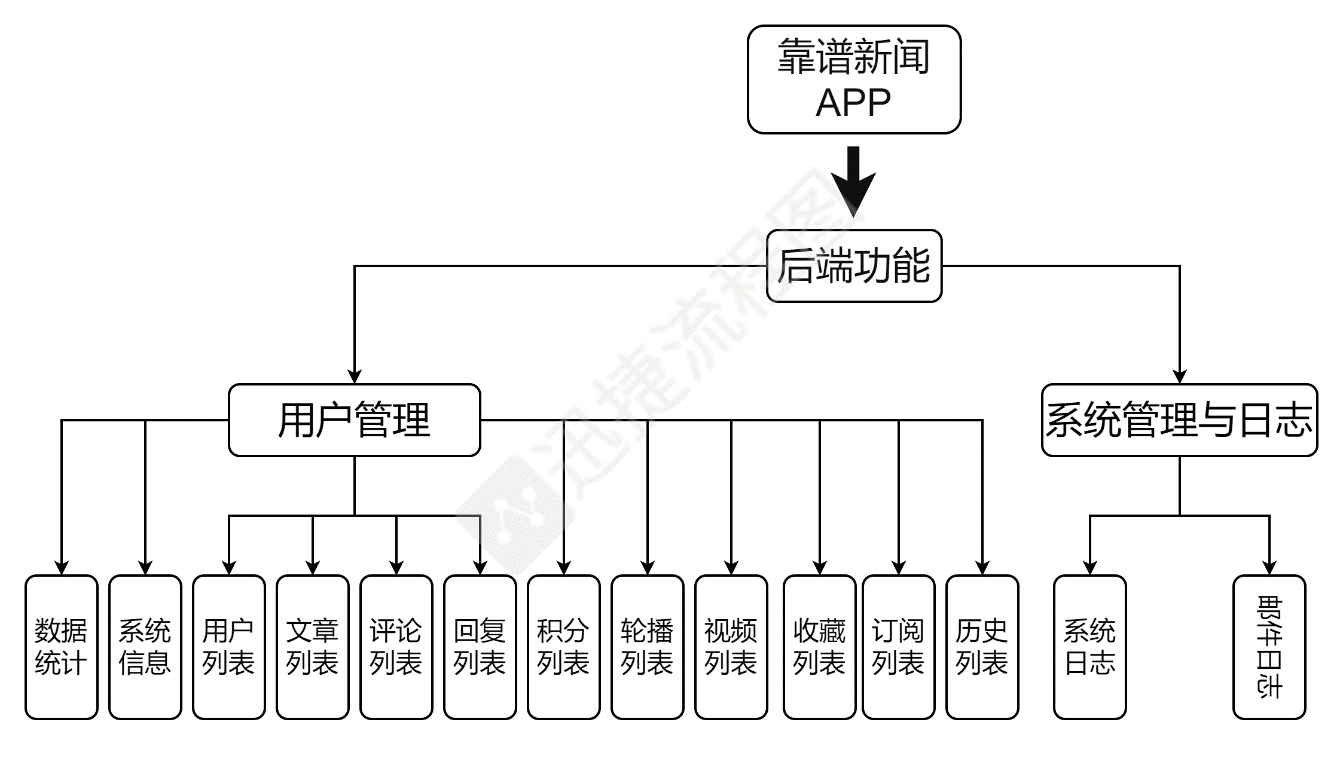
（3）明确性：APP的资料收集都要围绕“真实”的特色，不能东西拉扯胡乱拼凑，不能弄虚作假。

# 第四章 系统设计

根据系统功能分析：本系统从总体上分为前端和后端两大部分。

## 4.1系统前端和后端设计流程





## 4.2数据库设计

根据网站的需求分析，本网站使用MySQL数据库[3]保存数据，数据库名为photography，然后设计数据库表以及字段。网站涉及到的数据表有13个，表名分别是：t\_user表，t\_article表，t\_collection表，t\_comment表，t\_history表，t\_homelist表，t\_log表，t\_message表，t\_point表，t\_reply表，t\_subscribe表，t\_swiper表， t\_video表。

其中

用户表（t\_user表）用来保存用户的信息，用户表的结构如表4-1所示。

**表4-1 用户表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 备注 | 字段名 | 数据类型 | 字段类型 | 长度 | 键值 | 是否为空 |
| id | id | int(11) | int |  | PRI | NO |
| 昵称 | nickname | varchar(255) | varchar | 255 |  | YES |
| 用户名 | username | varchar(255) | varchar | 255 | UNI | YES |
| 密码 | pwd | varchar(255) | varchar | 255 |  | YES |
| 电话 | phone | varchar(11) | varchar | 11 |  | YES |
| 头像 | headImg | varchar(255) | varchar | 255 |  | YES |
| 二维码图片 | qrcodeImg | varchar(255) | varchar | 255 |  | YES |
| 性别 | sex | char(1) | char | 1 |  | YES |
| 生日 | birthday | date | date |  |  | YES |
| 地址 | address | varchar(255) | varchar | 255 |  | YES |
| 状态(0正常，1封禁) | status | varchar(6) | varchar | 6 |  | YES |
| 身份证 | card | varchar(18) | varchar | 18 |  | YES |
| 邮箱 | email | varchar(255) | varchar | 255 |  | YES |
| salt | salt | varchar(255) | varchar | 255 |  | YES |

新闻文章表（t\_article表）用来保存新闻，新闻文章表的结构如表4-2所示

**表4-2新闻文章表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 备注 | 字段名 | 数据类型 | 字段类型 | 长度 | 键值 | 是否为空 |
| ID | id | int(11) | int |  | PRI | NO |
| 文章标题 | title | varchar(255) | varchar | 255 |  | YES |
| 文章内容 | content | text | text | 65535 |  | YES |
| 文章图片 | img | varchar(255) | varchar | 255 |  | YES |
| 文章图片备注 | imgnote | varchar(255) | varchar | 255 |  | YES |
| 文章类型(0正常，1封禁) | status | varchar(255) | varchar | 255 |  | YES |
| 文章备注 | note | varchar(255) | varchar | 255 |  | YES |
| 文章发布时间 | time | timestamp | timestamp |  |  | NO |
| 文章点击（阅读） | clickcount | int(11) | int |  |  | YES |
| 文章喜欢（点赞） | likecount | int(11) | int |  |  | YES |

收藏表（t\_comment表）用来保存收藏，收藏表的结构如表4-3所示

**表4-3收藏表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 备注 | 字段名 | 数据类型 | 字段类型 | 长度 | 键值 | 是否为空 |
| id | id | int(11) | int |  | PRI | NO |
| 收藏标题 | title | varchar(255) | varchar | 255 |  | YES |
| 用户id | userid | int(11) | int |  |  | YES |
| 文章id | articleid | int(11) | int |  |  | YES |
| 视频id | videoid | int(11) | int |  |  | YES |
| 收藏时间 | time | timestamp | timestamp |  |  | NO |

评论表（t\_comment表）用来保存评论信息，评论表的结构如表4-4所示

**表4-4 评论表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 备注 | 字段名 | 数据类型 | 字段类型 | 长度 | 键值 | 是否为空 |
| id | commentid | int(11) | int |  | PRI | NO |
| 评论用户id | userid | int(11) | int |  |  | YES |
| 文章id | articleid | int(11) | int |  |  | YES |
| 视频id | videoid | int(11) | int |  |  | YES |
| 评论 | content | varchar(255) | varchar | 255 |  | YES |
| 点赞 | thumbsup | tinyint(4) | tinyint |  |  | YES |
| 状态 | status | varchar(255) | varchar | 255 |  | YES |
| 评论时间 | time | timestamp | timestamp |  |  | NO |
| 二级id | parentid | int(11) | int |  |  | YES |

历史记录表（t\_history表）用来保存历史记录，历史记录表的结构如表4-5所示

**表4-5 历史记录表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 备注 | 字段名 | 数据类型 | 字段类型 | 长度 | 键值 | 是否为空 |
| id | id | int(11) | int |  | PRI | NO |
| 用户id | userid | int(11) | int |  |  | YES |
| 历史标题 | title | varchar(255) | varchar | 255 |  | YES |
| 文章id | articleid | int(11) | int |  |  | YES |
| 视频id | videoid | int(11) | int |  |  | YES |
| 历史时间 | time | timestamp | timestamp |  |  | NO |

积分表（t\_point表）用来保存积分，积分表的结构如表4-6所示

**表4-6 积分表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 备注 | 字段名 | 数据类型 | 字段类型 | 长度 | 键值 | 是否为空 |
| id | id | int(11) | int |  | PRI | NO |
| 点数 | dpoint | int(11) | int |  |  | YES |
| 点数id | dpointId | int(11) | int |  |  | YES |
| 用户id | fuserid | int(11) | int |  | UNI | YES |

订阅表（t\_subscribe表）用来保存订阅信息，订阅表的结构如表4-7所示

**表4-7 订阅表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 备注 | 字段名 | 数据类型 | 字段类型 | 长度 | 键值 | 是否为空 |
| id | id | int(11) | int |  | PRI | NO |
| userid | userid | int(255) | int |  |  | YES |
| fuserid | fuserid | int(11) | int |  |  | YES |
| 订阅时间 | time | timestamp | timestamp |  |  | NO |

轮播表（t\_swiper表)用来保存轮播图片，轮播表的结构如表4-8所示

**表4-8 轮播表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 备注 | 字段名 | 数据类型 | 字段类型 | 长度 | 键值 | 是否为空 |
| id | id | int(11) | int |  | PRI | NO |
|  | articleid | int(11) | int |  |  | YES |
| 轮播图 | img | varchar(255) | varchar | 255 |  | YES |
| 路由名称 | routename | varchar(255) | varchar | 255 |  | NO |
| 重新标记 | reamrk | varchar(255) | varchar | 255 |  | YES |

视频表（t\_video表)用来保存视频，视频表的结构如表4-9所示

**表4-9视频表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 备注 | 字段名 | 数据类型 | 字段类型 | 长度 | 键值 | 是否为空 |
| id | id | int(11) | int |  | PRI | NO |
| 视频标题 | title | varchar(255) | varchar | 255 |  | YES |
| 视频来源 | pathurl | varchar(255) | varchar | 255 |  | YES |
| 封面图 | img | varchar(255) | varchar | 255 |  | YES |
| 视频来源 | from | varchar(255) | varchar | 255 |  | YES |
| 视频简介 | introduction | varchar(255) | varchar | 255 |  | YES |
| 视频点击（阅读） | clickcount | int(11) | int |  |  | YES |
| 视频喜欢（点赞） | likecount | int(11) | int |  |  | YES |
| 星星 | star | varchar(255) | varchar | 255 |  | YES |
| 发布时间 | time | timestamp | timestamp |  |  | NO |
| userid | userid | int(11) | int |  |  | YES |

# 第五章 系统详细设计

## 5.1 APP前端功能模块

1）学习积分模块：第一次进入学习积分页面会自动注册积分表，会自动返回重新进入方可正常使用学习积分页面**如图5-1所示**



**图5-1 学习积分页面**

关键代码：

|  |  |
| --- | --- |
|  | import { getpointById, addPoint } from "@/api/point.js"; |
|  | import { AddNewPoint } from "@/api/user.js"; |
|  |  |
|  | import { mapState } from "vuex"; // mapGetters |
|  | import { Toast } from "vant"; |
|  | import { Dialog } from "vant"; |
|  |  |
|  | export default { |
|  | name: "message\_add", |
|  | components: {}, |
|  | props: { |
|  | fuserid: Number |
|  | }, |
|  | data() { |
|  | return { |
|  | points: {} |
|  | }; |
|  | }, |
|  |  |
|  | created() { |
|  | this.getpointInfo(); |
|  | }, |
|  |  |
|  | computed: { |
|  | ...mapState("user", ["user", "Login"]) |
|  | }, |
|  | methods: { |
|  | async getpointInfo() { |
|  | // 获取积分详情数据 |
|  | let userid = this.user.id; |
|  | console.log(userid); |
|  | if (userid === undefined) { |
|  | console.log("未登录"); |
|  | Toast.fail("还没登录呢"); |
|  | } else { |
|  | try { |
|  | let params = { fuserid: this.user.id }; |
|  | // console.log(params); |
|  | let result = await getpointById(params); |
|  | // console.log(result); |
|  | if (result.code === "0") { |
|  | this.points = result.data; //数组的第一个元素 |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | Dialog.alert({ |
|  | confirmButtonText: "返回页面", |
|  | title: "提示", |
|  | message: |
|  | "因为你是第一次进入积分系统，暂无积分，重新进入即可获取积分详情" |
|  | }).then(() => { |
|  | this.$router.back(); |
|  | }); |
|  | let params = { fuserid: this.user.id }; |
|  | let point = await AddNewPoint(params); |
|  | console.log(point); |
|  | this.dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | } |
|  | }, //======================================= |
|  | async addpointitem() { |
|  | // 积分加一 |
|  | try { |
|  | let params = { fuserid: this.user.id }; |
|  | // console.log(params); |
|  | let result = await addPoint(params); |
|  | // console.log(result); |
|  | if (result.code === "0") { |
|  | console.log("积分加一"); |
|  | this.getpointInfo(); |
|  | Toast.success("积分+1"); |
|  | // history.go(1); |
|  | // location.reload(); |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, |
|  | m\_search() { |
|  | this.$router.replace("/message\_search"); |
|  | } |
|  | } |
|  | }; |

1. 收藏模块：用户在点击新闻或视频页面下方收藏按钮，将新闻收藏到这个页面，点击收藏页面的新闻将会跳转到当前的新闻页面**如图5-2所示**

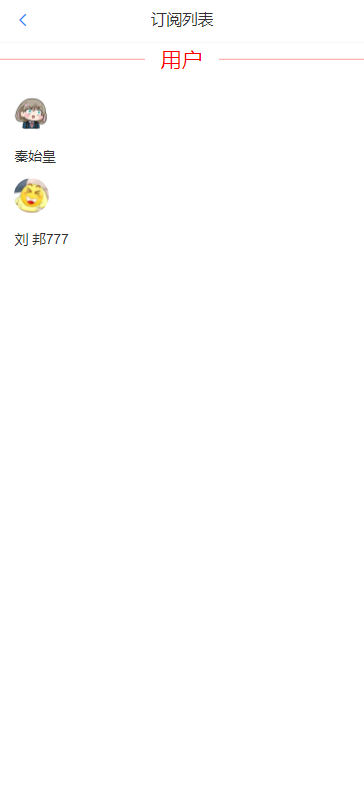


**图5-2 收藏**

关键代码：

|  |  |
| --- | --- |
|  | import { getCollectionListArticle, DelCollection } from "@/api/collection.js"; |
|  | import { mapState } from "vuex"; // mapGetters |
|  |  |
|  | export default { |
|  | data() { |
|  | return { |
|  | CollectionItems: "" |
|  | }; |
|  | }, |
|  | created() { |
|  | this.getCollectionItems(); |
|  | }, |
|  | computed: { |
|  | ...mapState("user", ["user", "isLogin"]) |
|  | }, |
|  | methods: { |
|  | async getCollectionItems() { |
|  | try { |
|  | let userid = { userid: this.user.id }; |
|  | let result = await getCollectionListArticle(userid); |
|  | console.log(result.data); |
|  | if (result.code === "0") { |
|  | this.CollectionItems = result.data; |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.$dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, //======================================= |
|  |  |
|  | async DelCollectionItems(id) { |
|  | try { |
|  | let params = { id }; |
|  | let result = await DelCollection(params); |
|  | console.log(result); |
|  | if (result.code === "0") { |
|  | this.$toast("删除成功！"); |
|  | history.go(-1); |
|  | // this.loadData(); //重新加载购物车数据 |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.$dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, //======================================= |
|  | async jump(articleid, videoid) { |
|  | console.log(articleid, videoid); |
|  | //判断是文章还是视频 |
|  | if (videoid === 0) { |
|  | console.log("是新闻页面"); |
|  | this.$router.push("/goods/detail/" + articleid); |
|  | } else if (articleid === 0) { |
|  | console.log("是视频页面"); |
|  | // this.$router.push("/videos/detail/" + videoid); |
|  | this.$router.push({path: "/videos/detail/" + videoid}); |
|  | } |
|  | } |
|  | } |
|  | }; |

1. 订阅模块：订用户后悔显示在订阅列表关键代码**如图5-3所示**

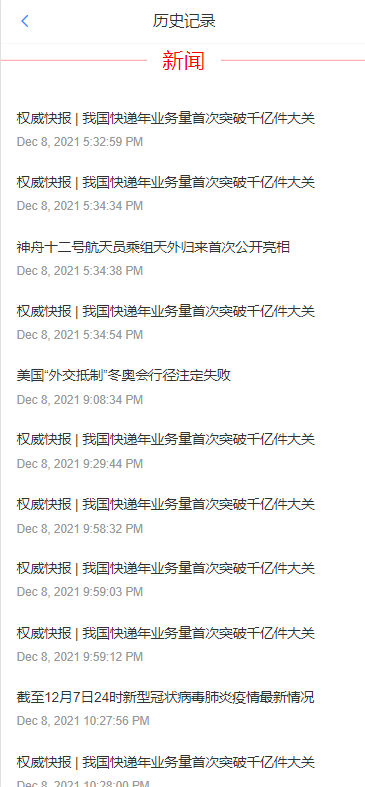


**图5-3 订阅**

关键代码：

|  |  |
| --- | --- |
|  | import { |
|  | DelSubscribeByFuserid, |
|  | getSubscribeNewList |
|  | } from "@/api/collection.js"; |
|  |  |
|  | import { mapState } from "vuex"; // mapGetters |
|  |  |
|  | export default { |
|  | data() { |
|  | return { |
|  | Subscribeitem: "" |
|  | }; |
|  | }, |
|  | created() { |
|  | this.getSubscribeitem(); |
|  | }, |
|  | computed: { |
|  | ...mapState("user", ["user", "isLogin"]) |
|  | }, |
|  | methods: { |
|  | async getSubscribeitem() { |
|  | // 获取订阅列表 |
|  | try { |
|  | let userid = { userid: this.user.id }; |
|  | let result = await getSubscribeNewList(userid); |
|  | console.log(result); |
|  | if (result.code === "0") { |
|  | this.Subscribeitem = result.data; |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.$dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, //======================================= |
|  |  |
|  | async DelSubscribeItems(userid, fuserid) { |
|  | // 删除订阅 |
|  | try { |
|  | let params = { userid, fuserid }; |
|  | let result = await DelSubscribeByFuserid(params); |
|  | console.log(result); |
|  | if (result.code === "0") { |
|  | this.$toast("删除成功！"); |
|  | history.go(-1); |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.$dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | } //======================================= |
|  | } |
|  | }; |

1. 历史记录模块：用户在点击新闻后会添加一条记录到历史记录中**如图5-4所示**



**图5-4 历史记录**

关键代码：

|  |  |
| --- | --- |
|  | import { getHistoryListArticle, DelHistory } from "@/api/collection.js"; |
|  | import { mapState } from "vuex"; // mapGetters |
|  |  |
|  | export default { |
|  | data() { |
|  | return { |
|  | HistoryList: "" |
|  | }; |
|  | }, |
|  | created() { |
|  | this.getHistoryItems(); |
|  | }, |
|  | computed: { |
|  | ...mapState("user", ["user", "isLogin"]) |
|  | }, |
|  | methods: { |
|  | async getHistoryItems() { |
|  | try { |
|  | let userid = { userid: this.user.id }; |
|  | let result = await getHistoryListArticle(userid); |
|  | console.log(result.data); |
|  | if (result.code === "0") { |
|  | this.HistoryList = result.data; |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.$dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, //======================================= |
|  |  |
|  | async DelHistoryList(id) { |
|  | try { |
|  | let params = { id }; |
|  | let result = await DelHistory(params); |
|  | console.log(result); |
|  | if (result.code === "0") { |
|  | this.$toast("删除成功！"); |
|  | history.go(-1); |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.$dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, //======================================= |
|  | async jump(articleid, videoid) { |
|  | console.log(articleid, videoid); |
|  | //判断是文章还是视频 |
|  | if (videoid === 0) { |
|  | console.log("是视频页面"); |
|  | this.$router.push("/goods/detail/" + articleid); |
|  | } else if (articleid === 0) { |
|  | console.log("是视频页面"); |
|  | this.$router.push("/videos/detail/" + videoid); |
|  | } |
|  | } |
|  | } |
|  | }; |

1. 首页轮播和新闻列表 **如图5-5所示**



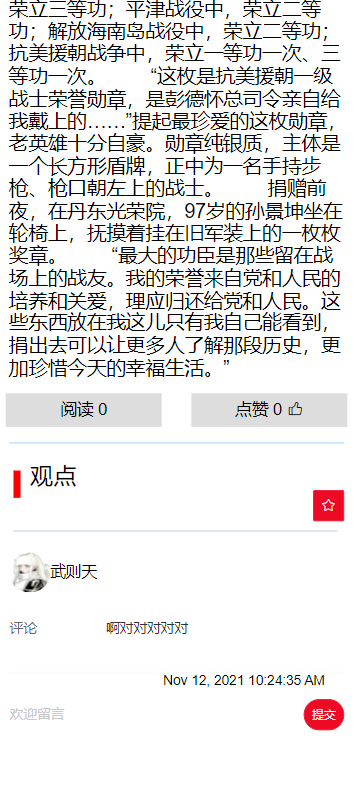
**图5-5 历史记录**

关键代码：

|  |  |
| --- | --- |
|  | import { getCarouselList, getcontentList } from "@/api/home.js"; |
|  | import { addClickcount } from "@/api/goods.js"; |
|  | import { AddHistory } from "@/api/collection.js"; |
|  | import { mapState } from "vuex"; // mapGetters |
|  | import FooterNav from "@/components/FooterNav.vue"; |
|  |  |
|  | export default { |
|  | components: { |
|  | Swiper, |
|  | "footer-nav": FooterNav |
|  | }, |
|  | data() { |
|  | return { |
|  | carouselItems: [], |
|  | contentItems: [], |
|  | HistoryItem: { |
|  | userid: "", |
|  | articleid: "", |
|  | videoid: "", |
|  | title: "" |
|  | } |
|  | }; |
|  | }, |
|  |  |
|  | created() { |
|  | this.getCarouselItems(); //===================================== |
|  | this.getcontentItems(); //===================================== |
|  | }, |
|  | computed: { |
|  | ...mapState("user", ["user", "isLogin"]) |
|  | }, |
|  | methods: { |
|  | // 获取轮播 |
|  | async getCarouselItems() { |
|  | try { |
|  | let result = await getCarouselList(); |
|  | if (result.code === "0") { |
|  | this.carouselItems = result.data; |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.$dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, //======================================= |
|  | // 获取新闻列表 |
|  | async getcontentItems() { |
|  | try { |
|  | let result = await getcontentList(); |
|  |  |
|  | if (result.code === "0") { |
|  | this.contentItems = result.data; |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.$dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, //======================================= |
|  |  |
|  | async AddHistoryitem(id, title) { |
|  | //添加历史记录 |
|  | let userid = this.user.id; |
|  | console.log(userid); |
|  | if (userid === undefined) { |
|  | console.log("未登录"); |
|  | } else { |
|  | try { |
|  | this.HistoryItem.userid = this.user.id; |
|  | this.HistoryItem.articleid = id; |
|  | this.HistoryItem.videoid = "0"; |
|  | this.HistoryItem.title = title; |
|  |  |
|  | let params = this.HistoryItem; |
|  |  |
|  | let result = await AddHistory(params); |
|  | let qweid = { id: id }; |
|  | let result1 = await addClickcount(qweid); |
|  |  |
|  | console.log(result); |
|  | console.log(result1); |
|  |  |
|  | if (result.code === "0") { |
|  | console.log("添加历史记录成功"); |
|  | } |
|  | } catch (error) { |
|  | this.$dialog.alert({ title: "添加失败", message: error }); |
|  | } |
|  | } |
|  | }, |
|  |  |
|  | integral() { |
|  | this.$router.push("/integral"); |
|  | }, |
|  | handleOpen(key, keyPath) { |
|  | console.log(key, keyPath); |
|  | }, |
|  | handleClose(key, keyPath) { |
|  | console.log(key, keyPath); |
|  | } |
|  | } |
|  | }; |

1. 新闻详情页面：阅读数量、点赞、评论、收藏

**如图5-6-1和5-6-2所示**



**图5-6-1 新闻详情 图5-6-2 阅读、点赞、评论、收藏**

关键代码：

|  |  |
| --- | --- |
|  | import { getGoodsDetailById, addLikecount } from "@/api/goods.js"; |
|  |  |
|  | import { AddComment, getCommentListArticle } from "@/api/comment.js"; |
|  | import { AddCollection } from "@/api/collection.js"; |
|  |  |
|  | import { mapState } from "vuex"; // mapGetters |
|  | import { Toast } from "vant"; |
|  |  |
|  | export default { |
|  | data() { |
|  | return { |
|  | //点赞 |
|  | clicklike: false, |
|  | //收藏 |
|  | collectionItem: { |
|  | userid: "", |
|  | articleid: "", |
|  | videoid: "", |
|  | title: "" |
|  | }, |
|  | //评论list |
|  | Comment: {}, |
|  | //上传数组 |
|  | articleitem: { |
|  | userid: "", |
|  | articleid: "", |
|  | content: "" |
|  | }, |
|  | loading: true, |
|  | // 新闻list |
|  | goods: {}, |
|  | imgurl: "https://img1.baidu.com/it/u=279030105,1477287187&fm=26&fmt=auto", |
|  |  |
|  | showShare: false, |
|  | options: [ |
|  | { name: "微信", icon: "wechat" }, |
|  | { name: "微博", icon: "weibo" }, |
|  | { name: "复制链接", icon: "link" }, |
|  | { name: "分享海报", icon: "poster" }, |
|  | { name: "二维码", icon: "qrcode" } |
|  | ] |
|  |  |
|  | //id: this.$route.params.id, |
|  | }; |
|  | }, |
|  |  |
|  | props: { |
|  | id: Number |
|  | }, |
|  |  |
|  | created() { |
|  | // 调用方法 |
|  | this.getGoodsInfo(); |
|  | this.getCommentInfo(); |
|  | }, |
|  | computed: { |
|  | ...mapState("user", ["user", "isLogin"]) |
|  | }, |
|  |  |
|  | methods: { |
|  | // 定义方法 |
|  | onSelect(option) { |
|  | this.$toast(option.name); |
|  | this.showShare = false; |
|  | }, |
|  | async getGoodsInfo() { |
|  | // 获取新闻详情数据 |
|  | try { |
|  | let params = { id: this.id }; |
|  | let result = await getGoodsDetailById(params); |
|  | // console.log(result); |
|  | if (result.code === "0") { |
|  | this.goods = result.data; //数组的第一个元素 |
|  | console.log(this.goods.id); |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, //======================================= |
|  | async getCommentInfo() { |
|  | // 获取评论详情数据 |
|  | try { |
|  | let params = { articleid: this.id }; |
|  | let result = await getCommentListArticle(params); |
|  | // console.log(result.data); |
|  | if (result.code === "0") { |
|  | this.Comment = result.data; |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, //======================================= |
|  | async onSubmit() { |
|  | // 添加评论 |
|  | if (this.articleitem.content === "") { |
|  | this.$toast("评论不能为空"); |
|  | return; |
|  | } |
|  | try { |
|  | this.articleitem.userid = this.user.id; |
|  | this.articleitem.articleid = this.id; |
|  | console.log(this.user.headImg); |
|  | let user = this.articleitem; |
|  | let result = await AddComment(user); //注意传递过去的参数user的属性名 |
|  | console.log(user); |
|  | if (result.code === "0") { |
|  | this.$toast("更新成功！"); |
|  | location.reload(); |
|  | return; |
|  | } |
|  | throw result; |
|  | } catch (error) { |
|  | this.$dialog.alert({ |
|  | title: "更新失败", |
|  | message: error |
|  | }); |
|  | } |
|  | }, |
|  | async AddCollectionitem(id, title) { |
|  | // 添加收藏 |
|  | let userid = this.user.id; |
|  | console.log(userid); |
|  | if (userid === undefined) { |
|  | console.log("未登录"); |
|  | Toast.fail("还没登录呢"); |
|  | } else { |
|  | try { |
|  | this.collectionItem.userid = this.user.id; |
|  | this.collectionItem.articleid = id; |
|  | this.collectionItem.videoid = "0"; |
|  | this.collectionItem.title = title; |
|  |  |
|  | let params = this.collectionItem; |
|  | console.log(params); |
|  | let result = await AddCollection(params); |
|  | if (result.code === "0") { |
|  | console.log("收藏成功"); |
|  | Toast.success("收藏成功"); |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | } |
|  | }, |
|  | async addLikecountitem() { |
|  | // 点赞 |
|  | if (this.clicklike == true) { |
|  | Toast.fail("已经点过赞了"); |
|  | } else { |
|  | try { |
|  | this.clicklike = true; |
|  | let params = { id: this.id }; |
|  | console.log(params); |
|  | let result = await addLikecount(params); |
|  | if (result.code === "0") { |
|  | console.log(result); |
|  | this.getGoodsInfo(); |
|  | Toast.success("点赞成功"); |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | } |
|  | } //======================================= |
|  | } |
|  | }; |

1. 消息列表，查看用户详情信息如**图5-7-1和5-7-2所示**



**图5-7-1消息列表 图5-7-2用户详情**

关键代码：

|  |  |
| --- | --- |
|  | async getUserAitem() { |
|  | try { |
|  | let result = await getUserA(); |
|  |  |
|  | if (result.code === "0") { |
|  | this.useritem = result.data; |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.$dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, |
|  | import { getDetailById } from "@/api/user.js"; |
|  | import { areaList } from "@vant/area-data"; |
|  |  |
|  | import { Toast } from "vant"; |
|  |  |
|  | import { mapState } from "vuex"; // mapGetters |
|  |  |
|  | export default { |
|  | data() { |
|  | return { |
|  | // 日期 |
|  | showPicker: false, |
|  | minDate: new Date(1980, 0, 1), |
|  | maxDate: new Date(2022, 0, 1), |
|  | // 地址 |
|  | showArea: false, |
|  | areaList, // 数据格式见 Area 组件文档 |
|  |  |
|  | userForm: { |
|  | //表单中的备注与数据库中用户表的备注要求一致 |
|  | id: null, |
|  | username: "", |
|  | nickname: "", |
|  | email: "", |
|  | phone: "", |
|  | sex: "男", |
|  | birthday: "", |
|  | address: "" |
|  | }, |
|  | headImg: [], |
|  | //规则:字母数字‘6~12 |
|  | regNamePwd: /^[a-z0-9]{6,12}$/i, |
|  | regPhone: /^((13[0-9])|(14[5|7])|(15([0-3]|[5-9]))|(18[0,5-9]))\d{8}$/, |
|  | regEmail: /^[a-zA-Z0-9\_-]+@[a-zA-Z0-9\_-]+(\.[a-zA-Z0-9\_-]+)+$/ |
|  | }; |
|  | }, |
|  | props: { |
|  | id: Number |
|  | }, |
|  | computed: { |
|  | ...mapState("user", ["user", "isLogin"]) |
|  | }, |
|  | async created() { |
|  | try { |
|  | let params = { id: this.id }; |
|  | let result = await getDetailById(params); |
|  | if (result.code === "0") { |
|  | this.userForm = result.data; |
|  | this.headImg = []; |
|  | this.headImg.push({ |
|  | url: this.userForm.headImg |
|  | }); |
|  | return; |
|  | } |
|  | throw result; |
|  | } catch (error) { |
|  | this.$dialog.alert({ |
|  | title: "获取用户数据失败", |
|  | message: error |
|  | }); |
|  | } |
|  | }, |
|  | methods: { |
|  | news() { |
|  | Toast("暂时不能发消息哦"); |
|  | } |
|  | } |
|  | }; |

1. 轮播、新闻列表**如图5-8所示**



**图5-8 视频列表**

关键代码

|  |  |
| --- | --- |
|  | import Swiper from "@/components/Swiper.vue"; |
|  | //import Mock from "mockjs"; //引入 |
|  | import FooterNav from "@/components/FooterNav.vue"; |
|  |  |
|  | import { getCarouselList, getvideoList } from "@/api/home.js"; |
|  | import { getDetailById } from "@/api/user.js"; |
|  |  |
|  | export default { |
|  | components: { |
|  | Swiper, |
|  |  |
|  | "footer-nav": FooterNav |
|  | }, |
|  |  |
|  | data() { |
|  | return { |
|  | carouselItems: [], |
|  | videoItems: [], |
|  | useriditem: [] |
|  | }; |
|  | }, |
|  |  |
|  | created() { |
|  | this.getCarouselItems(); //===================================== |
|  | this.getvideoItems(); //===================================== |
|  | }, |
|  | methods: { |
|  | // 获取轮播 |
|  | async getCarouselItems() { |
|  | try { |
|  | let result = await getCarouselList(); |
|  |  |
|  | if (result.code === "0") { |
|  | this.carouselItems = result.data; |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.$dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, //======================================= |
|  |  |
|  | // 获取视频列表 |
|  | async getvideoItems() { |
|  | try { |
|  | let result = await getvideoList(); |
|  |  |
|  | if (result.code === "0") { |
|  | this.videoItems = result.data; |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.$dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, //======================================= |
|  |  |
|  | //根据获取的userid获得用户名 |
|  | async getDetailByIditem() { |
|  | try { |
|  | let result = await getDetailById(); |
|  | if (result.code === "0") { |
|  | this.useriditem = result.data; |
|  | console.log(result); |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.$dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, //======================================= |
|  | integral() { |
|  | this.$router.push("/integral"); |
|  | } |
|  | } |
|  | }; |

1. 视频详情、订阅、收藏、评星、评论**如图5-9所示**



**图5-9视频详情**

关键代码

|  |  |
| --- | --- |
|  | import { |
|  | getvideosDetailById, |
|  | postUpdateStar, |
|  | getVideoFrom |
|  | } from "@/api/videos.js"; |
|  |  |
|  | import { AddCommentVideo, getCommentListVideo } from "@/api/comment.js"; |
|  |  |
|  | import { AddCollection, AddSubscribe } from "@/api/collection.js"; |
|  | import { mapState } from "vuex"; // mapGetters |
|  | import { Toast } from "vant"; |
|  |  |
|  | export default { |
|  | // name: "details", |
|  | data() { |
|  | return { |
|  | userid: "66", |
|  | nickname: "1", |
|  | videos: {}, |
|  | Comment: {}, |
|  | stars: { |
|  | id: "", |
|  | star: "" |
|  | }, |
|  | videoitem: { |
|  | userid: "", |
|  | videoid: "", |
|  | content: "" |
|  | }, |
|  | // 收藏 |
|  | collectionItem: { |
|  | userid: "", |
|  | articleid: "", |
|  | videoid: "", |
|  | title: "" |
|  | }, |
|  | Subscribeitem: { |
|  | userid: "", |
|  | fuserid: "" |
|  | }, |
|  | checked: true, |
|  | activeIcon: "", |
|  | inactiveIcon: "", |
|  | showShare: false, |
|  | options: [ |
|  | { name: "微信", icon: "wechat" }, |
|  | { name: "微博", icon: "weibo" }, |
|  | { name: "复制链接", icon: "link" }, |
|  | { name: "分享海报", icon: "poster" }, |
|  | { name: "二维码", icon: "qrcode" } |
|  | ], |
|  | playerOptions: { |
|  | playbackRates: [0.5, 1.0, 1.5, 2.0], // 可选的播放速度 |
|  | autoplay: true, // 如果为true,浏览器准备好时开始回放。 |
|  | muted: false, // 默认情况下将会消除任何音频。 |
|  | loop: false, // 是否视频一结束就重新开始。 |
|  | preload: "auto", // 建议浏览器在<video>加载元素后是否应该开始下载视频数据。auto浏览器选择最佳行为,立即开始加载视频（如果浏览器支持） |
|  | language: "zh-CN", |
|  | aspectRatio: "16:9", // 将播放器置于流畅模式，并在计算播放器的动态大小时使用该值。值应该代表一个比例 - 用冒号分隔的两个数字（例如"16:9"或"4:3"） |
|  | fluid: true, // 当true时，Video.js player将拥有流体大小。换句话说，它将按比例缩放以适应其容器。 |
|  | sources: [ |
|  | { |
|  | type: "video/mp4", // 类型 |
|  | src: "1" // url地址 |
|  | } |
|  | ], |
|  | poster: "", // 封面地址 |
|  | notSupportedMessage: "此视频暂无法播放，请稍后再试", // 允许覆盖Video.js无法播放媒体源时显示的默认信息。 |
|  | controlBar: { |
|  | timeDivider: true, // 当前时间和持续时间的分隔符 |
|  | durationDisplay: true, // 显示持续时间 |
|  | remainingTimeDisplay: false, // 是否显示剩余时间功能 |
|  | fullscreenToggle: true // 是否显示全屏按钮 |
|  | } |
|  | } |
|  | }; |
|  | }, |
|  | props: { |
|  | id: Number |
|  | }, |
|  | created() { |
|  | // 调用方法 |
|  | this.getvideoInfo(); |
|  | this.getCommentInfo(); |
|  | // this.playerOptions.sources.src = this.pathurl; |
|  | // console.log(this.pathurl); |
|  | }, |
|  | beforeDestroy() { |
|  | // console.log(this.playerOptions.sources[0].src); |
|  | }, |
|  | updated() { |
|  | this.getnickname(); |
|  | }, |
|  | methods: { |
|  | //上传星级 |
|  | async upstar() { |
|  | try { |
|  | this.stars.id = this.videos.id; |
|  | this.stars.star = parseFloat(this.videos.star); |
|  | let data = this.stars; |
|  | let result = await postUpdateStar(data); //注意传递过去的参数star的属性名 |
|  | if (result.code === "0") { |
|  | this.$toast("评分成功！"); |
|  | this.getvideoInfo(); |
|  | // history.go(-1); |
|  | return; |
|  | } |
|  | throw result; |
|  | } catch (error) { |
|  | this.$dialog.alert({ |
|  | title: "评分失败", |
|  | message: error |
|  | }); |
|  | } |
|  | }, |
|  | // 分享 |
|  | onSelect(option) { |
|  | this.$toast(option.name); |
|  | this.showShare = false; |
|  | }, |
|  |  |
|  | async onSubmit() { |
|  | // 添加评论 |
|  | if (this.videoitem.content === "") { |
|  | this.$toast("评论不能为空"); |
|  | return; |
|  | } |
|  | try { |
|  | this.videoitem.userid = this.user.id; |
|  | this.videoitem.videoid = this.id; |
|  | console.log(this.user.headImg); |
|  | let user = this.videoitem; |
|  | let result = await AddCommentVideo(user); //注意传递过去的参数user的属性名 |
|  | console.log(user); |
|  | if (result.code === "0") { |
|  | this.$toast("更新成功！"); |
|  | this.videoitem.content = ""; |
|  | this.getCommentInfo(); |
|  | return; |
|  | } |
|  | throw result; |
|  | } catch (error) { |
|  | this.$dialog.alert({ |
|  | title: "更新失败", |
|  | message: error |
|  | }); |
|  | } |
|  | }, |
|  |  |
|  | async getvideoInfo() { |
|  | // 获取视频详情数据 |
|  | try { |
|  | let params = { id: this.id }; |
|  | let result = await getvideosDetailById(params); |
|  | // console.log(result.data); |
|  | if (result.code === "0") { |
|  | this.videos = result.data; |
|  | this.playerOptions.sources[0].src = result.data.pathurl; |
|  | this.userid = result.data.userid; |
|  | this.videos.star = parseFloat(result.data.star); |
|  | console.log(this.videos.star); |
|  | // console.log(this.playerOptions.sources.src); |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, //======================================= |
|  |  |
|  | async getCommentInfo() { |
|  | // 获取评论详情数据 |
|  | try { |
|  | let params = { videoid: this.id }; |
|  | let result = await getCommentListVideo(params); |
|  | // console.log(result.data); |
|  | if (result.code === "0") { |
|  | this.Comment = result.data; |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | }, //======================================= |
|  | async AddCollectionitem(id, title) { |
|  | // 添加收藏 |
|  | let userid = this.user.id; |
|  | console.log(userid); |
|  | if (userid === undefined) { |
|  | console.log("未登录"); |
|  | Toast.fail("还没登录呢"); |
|  | } else { |
|  | try { |
|  | this.collectionItem.userid = this.user.id; |
|  | this.collectionItem.articleid = "0"; |
|  | this.collectionItem.videoid = id; |
|  | this.collectionItem.title = title; |
|  |  |
|  | let params = this.collectionItem; |
|  | console.log(params); |
|  | let result = await AddCollection(params); |
|  | if (result.code === "0") { |
|  | console.log("收藏成功"); |
|  | Toast.success("收藏成功"); |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | } |
|  | }, |
|  | async AddSubscribeitem(id) { |
|  | // 添加订阅 |
|  | let userid = this.user.id; |
|  | console.log(userid); |
|  | if (userid === undefined) { |
|  | console.log("未登录"); |
|  | Toast.fail("还没登录呢"); |
|  | } else { |
|  | try { |
|  | this.Subscribeitem.userid = this.user.id; |
|  | this.Subscribeitem.fuserid = id; |
|  |  |
|  | let params = this.Subscribeitem; |
|  | console.log(params); |
|  | let result = await AddSubscribe(params); |
|  | if (result.code === "0") { |
|  | console.log("订阅成功"); |
|  | Toast.success("订阅成功"); |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | } |
|  | }, |
|  | async getnickname() { |
|  | // 根据userid获得用户名 |
|  | try { |
|  | let params = { userid: this.userid }; |
|  | // console.log(this.userid); |
|  | let result = await getVideoFrom(params); |
|  | // console.log(result.data[0].nickname); |
|  |  |
|  | if (result.code === "0") { |
|  | this.nickname = result.data[0].nickname; |
|  | } else { |
|  | throw result.msg; |
|  | } |
|  | } catch (error) { |
|  | this.dialog.alert({ title: "错误", message: error }); |
|  | } |
|  | } //======================================= |
|  | }, |
|  |  |
|  | computed: { |
|  | ...mapState("user", ["user", "Login"]) |
|  | } |
|  | }; |

# 结论

## 6.1 总结

在本次系统开发过程中，由于我们在知识、经验方面都存在较大的不足，参考了他人的一些APP。因此，该系统必然会存在一些缺陷和不足。因为对APP的整体流程还不熟悉，分析不全面未能做到完全满足用户的需求。在系统功能上，未能实现部分功能，还有些问题没有考虑周到。尽管该APP存在很多不足，但是自己的想法，及亲自动手开发出来的，个人比较满意。

从这次APP的开发过程中，我学到了许多有用的东西，通过实践学到的东西比课堂上听到的更令人深刻难忘，感谢老师们的认真审查，同时我也很感谢在APP开发过程中替我解答疑惑的同学，严谨、负责、坚持、不惧困难的精神是本次APP开发给我最大的启示，任何一个很小的错误都可能导致程序的崩溃，特别是在数据库连接和代码方面稍有不慎就会出错。这次APP开发期间经历的困难与困难解决的过程中，对我今后生活学习有很大的帮助，最后预祝指导老师和小组同学们寒假愉快！

## 6.2 展望

虽然APP已基本开发完成，但该系统仍需不断完善和维护，通过吸取优秀项目的开发经验和他人意见，不断调整以达到最优。随着技术的不断进步，用户需求的不断改变，本系统在未来还有许多需要改进的地方，总结出以下几点：

1、前台界面的优化

前台界面是直接展现给用户的界面，因此美化的重要性不容小觑。本系统仍需在实际需求的情况下再次进行美化直至用户满意。

2、功能的完善

一个完整的APP应该还具发布新闻模块，但由于各方面条件的限制，本系统只开发了靠谱新闻APP的基础模块，因此该系统在将来还有很大的扩展空间。

综上所述，该APP未来要走的道路还有很长，需要开发者不断学习，改进，直到完善成为一个成熟的系统，这也是开发的最终目的。

# 参考文献

1. 贾祥素.SQL Server2012案例教程[M]. 清华大学出版社，2019.

[2] 李锴.新时期的Node。js[M]. 清华大学出版社 2018

[3] 吴菁 贾祥素 王海颖 .网页制作案例教程[M]. 清华大学出版社，2012.

[4] 黑马程序员 vue.js前端开发实战 [M]. 人民邮电出版社 2019

M:代表图书

# 致谢

在论文完成之际，首先感谢我的指导老师华铨平老师。感谢华铨平老师在选题、作品实现和论文写作等方面，提出了很多宝贵意见。每当我遇到自己解决不了的问题，总会想到我有这么一位老师能给我一些帮助，而当我去请教他时，他总会不厌其烦地帮助我，就算是一些很简单低级的问题华老师依然会给我耐心的讲解，遇到华老师不会的华老师也会耐心的帮我解决问题。华铨平老师以其严谨求实的治学态度、高度的敬业精神、孜孜以求的工作作风和大胆创新的进取精神对我产生了重大影响。他渊博的知识、开阔的视野和敏锐的思维给了我深深的启迪。正是因为华铨平老师的悉心指导和无私帮助，我才能完成此论文，并受益匪浅。

同时也要感谢我的授课教师和我的同学们，在他们的陪伴下，我才能在大学三年中有所收获，有所成长。能够有机会聆听各位老师的精彩讲课，得到各位老师的谆谆教导，必将对我今后的人生之路产生深刻的影响。在同窗三年的时间里，我和同学们共同学习，相互帮助，一起进步，这也都将成为一份珍贵的回忆。

最后要感谢的是我的父母，是他们培养了我，为我提供了良好的学习环境。在未来的日子里，我会更加努力地学习和工作，不辜负父母对我的殷殷期望。

分工

**小组成员**：庞天临、蒋传豪、张浩

庞天临主要实现功能

（1）后台接口开发

（2）后台管理系统开发

（3）Mysql数据库设计

（4）接口文档

（5）服务器配置

（6）apk打包

（7）前后端衔接

蒋传豪主要实现功能

（1） 前台app开发

（2） 前后端接口链接

（3） 部分Mysql数据库设计

（4） 前端vue框架开发

张浩主要实现功能

（1） 前台app开发

（2） 前端样式修改

（3） 数据库添加与补充

（4） PPT、图片、素材提供