

基于 Vue 的知 识分享网站设计与实现

陈昱竹 黄媛媛

四川大学锦城学院计算机与软件学院 四川 成都 611731

【摘要】随着越来越多的人使用互联网，知 识分享与交友软件层出不穷，传统的纸质报刊的受众变少，互联网信息量巨大，没有分类，人们想获取有效的知 识需要花费大量的时间。同时人们在互联网上获取信息是单方面的，人与人之间的沟通变少了。为了解决人们知 识搜索与分享的问题，设计了一个基于 Vue 框架与 Node.js 的知 识分享网站。本系统具备文章浏览，文章搜索，评论点赞等功能，经过测试，本系统可以满足使用网站用户的知 识分享、获取的需求，且用户体验度高，具有轻量，稳定，易实现等优点。

关键词：VueNode.js；知 识分享

1 引言

随着互联网的高速发展，越来越多的人使用手机，人们在手机上看新闻资讯，但是新闻资讯互动性不强，而且撰写者大多都是新闻记者，人们更需要一个能够进行咨询、知 识分享、互动的应用。为了更好地适应当下的需求，本项目设计了一个网站，每一个人都可以是分享者，分享内容从互联网专业知识到生活中的小问题，同时还有点赞，收藏，评论，关注等功能，这极大提高了分享者与阅读者的互动性，不仅可以让每一个人都分享知 识，还可以给用户一个网上交友的机会。本网站提供了登录注册，内容管理，用户管理，评论点赞，消息管理等功能。

2 需求分析

本系统的用户进入网站首先需要注册，填写基本信息，注册完成后去登录，登录成功后会浏览文章，并给自己喜欢的文章点赞或者收藏，如果特别喜欢某个用户可以关注该用户，用户除了浏览文章还可以发表文章，专门有一个界面去查看收到的所有消息。

本项目需要实现的功能有：

登录与注册。

用户访问网站内容。

用户发表文章。

用户可修改资料信息，关注或者取消关注用户。

用户给文章点赞评论收藏。

用户可查看自己收到的所有消息。

3 项目设计

3.1 项目前端设计

本项目采用了前后端分离式的形式。采用了 Vue.js 框架^[3]。为了提升方便样式设计，本项目引入了 Vant 与 Element 两个组件库。

本目前端的数据是通过 axios 向后端发送请求获取的，获取后的数据存储到状态管理库中。

3.2 项目后端设计

后端采用 node.js 运行环境。后端通过连接数据库来获取数据库中的数据，前端通 axios 从后端获取数据存入 Vuex 中，当需要改变数据库中数据时，前端向后端发送请求，后端通过从前端中得到的参数去数据库中查

找相应数据，并进行增删改查的操作。

3.3 数据库设计

数据存储使用 MySQL 数据库，本项目数据库有六张表，分别是：个人收藏表（collect），文章评论表（comment），个人关注表（concern_info），点赞表格（laud），文章内容表（message），用户信息表（user）。

其中用户数据表如表 1 所示：

名称	数据类型	长度	注释
user_id	INT	11	用户 ID
user_name	VARCHAR	50	用户名
user_password	VARCHAR	50	用户密码
user_headimg	VARCHAR	300	用户头像
message_id	VARCHAR	50	发表博客 ID
sex	VARCHAR	50	性别
age	INT	11	年龄

表 1 用户数据表

3.4 功能模块

本网站主要分为消息管理，评论管理，收藏管理，分类管理，用户管理，内容管理等六个功能模块。

消息管理模块：已登录用户可以在消息模块中查看，别人的评价，点赞与关注。未读的评论用小红点标记，当用户点击评论后，小红点消失，点赞与关注图标右上角有数字提醒用户点赞与关注人数，当用户查看之后，数字消失，消息界面有全部已读按钮，点击之后所有小红点消失。

评论管理模块：用户阅读文章之后可以在页面中评论，评论内容不能为空，同时用户可以在消息界面中看到别人的评论。

收藏管理模块：用户阅读文章之后可以选择收藏与取消收藏，同时用户可以在个人收藏夹界面中看到自己收藏过的文章。

分类管理模块：本项目把文章内容分为推荐，热榜，体育，知 识，娱乐，5 个部分，用户发表文章时候需要选择发表文章所属的分类，同时当用户浏览文章的时候也可以通过选择不同的分类来阅读相应类别的文章。

用户管理模块：访问本系统的用户分为访客和已登录用户，访客是指没有在本系统注册过的用户，访客进

入本系统之后, 只能浏览检索, 检索文章。

内容管理模块: 已登录用户发表文章上传到后端, 后端传回前端, 显示在首页。数据库存储一篇文章所具

有的 id, 标题, 文章内容

模块设计如图 1 所示:

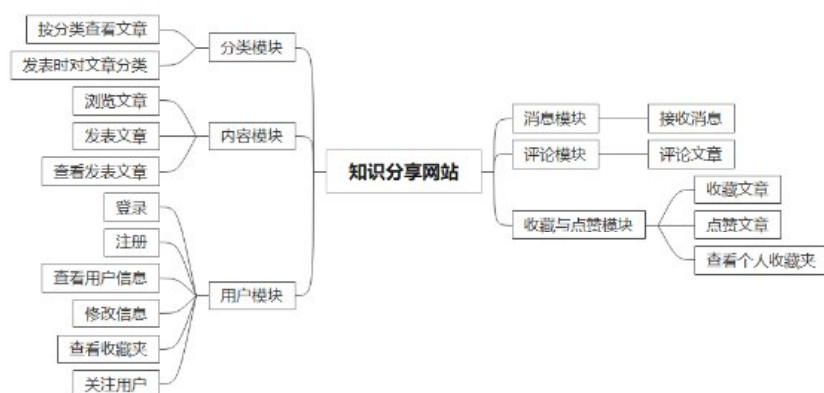


图 1 功能模块设计

3.5 模块实现

3.5.1 用户管理模块

用户管理模块是用来管理用户的个人数据（包括登录时的账号, 密码等）。

在用户登录界面中, 用户需要输入账号与密码并同意协议后登录, 账号必须是数字类型, 在数据库中保存为 user。

后端会维护一个用户表, 上面保存着已经登录的用户, 该表包括用户的 ID, 昵称, 密码, 头像, 已经发表文章的 ID, 性别, 年龄。注册中心需要上传个人头像, 输入昵称, 账号, 年龄, 选择性别并输入新密码与旧密码, 后端会验证旧密码是否输入正确, 若不正确, 显示提示信息, 要求用户重新输入。注册成功后跳转注册成功页面, 点击“返回登录页面”跳转回登录页面。

用户信息在用户注册时录入, 可在个人主页的个人消息中点击修改按钮, 跳转到编辑资料页面中修改, 可修改性别, 年龄, 密码, 昵称, 修改完成后点击确定按钮, 新的用户信息就会提交到后台, 数据库即可更新。

每当有用户登录后, 其数据都会被放入 oneperson 中, 标识此用户已经登录, 进入消息界面, 发表文章界面, 评论区域时, 通过 v-if 判断此用户是否在 oneperson 中, 来呈现不同的界面。

3.5.2 分类模块

分类模块用来管理文章的分类。

用户在首页点击想要查看的文章内容, 进入相应的文章界面。

用户发表文章时除了需要上传图片, 标题, 内容, 还需要选择分类, 文章分为五类, 推荐, 热榜, 体育, 知识, 娱乐。在数据库中保存为 message_type 字段。

用户发表文章时选择相应的类别, 通过 post 请求发送到后端, 在数据库中添加一条信息。

3.5.3 内容管理模块

内容管理模块用来管理用户发表的文章。

用户可通过关键词搜索文章, 通过相似度匹配出相应文章, 并按照时间顺序进行排序。输入框内容通过双向数据绑定, 首先判断是否为空, 然后判断此条记录是否保存于历史记录, 若未保存, 添加此记录到数据库中。

文章渲染到 v-for 中时, 可按用户需求, 文章按照默认排序与时间排序, 运用正则表达式, 遍历文章时把文章发表时间的年月日筛选出来, 使用 sort() 函数逐一进行对比。

浏览文章时, 当浏览内容超出页面时, 内容会随着滚动显示, 通 window.pageYoffset 方法, 设置当前页面相对的 Y 位置以及 window.addEventListener 的方法。在 window.pageYoffset 方法中, 当距离大于 100 的时候, 页面上方导航栏显示。导航栏中的返回箭头通过 this.\$router.back(-1) 的方法实现, 返回到首页的界面。页面刷新时, 从后台中获取文章作者的头像与昵称, 并显示到页面中。

整篇文章点赞数, 文章内容都是从后端直接获取的。点赞, 评论关注与收藏的显示隐藏样式都是用的 v-if。本设计前端统一使用 axios 采用 post 请求向后端发起请求来得到数据。得到数据后存入 Vuex 中。使用 beforeMount() 来获取数据然后放入 vuex 中, 再渲染界面。

3.5.4 评论管理模块

评论管理模块用来管理用户发表的评论

通过 v-for 把评论渲染到网站内容下的评论区区域, 用户评论后将数据发送到后端, 存入数据库中。

3.5.5 收藏管理模块

收藏管理模块是用来管理用户收藏的文章。

用户可在浏览文章时点赞与收藏, 在前端通过切换图片来标识用户是否点赞或收藏, 未点赞时候会显示目前该文章的点赞数。

后端用 post 请求, 查询数据库中文章的收藏状态, 返回 true 或者 false, true 代表已收藏, false 代表未收藏。用户收藏后再次点击可以取消收藏, 点击后, 后端会获取次用户 ID 与此文章 ID, 在数据库的收藏表中查询此条数据删除, 发生错误会返回“收藏表删除错误”。用户可在收藏夹查看自己目前所有已经收藏过的文章。

3.5.6 消息管理模块

消息管理模块用来管理用户接收到的消息。

每当有用户登录, 该用户 ID 会被放入状态管理库中的 oneperson 中, 消息界面渲染前会判断此用户 ID 是否

存在 oneperson 中, 未登录的用户在消息界面会看到提醒登陆的提示。

用户登录后点击消息页面可查看未读消息, 包括, 评论, 点赞, 收藏。通过 axios 方式从数据库中获取评论数据, 点赞数据, 收藏数据存入 vue 的状态管理库中, 并统计每部分数据的长度存入状态管理器中。

评论直接显示在消息界面, 用 v-for 循环遍历数据库中评论内容, 每个 li 标签中除了有评论信息还有一个 vant 组件, 该组件加入 dot 属性后可显示为一个小红点, 标识未读, 用户点击任意评论, 进入相应的网站内容, 浏览该网站后点击退出可返回到消息界面, 通过 to_message() 方法, 传入点击相应 li 标签的索引号与访问网站 ID 号, 把索引传入后端 index() 方法中, 同时使用路由的 push 方法跳转到该文章 ID 号对应的文章中, push() 方法中的参数可以是字符串或者对象, 此方法的作用是在跳转页面的同时把事件添加到 history 记录中。

Index() 会把传入的索引号加入已读文章队列中。用户每次进入消息界面, 前端都会通过 v-if 判断此条评论的索引是否在已读队列中, 若存在, 则隐藏图标。用

户回到消息界面后, 已经查看的评论对应的小红点会消失。

4 结束语

本文选取 Vue.js + Node.js + MySQL 技术, 根据用户网上互动与知识获取为出发点, 设计并实现了一个知识分享的网站。经过多次测试, 本网站可以满足用户的知识分享与交友的需求。本网站注册时密码没有设置相应的安全系数, 后续会在注册模块增加用户密码安全系数的设置, 禁止用户设置太过简单的密码来提高账号的安全性。

【参考文献】

- [1] 涂远杰, 郑剑. 基于 Flask 的博客网站设计与实现 [J]. 电脑知识与技术, 2020, No. 76(02):210.
- [2] 邓笑. 基于 Spring Boot 的校园轻博客系统的设计与实现 [D]. 华中科技大学, 2018.