图片包含 徽标

AI 生成的内容可能不正确。

**本科毕业论文**

**基于web的跨专业互助学习系统**

**学 院:** 数学与信息学院

**专 业:**  计算机科学与技术

**姓 名:** 黄满德

**学 号:** 202129210105

**指导教师:**  姚金涛  **职称** 副教授

**提交日期：** 2025 **年**  4 **月** 14 **日**

华南农业大学本科毕业论文（设计）原创性声明

本人郑重声明：所呈交的毕业论文（设计），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名： 日期： 年 月 日

华南农业大学本科毕业论文（设计）使用授权声明

本人完全了解学校有关保留、使用毕业论文（设计）的规定，同意学校保留并向国家有关部门或机构送交毕业论文（设计）的复印件和电子版，允许毕业论文（设计）被查阅和借阅。学校可以将本毕业论文（设计）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编毕业论文（设计）。

作者签名： 指导教师签名：

日期： 年 月 日 日期： 年 月 日

摘 要

当今高等教育中专业划分细致易造成知识局限，基于此提出开发基于 web 的大学生跨专业互助学习系统。其目的一是打破传统专业壁垒，让学生接触其他专业核心内容、前沿动态与实践案例以激发创新思维，二是为学生学习难题提供及时帮助，且激发学习兴趣。从意义方面来看，对于学生，能拓宽知识视野，领略各学科魅力价值；从教育角度，可分享整合多专业课程资源、学习资料与经验以优化教育资源，同时使用者还能借助平台咨询、解答问题，促进不同专业领域的交流学习。

本系统的前后端基础框架分别为Gin和Vue3，使用MySQL数据库存储基本数据，同时使用Websocket 构建了简单的聊天信息平台，使用MongoDb存储用户聊天信息；同时系统接入豆包语言大模型，给用户提供了和豆包进行对话的接口。系统中包含有文章、视频、热点管理，互助聊天、AI总结问答等基础功能。用户可以点赞、收藏、评论文章，并和文章作者聊天学习。系统通过多样化的互动交流和学习方式激发学习兴趣，让学习过程不再枯燥，提高学习的积极性和主动性，能促进不同专业领域学生间的交流学习，营造良好跨专业学习氛围。

关键词：Gin+Vue；跨专业学习平台；聊天学习网站；AI问答

**Abstract**

The detailed division of majors in today's higher education can easily lead to knowledge limitations. Based on this, it is proposed to develop a web-based cross-professional mutual aid learning system for college students. The purpose is to break down the barriers of traditional majors and allow students to contact the core content, cutting-edge dynamics and practical cases of other majors to stimulate innovative thinking. The second is to provide timely help for students to learn problems and stimulate their interest in learning. From a meaningful perspective, for students, it can broaden their knowledge horizons and appreciate the charm and value of various disciplines. From an educational perspective, they can share and integrate multi-professional course resources, learning materials and experiences to optimize educational resources. At the same time, users can also use the platform to consult and answer questions, and promote exchange and learning in different professional fields.

The front-end and back-end basic frameworks of this system are Gin and Vue3 respectively. The MySQL database is used to store basic data. At the same time, a simple chat information platform is built using Websocket, and MongoDb is used to store user chat information. At the same time, the system is connected to the large model of Doubao language, providing users with an interface for dialogue with Doubao. The system contains basic functions such as articles, videos, hotspot management, mutual aid chat, AI summary Q & A. Users can like, collect, comment on articles, and chat with article authors to learn. The system stimulates interest in learning through diverse interactive communication and learning methods, makes the learning process no longer boring, improves the enthusiasm and initiative of learning, and can promote the exchange and learning between students in different professional fields, creating a good inter-professional learning atmosphere.·

Key words: Gin + Vue; cross-professional learning platform; chat learning website; AI

[1 绪论 5](#_Toc194960993)

[1.1 课题来源及意义 5](#_Toc194960994)

[1.1.1 课题的来源 5](#_Toc194960995)

[1.1.2 课题研究的意义 5](#_Toc194960996)

[1.2 主要研究的目的及内容 5](#_Toc194960997)

[1.3 研究方法及设计思路 6](#_Toc194960998)

[1.3.1 研究方法 6](#_Toc194960999)

[1.3.2 设计思路 6](#_Toc194961000)

[2 相关技术简介 8](#_Toc194961001)

[2.1 前端技术简介 8](#_Toc194961002)

[2.2 后端技术简介 9](#_Toc194961003)

[2.3 数据库简介 10](#_Toc194961004)

[3 系统需求分析 12](#_Toc194961005)

[3.1 系统可行性分析 12](#_Toc194961006)

[3.1.1 技术可行性 12](#_Toc194961007)

[3.1.3 运行可行性 13](#_Toc194961008)

[3.1.4 时间可行性 13](#_Toc194961009)

[3.2 系统总体需求 13](#_Toc194961010)

[3.3 系统功能需求 15](#_Toc194961011)

[3.3.1 账户模块 15](#_Toc194961012)

[3.3.2 文章模块 15](#_Toc194961013)

[3.3.3 互助帮学模块 16](#_Toc194961014)

[3.3.4 用户交流模块 16](#_Toc194961015)

[3.3.5 后台管理 16](#_Toc194961016)

[3.4 本章小结 17](#_Toc194961017)

[4 整体系统设计 18](#_Toc194961018)

[4.1 系统总体设计 18](#_Toc194961019)

[4.1.1 web系统设计准则 18](#_Toc194961020)

[4.1.2 系统总体结构 18](#_Toc194961021)

[4.2 具体功能模块设计 20](#_Toc194961022)

[4.2.1 用户层设计 20](#_Toc194961023)

[4.2.2 文章管理设计 21](#_Toc194961024)

[4.2.3 互助帮学设计 22](#_Toc194961025)

[4.2.4 沟通模块设计 22](#_Toc194961026)

[4.2.5 后台模块设计 23](#_Toc194961027)

[4.3 数据库设计 23](#_Toc194961028)

[4.3.1 整体数据结构设计 23](#_Toc194961029)

[4.3.2 数据库物理结构设计 23](#_Toc194961030)

[4.4 本章小结 26](#_Toc194961031)

[5 系统实现 26](#_Toc194961032)

[5.1 开发工具及运行环境介绍 27](#_Toc194961033)

[5.2 用户注册登录模块实现 27](#_Toc194961034)

[5.3 文章管理模块实现 29](#_Toc194961035)

[5.3.1 文章创建编辑 29](#_Toc194961036)

[5.3.2 文章浏览 30](#_Toc194961037)

[5.4 互助功能模块实现 31](#_Toc194961038)

[5.5 交流功能模块实现 31](#_Toc194961039)

[5.6 后台管理模块实现 32](#_Toc194961040)

[5.7 系统表单组件实现 32](#_Toc194961041)

[5.8 本章小结 32](#_Toc194961042)

1 绪论

本章主要介绍本课题研究的来源及意义，国内外研究现状，主要研究的目的及内容，以及研究的方法和设计思路。

1.1 课题来源及意义

1.1.1 课题的来源

在如今大学教育环境下，随着互联网技术的蓬勃发展以及知识经济时代的全面来临，信息的传播与获取变得愈发便捷，为学生突破专业局限、进行跨专业学习创造了前所未有的良好条件。跨专业学习作为拓宽个人知识视野、提升综合素养以及增强职场竞争力的重要途径，却面临着诸多现实阻碍，陷入传统学习资源分散与高效整合需求迫切的两难境地。因为高等教育专业划分过于细致，学生在学习过程中容易出现知识的局限性，会出现因难以找到系统全面且贴合自身水平的学习资料，导致学习计划难以顺利推进，而现有的各类学习平台大多存在课程关联度低，缺乏个性化学习引导，不同专业知识融合性差以及学习效果评估不够精准等问题。

本课题正是基于上述背景应运而生，以打造综合性跨专业学习网站为切入点，旨在攻克现有学习平台存在的诸多问题，充分利用前沿的互联网技术，构建起一个资源丰富、易于使用、学校模态多样的跨专业学习网站。

1.1.2 课题研究的意义

在高等教育专业划分细致的背景下，学生往往局限于自身所学专业知识范畴内。此跨专业互助学习系统，能让学生接触到其他专业同学分享的学习经验、学习方案、实践案例等，使其不再受限于单一专业知识体系，极大地拓宽了知识视野，帮助学生构建更全面、多元的知识架构。同时，系统提供多样化的互动交流学习方式，例如互助聊天，与文章作者交流学习，通过AI总结获取知识等等，这一系列设计有助于调动学生学习的积极性和主动性，使其更主动的投入到学习中。

1.2 主要研究的目的及内容

本课题的主要研究目的，旨在通过开发基于 web 的大学生跨专业互助学习系统，一方面帮助学生突破传统专业壁垒，使其接触其他专业核心内容、前沿动态与实践案例以激发创新思维，另一方面为学生学习中遇到的难题提供及时帮助并激发学习兴趣，进而促进不同专业领域学生间的交流学习，营造良好的跨专业学习氛围。在研究内容方面，涵盖系统架构搭建，前端选用 Vue3框架、后端采用 Gin框架，使用MySQL 数据库存储基本数据，借助 Websocket 构建聊天信息平台，使用 MongoDb 存储用户聊天信息，同时接入豆包语言大模型提供对话接口；在功能模块开发上，着重打造文章、视频、热点管理，互助聊天、AI 总结问答等基础功能，支持用户点赞、收藏、评论文章及与作者聊天学习等互动功能；通过多样化互动交流和学习方式，激发学生兴趣，提高其积极性和主动性，最终实现提升跨专业学习效果与促进专业领域交流融合的目标。

1.3 研究方法及设计思路

1.3.1 研究方法

通过广泛查阅国内外有关跨专业学习、在线学习平台等多方面文献资料，为系统设计开发提供理论支撑，确保研究科学合理。接着运用调查研究法，通过设计调查问卷并实地访谈高校学生群体，了解学生跨专业学习实际需求等情况，为功能模块设计找准现实依据。同时利用案例分析法，了解学习的在线学习平台案例，借鉴其优势经验融入到本系统开发中，提升系统质量。

1.3.2 设计思路

**（1）用户需求导向**：基于前期通过调查研究法所获取的大学生跨专业学习的实际需求，明确系统要解决的关键问题，如打破专业壁垒、提供个性化学习引导、优化互动交流体验以及方便学习资源获取等。以满足这些核心需求为出发点，来规划系统整体的功能框架和操作流程，确保系统能够切实贴合学生的使用场景，提高其对系统的认可度和使用率。

**（2）功能架构规划**：从功能层面进行系统架构设计，将其划分为多个功能模块，包括文章、视频、热点管理模块，用于整合和展示多专业的优质学习资源；互助聊天模块，方便不同专业的学生实时交流互动、答疑解惑；AI 总结问答模块，借助智能技术辅助学生快速获取知识要点、解答学习疑问等。同时，设计用户的点赞、收藏、评论等互动功能，增强用户参与感与学习趣味性，通过各功能模块之间的协同配合，构建起一个功能完备、操作便捷的跨专业学习平台。

**（3）技术实现保障**：根据系统功能需求，选择合适的技术框架来确保系统的稳定运行与高效实现。前端采用 Vue3 框架，注重打造简洁、美观且易用的用户界面，提升用户交互体验；后端运用 Gin 框架，保障系统的高效处理能力和良好的扩展性。利用 MySQL 数据库存储如用户信息、学习资源等基本数据，借助 Websocket 构建实时通信的聊天信息平台，采用 MongoDb 存储海量的用户聊天信息，并且接入豆包语言大模型，为用户提供智能对话接口，运用这些技术手段共同支撑系统各项功能的顺利实现，为学生提供优质的跨专业学习服务。

2 相关技术简介

本章主要介绍实现跨专业互助学习系统所使用到前后端技术栈，包括了Vue、Vite、Gin等等。

2.1 前端技术简介

在前端的技术选型上，整体使用Vue 3+Vite作为前端页面开发的工具。Vue 3 是渐进式 JavaScript 框架 Vue.js 的最新主要版本，而 Vite 是一个基于原生 ES 模块提供支持的构建工具。

Vue 3 使用新的响应式系统，摒弃Vue 2 中 Object.defineProperty 的实现方式，转而采用 Proxy 来构建响应式机制。在复杂的数据结构频繁更新的场景下，Vue 3 的响应式系统能够更迅速地响应变化并更新相应的 DOM 元素，有效提升了应用的响应速度和流畅度。在学习曲线上，Vue 3 的语法和 API 设计相对简洁直观。Vue 3 在组件的定义和使用上，采用了类似 HTML 标签的方式来挂载组件，代码结构清晰，易于阅读。而 React 则依赖于 JavaScript 的函数式编程概念，像 JSX（JavaScript XML）语法，需要开发者对 JavaScript 函数、返回值等有更深入的理解，对于新手来说可能存在一定的学习门槛。Vue 3 的官方文档非常详细且条理清晰，示例丰富，从基础概念到高级应用都有详细的讲解和代码示例，方便开发者快速查找和学习相关知识。

图示

AI 生成的内容可能不正确。

图1 Vue原理

Vite是一个超快的前端构建工具。在传统的前端开发构建工具中，启动开发服务器往往需要对整个项目进行打包等预处理操作，这一过程随着项目规模的增大，耗时会显著增加。而 Vite 利用了现代浏览器原生支持 ES 模块的特性，在开发环境下省略了繁琐的打包过程，直接基于源文件启动开发服务器，从而实现了极快的冷启动速度，大大缩短了开发者等待项目启动的时间，提高了开发效率，让开发者能够迅速进入开发调试环节。Vite在小型项目中的热更新表现相当出色，当项目中的某个文件发生变化时，它能够精准地识别出变化的模块，并仅对该模块进行重新编译和更新，而无需像传统构建工具那样重新构建整个应用。Vite 采用按需编译的策略来优化项目的构建过程。不同于传统构建工具会一次性对所有模块进行编译处理，Vite 只会在浏览器实际请求某个模块时，才对其进行编译并提供给浏览器。这一策略避免了对那些暂时未被使用到的模块进行不必要的编译工作，有效减少了编译时间和资源消耗，特别是对于大型项目，按需编译能够显著提升整体的开发和构建效率。

图示

描述已自动生成

图2 Vite 原理

2.2 后端技术简介

Gin框架是一个用 Go 语言编写的轻量级 Web 框架，由 Golang 开发者实现。它的设计目标是高效、快速且易于使用。Gin 基于 Radix 树实现路由，这种数据结构使得路由查找速度极快，能够高效处理大量的并发请求。并且它的中间件机制设计合理，开销小，有助于提升整体性能。此外，Gin提供了简洁明了的 API，开发者可以轻松上手。它采用了类似 Express.js 的路由定义方式，让熟悉 Node.js 的开发者也能快速适应。同时，Gin支持多种中间件，如日志记录、错误处理、跨域处理等。开发者可以根据需求自定义中间件，增强应用的功能。Gin 还内置了对 JSON 数据的验证功能，方便开发者对请求数据进行验证和处理。

图示

描述已自动生成

图3 Gin框架结构

2.3 数据库简介

在项目中一共使用了三种数据库Mysql、MongoDB、Redis。

MySQL 是一款开源的关系型数据库管理系统，具备 ACID（原子性、一致性、隔离性、持久性）特性，能确保在复杂的事务操作中数据的准确性和完整性，适用于对数据完整性、准确性要求较高的各类业务场景。

MongoDB 是一款开源的非关系型数据库，采用文档型数据存储方式，旨在为开发者提供灵活、可扩展的数据存储解决方案。文档结构可以动态变化，同一集合（类似关系型数据库中的表概念）中的文档可以有不同的字段，这对于一些数据结构多变或者需求不断演进的应用场景非常友好。MongoDB对读写操作进行了优化，尤其是在处理大量非结构化、半结构化数据时，能够快速地写入和查询数据，在某些场景下读写性能优于传统关系型数据库。

Redis是一款开源的高性能键值对（Key-Value）存储数据库，常用于缓存、消息队列、实时统计等场景，它的数据存储在内存中，因此读写速度极快。数据存于内存，使得其读写操作的响应时间极短，能够快速地获取或更新数据，非常适合作为缓存层来减轻后端数据库的压力，提高应用整体性能。

3 系统需求分析

3.1 系统可行性分析

3.1.1 技术可行性

本跨专业互助学习系统选用 vue3、Gin 框架、websocket、mysql、mongoDB 等技术栈。

在前端开发中我们使用Vue3，其采用组件化的开发模式，使得代码结构清晰明了，具备很强的可复用性。同时，基于响应式原理，能够打造出交互体验良好、流畅度颇高的用户界面，为学习者提供便捷、舒适的操作环境，十分契合系统前端展示以及交互操作的功能需求。而且 vue3 在前端开发领域已经占据了重要地位，一直在不断更新迭代，性能和稳定性都有可靠保障，也经过了大量项目实践的检验，技术成熟度高，遇到常见问题很容易在相关技术社区找到解决方案。

Gin框架在后端开发中方面发挥关键作用，它是一个轻量级的 Web 框架，拥有高效的路由处理机制以及中间件支持功能，能够快速搭建起稳定可靠的服务端，轻松应对系统运行过程中的大量并发请求，保障系统后端服务的稳定运行，有力地支撑起系统各项业务逻辑的实现，在众多 Web 开发项目中被广泛应用，有着成熟的技术生态和丰富的开发案例可供参考。

Websocket术专注于实现实时通信功能，对于跨专业互助学习系统来说意义重大。它能够让不同专业的学习者在系统中进行实时的交流互动，比如在线讨论问题、实时答疑解惑等，极大地增强了系统的交互性和实时性，并且其技术规范成熟，在各类需要实时通信的应用场景中都有着出色的表现。

mysql 具备强大的数据存储和管理能力，能够高效、稳定地存储系统中的用户信息、学习资料数据、交流记录等各类结构化数据，且有着完善的事务处理机制，保证数据的一致性和完整性，同时有着丰富的工具和插件支持，方便开发过程中的数据库操作与维护。

MongoDB是非关系型数据库，擅长处理一些非结构化或半结构化的数据，例如学习者上传的多样化学习资料、个性化的学习笔记等，与 mysql 相辅相成，共同满足系统复杂的数据存储需求，其具备灵活的数据模型和良好的扩展性，在大数据量存储和高并发读写场景下表现优异，在现代软件开发中应用广泛。

在硬件方面，常规的服务器设备就能满足系统开发以及初期运行的基本需求，随着系统用户数量的逐步增加以及功能拓展，可以依据实际情况灵活地进行硬件资源的扩展升级。开发过程中使用的集成开发环境为 Visual Studio Code ，能提供丰富多样的功能以及实用的插件。在AI辅助开发功能日益发展的今天，搭配上Copilot，MarsCode插件后，可显著提高开发效率。

3.1.3 运行可行性

从使用习惯上看，系统使用简洁直观、易于理解的页面设计，类似常见社交平台或者学习类APP的操作模式，便于使用者快速上手该系统。从人力配置来看，在系统运营初期，只需要少量的维护人员即可完成系统的使用，人力需求相对容易满足，运行可行性高。从成本上看，所用技术和工具均为开源免费的，成本主要集中在服务器的租赁和大模型接口的调用Token费用，后期运营维护成本在可控范围内。综上所述，本系统运行可行性高，值得开发。

3.1.4 时间可行性

从时间上看，系统开发花费的时间人力成本大概为90pd。在已经具备基本的技术栈背景下，在前两周，会进行详细的需求分析，明确系统的功能架构和模块区分。在后续的一个半月，可以根据前期设计，展开编码工作，优先实现核心功能模块，同时有序推进各个模块的开发。在最后的时间，进行系统的集成测试，漏洞修复，使得系统达到预期要求。

3.2 系统总体需求

通过跨专业互助学习系统，可以拓宽学生的视野，让学习了解各个学科中的独特魅力和价值。在功能模块上，系统包含用户管理、文章增删查改、文章收藏、点赞、评论、聊天等等功能。

用户管理：系统支持通过邮箱验证注册的方式，提供账号密码找回功能，通过邮箱验证重置密码，未来也会支持通过社交账号的登录功能。

文章管理：登录后的用户可以通过文章创作中心进行文章的创建发布管理，在管理员审核后，文章即可被浏览、点赞、收藏、转发、评论，同时我们提供了文章访问、评论、点赞量等信息统计。

请求帮学模块：在请求帮学模块，用户可以发布请求帮学信息内容，同时提供标签选择，便于其他用户快速定位并响应请求。用户可查看自己发布的帮学请求状态，如待解答、已解答、已关闭等。

用户帮学模块：系统提供了AI聊天机器人，以及用户聊天机器人功能，同时也提供了在线视频通话功能，方便用户在学习上的交流沟通。也支持文件的上传和下载，方便用户分享作业、课件、学习资料等等。

信息审核模块：系统存在管理员用户角色，采用关键词过滤、帮助管理员管理评论、文章、帮学模块的信息审核。

图示

AI 生成的内容可能不正确。

图4 用户用例图

图示

AI 生成的内容可能不正确。

图5 管理员用例图

3.3 系统功能需求

3.3.1 账户模块

账户模块主要负责用户注册、登录、退出、密码找回等操作。

用户没有登录时，为游客状态，只能浏览文章和用户发布的问题，不能对文章进行任何操作，包括点赞、收藏等等。在注册界面用户填写完成用户名和密码即可正常登录系统。

在登录注册界面，提供重置密码功能，可以通过邮箱验证，完成对密码的重置。

登录后，用户点击右上角自己的头像时，会出现退出选项，点击退出按钮就可以退出系统。

3.3.2 文章模块

文章模块是系统的核心功能之一，登录后的用户可以创建文章，并给文章打上标签、话题、摘要等内容，点击发布文章，待管理员审核通过后即可在主页浏览到对应文章。每篇文章都可以根据AI大模型生成一段概述，这段概述在浏览文章时候可以看到。

用户可以通过关键字搜索文章，文章可以进行点赞、评论、收藏、转发、沉浸式阅读。浏览文章的同时，也会展示了文章作者的相关信息，通过点击按钮作者和行私聊。

3.3.3 互助帮学模块

在请求帮学模块中，用户可以发布一个简单的帖子，内容可以是提问问题、资料查询等等、学习路线咨询等等。同时，使用者也可以上传资料，审核过后即供给其他用户下载。

3.3.4 用户交流模块

在用户交流模块中，系统提供了私信的功能，在对方未回复你之前只允许发送一条私信内容，双方回复之后既可开启畅聊，提供了表情图片等信息的发送功能。同时系统也提供了视频聊天的功能，通过发起视频聊天的按钮向在线用户进行视频聊天。

3.3.5 后台管理

后台管理模块对网站的平稳运行、学习资源质量保障以及用户使用体验的提升起着不可或缺的关键作用。

在账户管理方面，管理员能够全面查看用户的注册信息，涵盖基本资料、且负责对用户账号实施全生命周期管理。比如，对于违反网站使用规则、存在恶意行为或者长期未活跃的账号，管理员有权进行冻结、删除等相应处理，以此来确保网站用户群体的规范性以及整体环境的安全性。

文章管理功能上，管理员承担着重要职责。在文章发布之前，会对各类专业性文章、学习心得分享等内容的准确性、科学性以及是否符合网站定位和规范进行严格细致的审核，确保每一篇发布出来的文章都能够为不同专业的用户提供有价值、可靠的跨专业学习知识与经验借鉴。文章会按照不同的专业领域、知识主题等进行合理分类展示，方便用户根据自身的学习需求快速检索查找。而在文章发布之后，一旦发现文章存在知识错误、观点过时或者不符合网站当前发展方向等情况，管理员可以及时对其进行编辑修改，若问题较为严重无法有效修正的，则果断进行下架或者删除处理，始终保持网站学习文章资源的高质量。

评论管理方面，管理员在该模块中发挥着关键作用。负责对用户在文章评论区、学习交流讨论区等各个互动板块所发布的内容进行审核管理。一方面，严格审查评论内容，一旦发现存在违规言论、恶意诋毁、低俗或者与学习主题完全无关的评论，会立即进行删除处理，全力维护网站积极健康、专注于跨专业学习交流的良好互动氛围。另一方面，管理员还会关注那些具有建设性、能引发深度思考和热烈讨论的优质评论，对于这类评论可进行置顶、推荐等操作，以鼓励更多用户积极参与到高质量的学习互动中来，进一步提升网站的活跃度和用户体验。

公告内容的配置功能也是后台管理模块的重要组成部分。管理员可在此功能模块中进行各类公告信息的编辑、发布与管理操作。例如，当网站有新的学习活动推出、专业板块更新或者重要的使用规则调整时，管理员能够及时撰写相应的公告内容，可灵活设置公告的标题、正文、发布时间以及展示有效期等关键要素，确保公告信息准确传达给用户。同时，对于已发布的公告，管理员有权根据实际情况进行修改、删除或者延长展示期限等操作，方便根据网站运营的不同阶段以及各类突发情况，精准地向用户传递重要资讯。

3.4 本章小结

本章对跨专业互助学习系统进行了细致的需求分析一是可行性分析，从技术、运行、时间三方面说明，所选技术成熟可靠，运行易上手且成本可控，开发时间人力成本有相应安排。二是总体需求，介绍系统功能模块及各模块大致功能，含用户、文章、帮学、交流、审核等方面，配有对应用例图。三是系统功能需求，细分账户、文章、互助帮学、用户交流、后台管理等模块，阐述各模块具体操作及管理员相应职责。

4 整体系统设计

本章主要介绍在设计跨专业互助学习系统时采用的设计思想和细节技术栈，同时阐述了选择该项技术的主要原因。根据网络上的最佳实践，从系统总体结构分析到各个功能的详细分析，细致的描述了一遍整个系统的技术路线。

4.1 系统总体设计

4.1.1 web系统设计准则

作为一个基于web的项目系统，应该满足易用性的要求，同时具备拓展性、兼容性，在性能方面也要达到反馈速度迅速且稳定，在代码方面，也需要有良好的维护性，以保证后续的维护。总的来说，整个系统的设计应该遵循以下的原则：

1. 易用性：界面的设计要符合用户操作习惯，操作流程简洁明了，方便不同技术水平的用户轻松上手，降低用户的学习成本。
2. 可维护性：代码结构清晰、易于理解和修改，方便开发团队后续进行功能优化、问题修复等维护工作，遵循良好的编程规范和设计模式。
3. 安全性：对用户的隐私数据（如账号密码、个人信息、交易记录等）进行加密存储和传输，防止数据泄露。具备防范常见网络攻击（如 SQL 注入、跨站脚本攻击等）的能力。

4.1.2 系统总体结构

系统基于前后端分离的设计理念，采用功能化分层的设计思想，将系统划分为，表现层、数据缓存层、服务层、数据层，具备良好的维护性和可拓展性。系统整体结构如下图6所示。

表现层作为用户与系统接触的第一道 “桥梁”，是整个系统直观呈现在用户眼前的部分，也是用户与之产生交互的首要界面。表现层依托于Vue3框架构建，使用组件化，结构化的思想，对页面总体设计成上中下三栏布局，分别对应导航区域，内容区域，底部信息透出区域。使用Vue-Router实现内容区域流畅切换，使用Pinia作为全局的状态管理，封装了通用的请求响应拦截器，实现API的统一管理，以及更多的状态同步功能，以此应对复杂的交互场景。针对拓展性原则，使用了Formily表单库，封装了表单系统，对表单的生命周期进行全面的封装，包括表单初始化、回填、联动、校验等内容，通过一份JsonSchema即可渲染表单，提升表单的开发效率。

数据缓存层使用ServiceWorker、Redis技术对请求内容做了一个统一的缓存处理，根据浏览器强化缓存和协商缓存思想，用户首次请求将数据缓存在该层中，待下次发起相同请求的时候从缓存中获取数据，无需再从数据库中查询，优化了性能的同时也减少了服务端的压力。

服务层整体采用Go语言编写，使用Gin框架，将各个功能模块拆分，使用Gin框架路由，基于功能模块做分组，接收客户端请求。同时使用令牌桶算法，限制了客户端请求数量。在服务端设计上，整体封层为Controller、Logic、Dao三层经典架构设计，这种分层架构设计使得服务层的代码结构清晰，各层职责明确，便于团队协作开发和系统的长期维护。同时，结合功能模块拆分和请求限流机制，为系统的高性能、高可用性和可扩展性提供了有力保障。

在数据层中，服务层借助 Gorm 这一强大的 ORM（对象关系映射）库与关系型数据库展开交互。Gorm 能把数据库中的表、行、列等概念，自然地映射为 Go 语言中的结构体、对象和字段，极大地简化了数据库操作的代码编写工作。在系统架构中，除了使用关系型数据库与 Gorm 交互，还引入 MongoDB 这一流行的非关系型数据库，可以进一步丰富数据存储和处理能力。在实际应用中，可以根据数据的特点和业务需求，合理地将数据存储在关系型数据库和 MongoDB 中。

图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。

图6 系统架构图

4.2 具体功能模块设计

4.2.1 用户层设计

在跨专业互助学习系统前端页面中，用户存在有三种状态，未登录态、登录态、管理员。三种状态分别对应有不同的功能。对于需要用户信息的功能，例如创建文章，点赞、评论、视频、提问等都需要用户进行登录注册之后才可以进行操作。

对于用户操作时那些需要用户信息的操作时，会提醒用户进行登录注册。对已登录用户，则无限制。

登录页面中，提供了通过第三方验证信息进行登录的功能，同时也提供了找回密码功能，通过邮件信息获取验证信息，重置密码。

图示

AI 生成的内容可能不正确。

图7 用户层设计用例图

4.2.2 文章管理设计

在文章管理中，用户在文章创建界面对文章进行编辑创作，在发布文章前需要填写文章分类信息、文章话题、文章封面、摘要等信息，发布后，会自动调用大模型接口，获取整篇文章的AI总结信息。待系统管理员审核通过后，才可以被用户浏览、点赞、评论等等。

图示

AI 生成的内容可能不正确。

图8 文章管理设计

4.2.3 互助帮学设计

在互助帮学模块中，用户登录后，即可在互助帮学界面发送帖子，在经过管理员审核之后便可在互助模块中被其他用户浏览、评论。通过点击用户头像，也可以跳转到聊天界面，进行沟通交流。

图示

AI 生成的内容可能不正确。

图9 互助模块设计

4.2.4 沟通模块设计

在沟通模块中，用户可以通过评论、文章、互助信息的用户简介跳转到聊天信息页面。进入聊天信息界面后发起websocket请求，与用户进行一对一沟通，聊天框支持文字、表情、图片的发送。

4.2.5 后台模块设计

在后台管理模块中，有用户管理，文章管理，互助帮学管理，公告配置四大功能。在三个管理页面中，主要由上方的筛选框和下方的表格组成，通过各种筛选配置，进行数据的筛选，映射在表格中。表格中有操作列，可以进行编辑、删除两项操作，点击编辑会弹出编辑弹窗表单，供给管理员进行编辑，点击删除会进行二次询问，确定后进行删除操作。

4.3 数据库设计

4.3.1 整体数据结构设计

本跨专业互助学习系统的数据结构设计，通过多个数据表以及严谨的外键关联关系，充分考虑了用户、内容（文章、互助学习资料）、分类、标签、话题以及用户互动等多方面的需求，实现了跨专业的知识分享、互助学习以及用户之间的交流互动等功，各个表的字段设计和约束设置也有助于提高数据的准确性、完整性以及查询效率等。整体ER图如下所示。

4.3.2 数据库物理结构设计

本系统涉及的表结构有用户信息表、文章信息表、文章评论表、文章分类表、点赞表、收藏表、评论表、互助信息表等，下面是数据库主要数据表信息。

表1 用户信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 默认值 | 字段描述 |
| user\_id | varchar(20) | 0 | 用户ID |
| user\_name | varchar(20) |  | 用户名 |
| nickname | varchar(20) |  | 昵称 |
| overview | varchar(255) |  | 简介 |
| email | varchar(320) |  | 邮箱 |
| phone\_number | varchar(11) |  | 手机号 |
| password | varchar(255) |  | 密码 |
| confirm\_password | varchar(255) |  | 重复密码 |
| gender | tinyint | 0 | 性别 0:未知 1:男 2:女 |
| is\_email\_verified | tinyint(1) | 0 | 邮箱是否已验证 1为验证 |
| third\_party\_accounts | Json | 0 | 第三方账号信息 1为验证 |
| avatar | varchar(255) |  | 头像信息 |
| cover | varchar(50) |  | 封面图片信息 |

续表1 用户信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 默认值 | 字段描述 |
| following\_question\_ | int unsigned | 0 | 我关注的问题数 |
| notification\_unread | int unsigned | 0 | 未读通知数 |
| inbox\_unread | int unsigned | 0 | 未读私信数 |
| headline | varchar(40) |  | 一句话介绍 |
| bio | varchar(160) |  | 个人简介 |
| company | varchar(255) |  | 公司名称 |
| location | varchar(255) |  | 地址 |
| created\_at | datetime | currenttime | 注册时间 |
| updated\_at | datetime | currenttime | 更新时间 |

表2 文章信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 默认值 | 字段描述 |
| article\_index | int unsigned |  | 文章序号 |
| article\_id | varchar(20) |  | 文章ID |
| title | varchar(255) |  | 文章标题 |
| content | text |  | 文章内容 |
| user\_id | varchar(20) |  | 文章作者 |
| category\_id | int |  | 分类id |
| tag\_id | int |  | 标签id |
| abstract | varchar(200) |  | 文章摘要 |
| cover | varchar(200) |  | 文章封面 |
| gpt\_summarize | varchar(200) |  | gpt生成的文章摘要 |
| view\_count | int | 0 | 浏览量 |
| created\_at | datetime | currenttime | 创建时间 |
| updated\_at | datetime | currenttime | 更新时间 |
| article\_status | int | 0 | 文章状态0未审核1审核过 2未通过 |

表3 文章评论表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 默认值 | 字段描述 |
| comment\_id | varchar(20) |  | 评论ID |
| article\_id | varchar(20) |  | 文章ID |
| user\_id | varchar(20) |  | 用户ID |
| content | text |  | 评论内容 |
| create\_time | datetime | CURRENT\_TIMESTAMP | 创建时间 |
| update\_time | datetime | CURRENT\_TIMESTAMP | 更新时间 |
| parent\_comment\_id | varchar(20) |  | 父级评论 |

表4 文章分类表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 字段名 | | 字段类型 | | 默认值 | 字段描述 | | |
| category\_id | | int | |  | 分类id | | |
| category\_name | | varchar(100) | |  | 分类名称 | | |
| parent\_id | | int | |  | 父亲分类 | | |
| create\_time | datetime | | CURRENT\_TIMESTAMP | | | 创建时间 |
| update\_time | datetime | | CURRENT\_TIMESTAMP | | | 更新时间 |

表5 文章点赞表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 默认值 | 字段描述 |
| like\_id | int |  | 点赞id |
| article\_id | varchar(20) |  | 文章id |
| user\_id | varchar(20) |  | 用户id |
| liked\_at | timestamp | CURRENT\_TIMESTAMP | 点赞时间 |

表6 互助信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 默认值 | 字段描述 |
| assistance\_index | int unsigned |  | 互助序号 |
| assistance\_id | varchar(20) |  | 互助ID |
| title | varchar(255) |  | 互助标题 |
| content | text |  | 互助内容 |

续6 互助信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名 | 字段类型 | 默认值 | 字段描述 |
| user\_id | varchar(20) |  | 用户ID |
| category\_id | int |  | 分类id |
| tag\_id | int |  | 标签id |
| topic\_id | int |  | 话题id |
| abstract | varchar(200) |  | 互助摘要 |
| cover | varchar(200) |  | 互助封面 |
| created\_at | datetime | CURRENT\_TIMESTAMP | 创建时间 |
| updated\_at | datetime | CURRENT\_TIMESTAMP | 更新时间 |
| assistance\_status | int | 0 |  |

4.4 本章小结

本章对跨专业互助学习系统设计进行了详细介绍。在系统总体设计方面，基于 web 项目特性明确设计准则，涵盖易用性、可维护性、安全性等原则。其总体结构采用前后端分离与功能化分层理念，表现层用 Vue3 等实现页面交互与布局；服务层以 Go 语言结合 Gin 框架保障高效协作；数据层通过 Gorm 与关系型数据库交互并引入 MongoDB 丰富存储。具体功能模块上，整体介绍了用户模块、文章模块、互助模块、沟通交流模块、后台管理模块的整体设计，同时也通过表结构将系统的SQL表字段详细信息列出，全方位呈现了该学习系统的设计思路与架构。

5 系统实现

本章主要阐述各个模块中的主要功能实现，以及实现功能的原理步骤，展示系统的实际效果。

5.1 开发工具及运行环境介绍

开发工具有：

1. vscode：编程开发的集成工具，配合智能化插件高效开发系统。
2. Navicat：数据库管理工具，通过运行sql语句管理数据信息。
3. Chorme浏览器：用于系统的查看和调试

系统的运行环境：

1. 操作系统：Window 11 / Mac
2. Go版本：Go 1.23 及以上
3. Node版本“16.0 以上
4. 数据库 MySQL 5.7以上

5.2 用户注册登录模块实现

在前端部分，需要创建一个表单，用于手机用户信息，如用户名、密码、电子邮件等。提交表单之前，进行基本的验证，如检查用户名是否已存在、是否填写了所有必填字段，密码是否满足复杂度要求等。在用户填写完邮箱后，需要点击获取验证码按钮请求后端。在后端随机生成一个数字，通过smtp包，使用自己的私人邮箱发送邮件验证码给用户，同时设置一个过期时间，将验证码存储在Redis中。用户获取到验证码后，需要携带验证码，使用AJAX将表单内容以JSON格式发送到后端服务器。后端会重新查询验证码，验证是否通过。通过后会将生成Token返回给用户，将用户数据加密后，存储在数据库中。后续请求都会携带对应的Token信息，供给后端验证以此来防范CSRF攻击。注册登录时序图10所示。用户登录界面如图11所示。

图示

AI 生成的内容可能不正确。

图10 登录注册时序图

图形用户界面, 网站

AI 生成的内容可能不正确。

图11 注册界面样式

5.3 文章管理模块实现

5.3.1 文章创建编辑

在首页点击创作者中心，可以跳转到文章创建页面，文章创建编辑模块使用WangEditor编辑器，能够实现文字加粗、下划线、斜体、文字颜色、图片插入等基本的功能。在完成文章编写后，需要填写文章的基本信息，报错文章所属模块、摘要、封面信息等。点击发布，文章内容保存在content字段，在文章，存储在数据库中。同时系统接入了豆包大模型接口，存储在数据库前会进行文章内容的总结，生成一份智能总结，存储在数据库中，下次浏览文章即可看到。文章创建页面如下图12所示。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图12 文章创建

5.3.2 文章浏览

在首页可以通过左侧筛选栏，筛选出属于不同模块的文章内容。同时通过上方的筛选框可以进行文章关键字的查询。在首页的文章内容的展示区域，露出了文章的标题、封面、浏览量、点赞量等信息。点击文章既可跳转到文章浏览页面。在文章浏览界面左侧，可以进点赞、评论、收藏等操作。右侧栏有作者的基本信息，通过点击私聊按钮可跳转到私聊界面和作者进行交流。

图形用户界面, 应用程序

AI 生成的内容可能不正确。

图13 首页

图形用户界面, 应用程序, Teams

AI 生成的内容可能不正确。

图14 文章浏览页面

5.4 互助功能模块实现

在互助帮学界面中，在最上方为发布帖子的输入框，用户可以在上方进行帖子的编辑，包括文字图片表情等内容，点击发布按钮展开popover组件填写帖子的基本信息后，即可确定发布，管理员在后台审核过后，帖子即可在主页被浏览到。用户可以点击帖子下方的评论icon对帖子进行评论。同时也可以点击用户头像，再点击私聊按钮对用户进行私聊操作。互助帮学模块的实现如下图15所示。

图形用户界面, 应用程序, Teams

AI 生成的内容可能不正确。

图15 互助帮学页面

5.5 交流功能模块实现

本系统提供了私聊的功能，可以发送表情，图片，文字信息。跳转到私聊界面后，会发起websocket请求，链接到连接池中。私聊信息通过websocket进行内容发送。

5.6 后台管理模块实现

在系统的后台管理模块中，提供了管理员管理用户账号、文章、求助信息、公告配置的内容。三个管理模块由筛选框表格页组成，在前端，封装了Filter-Bar过滤组件，通过配置项渲染筛选框，通过简单的配置信息即可渲染出顶层筛选框，每当筛选框内容改变的时候，会进行列表请求，获取新的数据。以此渲染到表格中。同样的，表格部分也是通过配置信息以及Table组件，进行渲染展示。表格除了基本的信息展示外，提供了操作列表，供给管理员进行编辑和删除操作，通过点击编辑按钮弹出弹窗表单，进行内容编辑。

5.7 系统表单组件实现

本系统使用的表单基于Formily封装，Formily是阿里巴巴开源的一个表单系统，底层是对表单数据用 es6 的 Proxy 代理，对数据进行劫持来做发布订阅，以达到更好的性能，无需做任何优化即可获得超高的性能优势，依赖追踪，精确更新，按需渲染，让我们的表单真正做到了只需关注业务逻辑，无需考虑性能问题。Formily提供了 SchemaField 这样的协议驱动组件，同时是基于标准 JSON-Schema 的驱动，让表单开发可以变得更加动态化，可配置化。基于Formliy，我在其基础上封装了一个通用表单组件，精简的整个表单系统的能力，只暴露出了表单初始化、回填、校验、提交、提交成功、提交失败等生命周期钩子，同时充分使用了JSON-Schema协议驱动表单的渲染，配置。将表单开发抽象成配置信息的开发，大大提高了表单能力的开发效率。

5.8 本章小结

本章详细的介绍了跨专业互助学习系统各个功能模块的实现。介绍了开发使用的技术栈以及详细的功能是如何开发，包括注册登录、文章管理、互助内容管理、后台管理等等。

6 系统测试实现