



浙江工业大学

本科毕业设计论文

题目： 课程知识图谱系统的设计及开发

学 院： 信息工程学院

专 业： 通信工程

班 级： 2104 班

学 号： 202105110503

学生姓名： 陈耿洋

指导老师： 方路平教授

提交日期： 2025 年 5 月

**Dissertation Submitted to Zhejiang University of Technology
for the Degree of Bachelor**

**Design and Development of Course Knowledge
Graph System**

**Student: Gengyang Chen
Advisor: Prof. Fang Luping**

**College of Information Engineering
Zhejiang University of Technology
June 2025**

浙江工业大学
本科生毕业设计(论文、创作)诚信承诺书

本人慎重承诺和声明：

1. 本人在毕业设计（论文、创作）撰写过程中，严格遵守学校有关规定，恪守学术规范，所呈交的毕业设计（论文、创作）是在指导教师指导下独立完成的；

2. 毕业设计（论文、创作）中无抄袭、剽窃或不正当引用他人学术观点、思想和学术成果，无虚构、篡改试验结果、统计资料、伪造数据和运算程序等情况；

3. 本人严格按照浙江工业大学本科生毕业论文（设计）教学工作规范相关规定使用 AI 工具，未用于选题设计、创新方法构建、数据生成、结论总结等核心学术环节，AI 工具使用情况详见附页；

4. 若有违反学术纪律的行为，本人愿意承担一切责任，并接受学校按有关规定给予的处理。

学生（签名）：

陈耿祥

2025 年 5 月 30 日

本科生毕业设计（论文、创作）任务书

专业 通信工程 班 级 2021 通信工程 04 学生姓名/学号 陈耿洋/202105110503

一、设计（论文、创作）题目：课程知识图谱系统的设计及开发

二、主要任务与目标：

设计并实现一个课程知识图谱构建系统，该系统允许教师和学生手动输入或编辑知识点及其关系，并关联各种形式的教学材料，包括但不限于文本资料、相关链接、教学视频和习题。此外，系统应能通过图形化界面直观地展示这些信息，特别是利用思维导图等可视化工具来帮助用户更好地理解和学习课程内容。系统还需提供 API 接口以供外部应用调用，如在线学习平台。

三、主要内容与基本要求：

- 1.完成课程知识图谱系统的总体架构设计，包括数据存储、用户界面设计、前后端交互等模块。
- 2.开发基于 Vue.js 的前端界面，使用户能够方便地添加、编辑和查看知识点以及它们之间的关系；同时，也允许用户上传或关联各类教学材料（文本、链接、视频、习题等）。
- 3.使用 Spring Boot 框架搭建后端服务，处理业务逻辑，管理数据传输和安全问题，为前端提供稳定的数据支持和服务接口。
- 4.构建 MySQL 数据库模式，用于存储课程资料、知识点、教学材料及其相互关系。
- 5.提供 RESTful API 接口，支持前后端分离架构，使前端可以调用后端服务，同时也便于第三方应用集成。
- 6.集成思维导图或其他适合的知识可视化工具，使用户可以通过图形化的方式浏览和理解知识结构。
- 7.撰写毕业论文和提交相关设计文档、代码、测试报告等。

四、计划进度：

- 1.2024.12.1~2025.2.28 收集有关知识图谱、前端开发（Vue.js）、后端开发（Spring Boot）、数据库管理（MySQL）和技术可视化（如思维导图）的技术资料；完成文献综述和外文翻译；熟悉课题，做好开题准备。
- 2.2025.3.1~3.7 完成开题报告，参加开题交流；
- 3.2025.3.8~4.18 完成系统的主要功能模块的设计与编码，接受中期检查；
- 4.2025.4.19~5.23 进行系统集成测试，优化性能，完成最终版本。撰写毕业论文初稿，并进行修改；
- 5.2025.5.24~6.2 准备毕业论文答辩，提交所有必要的文档资料

五、主要参考文献：

- [1] Vue.js 官方文档 <https://vuejs.org/>
- [2] Spring Boot 官方文档 <https://spring.io/projects/spring-boot>.
- [3] jsmind 官方文档 <https://github.com/hizzgdev/jsmind>

[4] 王志鹏, 杨强. 知识图谱构建技术综述[J]. 软件学报. 2021

任务书下发日期 2024 年 12 月 1 日

设计（论文、创作）工作自 2024 年 11 月 30 日 至 2024 年 5 月 27 日

设计（论文、创作）指导教师

方路平

系主任（专业负责人）

蔡树凤

主管院长

何德峰

课程知识图谱系统的设计及开发

摘 要

随着教育信息化的深入推进,传统教学资源管理方式在知识碎片化、关联性不足及可视化程度低等方面逐渐显现出弊端,难以满足师生对知识深度理解与高效学习的迫切需求。基于此,本研究以构建课程知识图谱系统为核心,旨在通过技术手段实现课程知识点的结构化组织、多模态资源整合及交互式可视化展示,从而提升教学资源的利用效率与学习者的认知体验,为教育信息化发展提供实践参考。

本研究围绕课程知识图谱系统的设计与实现展开,采用前后端分离模式,构建包含数据层、服务层与应用层的三层架构。数据层基于 MySQL 设计关联数据库,支持知识点、教学资源及关系的高效存储;服务层基于 Node.js 和 Express.js 框架开发 RESTful API,实现业务逻辑处理与接口封装;应用层通过 Vue.js 搭建交互式前端界面,集成思维导图工具(jsMind)实现知识结构的动态可视化。支持教师手动管理知识点、多模态资源关联,教师与学生可视化展示与交互,开放了接口与集成能力。

本研究成功设计并实现了课程知识图谱系统,通过结构化知识建模与动态可视化,显著降低知识管理的复杂度,助力师生快速构建与理解课程体系。同时打破了传统资源孤岛,实现知识点与多样化教学材料的深度关联,为个性化学习提供数据基础。未来可通过引入自然语言处理技术实现知识自动抽取,结合推荐算法优化资源推送策略,进一步提升系统的智能化与实用性。

关键词: 知识图谱, 教育信息化, Vue.js, 可视化教学

DESIGN AND DEVELOPMENT OF COURSE KNOWLEDGE GRAPH SYSTEM

ABSTRACT

With the deepening of educational informatization, traditional teaching resource management methods have gradually shown drawbacks in knowledge fragmentation, insufficient correlation, and low visualization level, which are difficult to meet the urgent needs of teachers and students for deep understanding and efficient learning of knowledge. Based on this, this study focuses on constructing a curriculum knowledge graph system, aiming to achieve structured organization of curriculum knowledge points, integration of multimodal resources, and interactive visualization display through technological means, thereby improving the efficiency of teaching resource utilization and learners' cognitive experience, and providing practical reference for the development of educational informatization.

This study focuses on the design and implementation of a course knowledge graph system, using a front-end and back-end separation model to construct a three-layer architecture consisting of a data layer, a service layer, and an application layer. The data layer is designed with an associated database based on MySQL, supporting efficient storage of knowledge points, teaching resources, and relationships; The service layer develops RESTful APIs based on the Node.js and ExpressJS frameworks to achieve business logic processing and interface encapsulation; The application layer uses Vue.js to build an interactive front-end interface and integrates the mind mapping tool (jsMind) to achieve dynamic visualization of knowledge structures. Support teachers to manually manage knowledge points and link multimodal resources, visualize and interact with teachers and students, and open up interfaces and integration capabilities.

This study successfully designed and implemented a course knowledge graph system, which significantly reduces the complexity of knowledge management through structured knowledge modeling and dynamic visualization, and helps teachers and students quickly construct and understand the course system. At the same time, it breaks the traditional resource island, achieves deep correlation between knowledge points and diversified teaching materials,

and provides a data foundation for personalized learning. In the future, automatic knowledge extraction can be achieved by introducing natural language processing technology, combined with recommendation algorithms to optimize resource push strategies, further enhancing the intelligence and practicality of the system.

KEY WORDS: knowledge graph, educational informatization, vue.js, visual teaching

目 录

摘 要	I
ABSTRACT	II
第 1 章 绪 论	1
1.1 研究背景	1
1.2 研究意义	2
1.3 国内外研究现状	3
1.4 研究的主要任务	3
1.5 论文结构安排	4
第 2 章 系统开发平台与技术	5
2.1 系统开发工具	5
2.1.1 Visual Studio Code	5
2.1.2 DBEaver	5
2.2 系统开发技术	6
2.2.1 Vue	6
2.2.2 Element Plus	7
2.2.3 jsMind	8
2.2.4 Express.js	9
2.2.5 FFmpeg	10
2.2.6 Graphviz	11
2.2.7 BLOB	11
2.2.8 PDF.js 组件包	13
2.2.9 微软 Office Online 服务	13
2.3 本章小结	14
第 3 章 系统设计及实现	15
3.1 系统总体设计	15
3.1.1 系统模块功能	15
3.1.2 系统布局设计	16
3.2 知识图谱操作模块	16
3.2.1 功能设计	16
3.2.2 技术实现	17
3.3 知识图谱可视化界面功能项设计	20
3.3.1 功能设计	20

3.3.2 技术实现	20
3.4 多模态资源关联模块	21
3.4.1 功能设计	21
3.4.2 技术实现	23
3.5 后端存储模块	25
3.5.1 技术实现	25
3.6 本章小结	26
第 4 章 系统功能展示.....	27
4.1 知识图谱可视化界面功能展示	27
4.2 知识图谱操作模块功能实现	29
4.3 多模态资源关联模块功能实现	32
4.4 本章小结	34
第五章 总结	35
参考文献	36
致谢	39

第1章 绪论

1.1 研究背景

思维导图是英国学者东尼·博赞于 20 世纪 70 年代提出的一种将发散思维进行可视化的一种思维工具^[1]。直到 20 世纪 60 年代左右,研究人员开始研究知识表示以更好地理解自然语言,并开发了早期系统,例如麻省理工学院的 ELIZA。在 2000 年代初期,知识表示和语义 Web 社区共同努力在网络规模上标准化广泛使用的知识表示语言,例如资源描述框架(RDF)和 Web 本体语言(OWL),使用这些语言的大规模知识库后来被更广泛地称为知识图谱,由于它们有用的图结构,可以实现逻辑推理和基于图的学习。2007 年左右,随着 DBpedia 项目^[2]的启动,从维基百科中抽取结构化信息并转化为机器可读的形式,成为构建大规模知识图谱的一个重要里程碑。同样重要的还有 Freebase^[3],一个由 MetaWeb Technologies 开发的大规模协作知识库,后来被谷歌收购。到 2012 年,谷歌正式推出了“知识图谱”^[4],旨在改进其搜索引擎的结果,改善用户的搜索质量和搜索体验,使搜索结果不仅仅是网页链接,还包括直接的答案和相关信息。随着人工智能技术的发展,知识图谱在其中扮演的角色越发重要,知识图谱技术在智能搜索、信息提取^[5]、多样化推荐、智能应答^[6]等多方面均被运用。然而,传统教育系统仍面临知识碎片化、关联性不足及可视化程度低等挑战,难以满足师生对知识深度理解的需求。与此同时,智慧教育作为数字时代的教育新形态,被《中国智慧教育蓝皮书》明确为教育变革的核心方向,其核心理念与实践路径正推动教育体系的全面重构^[7]。国家层面亦出台《智慧教育平台 基本功能要求》和《智慧教育平台 数字教育资源技术要求》等政策文件为智慧教育的发展提供了进一步的支持^[8]。在国家推动教育数字化战略全面实施背景下,我国成功构建了世界最大规模的教育教学资源库,并陆续出台了多项数字教育领域的标准化指南。同时,在线开放课程(MOOC)的开发与应用规模已跃居国际前列,成为全球教育数字化发展的标杆^[9]。

课程知识图谱系统作为智慧教育体系的关键组成部分,其构建与应用尤为重要。现有知识图谱如 Freebase、DBpedia 虽在通用领域表现优异,但教育场景中仍存在多模态资源整合不足、动态交互能力有限等问题。此外,实体、关系、尾实体传统三元组表示

方式虽高效，但因其高度结构化特性，导致操作灵活性不足。近年来，知识图谱表示学习^[10]通过向量化嵌入优化了知识操作效率，为自动化构建提供了技术基础。因此，当前针对探索课程知识图谱的构建方法，不仅契合智慧教育的战略方向，同时为加速教育数字化进程与实现教育智能化提供了关键支撑。

1.2 研究意义

拓展知识图谱系统的模块化设计理论。本研究针对课程知识图谱的构建与展示需求，提出了涵盖知识点管理、多模态资源关联、可视化交互及开放接口的模块化系统架构。传统研究多聚焦于知识图谱的构建算法或通用领域应用，而本工作通过细化教育场景中的功能模块设计，为领域知识图谱系统的开发提供了新的理论框架。例如，可视化交互模块结合思维导图工具，如 jsMind，深化了知识图谱动态展示与用户认知效率关联的理论探讨，弥补了现有研究在交互设计层面的不足。

推动教育场景下知识组织与多模态融合理论的创新。传统课程知识图谱构建多依赖人工标注或半自动化工具，而本研究通过多模态资源关联模块的设计，系统化整合文本、视频、习题等异构资源，并建立其与知识点的语义映射关系。这一实践为教育领域知识的多维度表达与融合提供了理论依据，丰富了知识组织理论的内涵。同时，开放接口模块的设计探索了知识图谱与外部系统，如在线学习平台的协同机制，为教育信息化的生态化发展贡献了新的理论视角。

提升课程知识管理的效率与用户体验。通过知识点管理模块的网页化实现，教师可直接在线编辑知识点及其关联关系，支持多人协作与实时更新，显著降低传统人工整理的时间成本。可视化展示与交互模块利用图形化界面，如思维导图，直观呈现知识结构，有效提升用户对复杂知识体系的理解效率。此外，多模态资源关联模块通过统一平台整合分散的课程资源，解决资源孤岛化问题，为教学与学习提供一站式支持。

支持智慧教育生态的构建与扩展。开放接口与集成能力模块通过标准化 RESTful API 设计，允许第三方应用，如智能推荐系统、在线考试平台，无缝接入课程知识图谱系统，促进教育数据的流通与共享。例如，基于知识图谱的个性化学习路径推荐可借助接口实时获取知识点关联数据，从而提升推荐的精准性。这一设计不仅扩展了系统的应用场景，还为教育信息化平台间的协同创新提供了实践范例，助力智慧教育生态的可持续发展。

推动个性化学习模式的发展，依托完善的教学知识图谱，可以更精准的解析学生的

学习状态和各知识点的掌握情况，可生成定制化的学习路径，并匹配精准的资源推荐策略。这不仅有助于实现差异化教学，还能有效提升学习成效。

1.3 国内外研究现状

自 Google 在 2012 年提出知识图谱概念^[4]以来，该技术已成为人工智能研究的关键方向，在机器学习、自然语言处理等多学科领域发挥着基础支撑作用。当前研究主要聚焦于知识表示建模、实体关系抽取、跨源知识融合、语义推理机制、存储查询优化及质量评估体系六大方向^[11]。科技巨头如 Google、Meta 和 Microsoft 已将知识图谱技术深度整合至搜索优化、智能助手及语义理解系统中，显著提升了信息处理效能。

在教育领域应用方面，研究者开发了多种教育知识图谱模型，来推动其在智慧教育中的多方面应用。李振、周东岱等学者提出的 EKGCM 模型^[12]通过四阶段构建流程，解决了通用图谱在教育领域的迁移适应问题。李艳燕、张香玲团队构建的学科知识图谱^[13]则探索了智慧教育场景的具体应用路径与场景。钟卓等人研发的 KQA 模型^[14]从四个维度重构教育知识体系，有效应对内容碎片化与能力刻画不足等问题。也有研究者致力于对如何高效构建知识图谱的研究，如黄焕设计了一个用于提高学生编程能力的知识图谱的参考案例^[15]。还有清华大学唐杰团队开发的 Science Knowledge Graph 运用条件随机场技术进行学者信息抽取，通过层次分析法揭示学术关系和研究趋势^[16]。上海交通大学的 AceKG 则创新性地将语义数据嵌入网状架构，可视化呈现学术发展脉络^[17]。这些知识图谱技术可以帮助教育研究者更加便捷的了解学科的学术前沿与发展方向，可有效辅助师生群体精准把握核心知识点，及时跟进学科演进脉络，从而提升教学实践的有效性与前瞻性。

国际研究方面，Knewton 公司通过知识图谱建模学习路径节点关系^[18]，实现个性化学习推荐并提高了学习效率。Alshalabi 团队提出加权有向图算法，优化在线学习路径规划^[19]。Maria Sette 的研究则强调教师经验知识的图谱化整合，提升教学资源和学习顺序推荐精准度^[20]。这些实践表明，知识图谱技术正在重塑教育生态，既支持教师教学决策优化，又促进学生个性化学习体验。

1.4 研究的主要任务

一、系统总体架构设计。完成课程知识图谱系统的整体架构规划，涵盖数据存储模块、用户交互界面设计、前后端通信机制等核心模块。需确保系统功能模块化、可扩展

性及稳定性，为后续开发奠定基础。

二、前端功能开发与界面实现。基于 Vue.js 框架开发用户友好的前端界面，支持教师动态添加、编辑、删除知识点及关联关系，并提供教学材料如文本、链接、视频、习题等的上传与关联功能。要求界面简洁直观，操作流程符合用户体验设计原则。

三、后端服务与业务逻辑实现。采用 Node.js 和 Express.js 框架搭建高效、安全的后端服务，负责处理用户请求、数据校验、权限管理及业务逻辑执行。需保障数据传输安全性，并通过 RESTful API 为前端及第三方应用提供标准化接口。

四、数据库设计与数据管理。基于 MySQL 构建课程知识图谱的数据库模型，实现知识点、教学材料及其复杂关系的结构化存储。要求设计合理的数据表结构，优化查询效率，确保数据一致性和完整性。

五、系统接口开发与外部集成。提供开放的 RESTful API 接口，支持前后端分离架构，并允许外部应用，如在线学习平台通过标准化接口调用系统功能，实现数据共享与功能扩展。

六、知识可视化工具集成。集成思维导图，如 jsMind 或其他可视化工具，支持用户通过图形化界面，如树状图、网络图，动态浏览知识结构，增强课程内容的理解与学习效率。

1.5 论文结构安排

本文共分为五章，各章主要内容如下：

第一章，总结了课程知识图谱系统的研究背景，国内外相关领域的研究及应用，课程知识图谱系统研究的实际意义，课题研究的主要任务和本文的组织结构。

第二章，介绍了课程知识图谱系统的开发环境和使用技术。

第三章，介绍课程知识图谱系统的详细设计，功能组成、工作流程。

第四章，展现本课题系统的各功能的实现效果。

第五章，总结全文，提出系统存在的问题。

第 2 章 系统开发平台与技术

2.1 系统开发工具

2.1.1 Visual Studio Code

Visual Studio Code 是本系统开发过程中采用的核心代码编辑器。作为微软推出的开源轻量级工具，其具备以下特性：通过丰富的插件市场如 Vetur、Spring Boot Tools、MySQL Client 等，可灵活支持前端(Vue.js)、后端(Spring Boot)及数据库(MySQL)的全栈开发需求；内置调试终端、Git 版本控制、代码智能提示与语法高亮，显著提升编码效率；相较于传统 IDE 如 Eclipse 或 IntelliJ IDEA，VS Code 在资源占用与启动速度方面优势显著，适合快速迭代的中小型项目开发。



图 2-1 VS Code IDE 图标
Figure 2-1 VS Code IDE icon

在本系统中，VS Code 的生态为技术实现提供了关键支持：Node.js 与 Express.js 框架是本系统后端服务的核心技术栈。在 2019 年 Stack Overflow 组织的开发者调查中，Visual Studio Code 被认定为是最受开发者欢迎的开发环境。

2.1.2 DBeaver

DBeaver 是本系统数据库设计与运维过程中采用的核心管理工具。作为一款开源、跨平台的通用数据库客户端，其具备以下特性与项目适配性：提供表结构设计、数据增

删改查、SQL 脚本编辑与执行的图形化界面，降低数据库操作门槛，提升开发效率；支持 Windows、macOS 及 Linux 系统，适配团队协作中的异构开发环境；支持 CSV、Excel 等格式的数据迁移，便于教学资源的批量处理。



图 2-2 DBeaver 图标
Figure 2-2 DBeaver icon

DBeaver 的官方文档中明确强调其“企业级数据库工具”定位，其稳定性已被 GitHub、Apache 等开源项目验证。在本系统中，其通过 mysql 驱动连接 Node.js 后端及 MySQL 数据库的协同工作流，体现了全栈技术工具链的高效整合。

2.2 系统开发技术

2.2.1 Vue

Vue^[21]是一套用于构建前端页面的渐进式框架，是本系统前端开发的核心框架，其轻量级、响应式设计及组件化特性为课程知识图谱系统的动态交互与可视化界面实现提供了技术支撑。与 React、Angular 等主流 MVVM 前端框架不同的是，Vue 被设计成可以从底部向上到各阶层应用。Vue 的核心库只关注画层，容易学习，容易与第三方库和现有的项目合并。另外，Vue 还可以利用现代工具链和各种支持类库，为复杂的单页应用提供驱动。



图 2-3 Vue.js 图标
Figure 2-3 Vue.js icon

通过双向数据绑定机制(v-model 指令)实现视图与数据的实时同步, 简化知识点编辑、关系更新等功能的开发复杂度; 将知识点卡片、资源上传表单、可视化面板等模块封装为可复用组件(.vue 文件), 提升代码可维护性与协作效率。

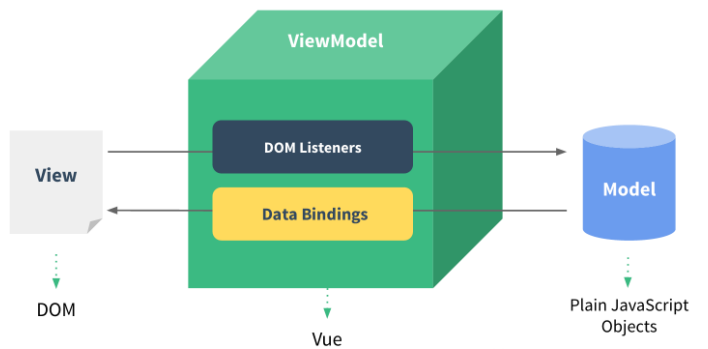


图 2-4 vue 的 mvvm 模型和生命周期函数
Figure 2-4 MVVM Model and Lifecycle Function of Vue

Vue.js 的官方文档明确指出其“渐进式框架”定位, 已被阿里巴巴、GitLab 等企业应用于生产环境。在本系统中, 其与 Node.js 后端、MySQL 数据库及 DBeaver 管理工具的技术协同性, 验证了全栈技术选型的合理性。

2.2.2 Element Plus

Element Plus 是一套基于 Vue 3 的开源 UI 组件库, 由饿了么团队开发。它继承了 Element UI 的设计理念, 并针对 Vue 3 的特性进行了优化, 其丰富的预制组件与高度

可定制性为课程知识图谱系统的界面设计与交互实现提供了高效解决方案,帮助开发者快速构建现代化的 Web 应用程序。



图 2-5 Element Plus 图标
Figure 2-5 Element Plus icon

Element Plus 是 Vue 3 生态中最受欢迎的组件库之一,被阿里云、腾讯文档等众多大型项目采用,其 GitHub 仓库 Star 数超 20k 与持续更新日志证明了其技术生命力。

2.2.3 jsMind

jsMind 是一款基于 JavaScript 开发的开源思维导图工具,是一个显示/编辑思维导图的纯 Javascript 类库,其基于 html5 canvas 和 svg 进行设计,支持在网页端动态创建、编辑和展示思维导图。其核心目标是为开发者提供一个轻量级、高性能的思维导图解决方案,适用于知识管理、项目规划、教育展示等多种场景。

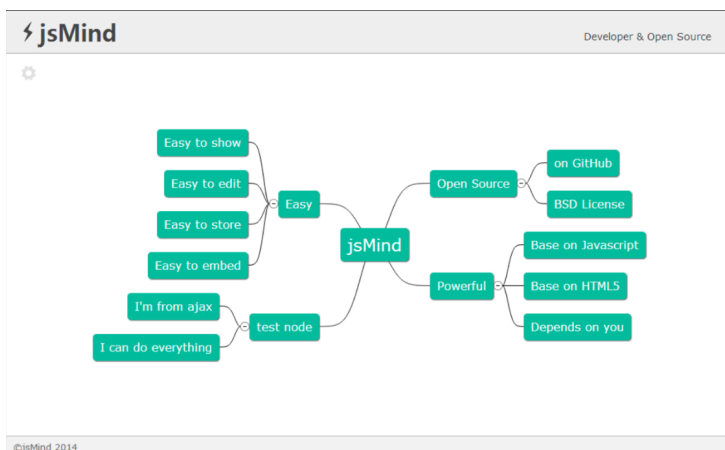


图 2-6 jsMind 的效果图示例
Figure 2-6 Example rendering of jsMind

jsMind 提供了强大的脑图编辑功能,支持节点的添加、删除、移动等操作。它内置

了多种主题，用户可以根据需要选择合适的主题，或者自定义主题。此外，通过 CSS 可以自定义节点的样式，包括背景色、字体颜色、鼠标悬停效果等。jsMind 还提供了丰富的 API，方便开发者进行二次开发和功能扩展。

2.2.4 Express.js

Express.js 是一个基于 Node.js 的轻量级 Web 应用框架，旨在简化 Web 应用和 API 的开发，其轻量、灵活的特性为课程知识图谱系统的 API 设计与业务逻辑实现提供了高效支持。

Express.js 通过中间件机制支持按需扩展如请求解析、跨域处理、身份鉴权等功能，简化了复杂逻辑的模块化开发；支持 RESTful API 设计规范；同时拥有庞大的开源生态，提供丰富的插件和第三方中间件。

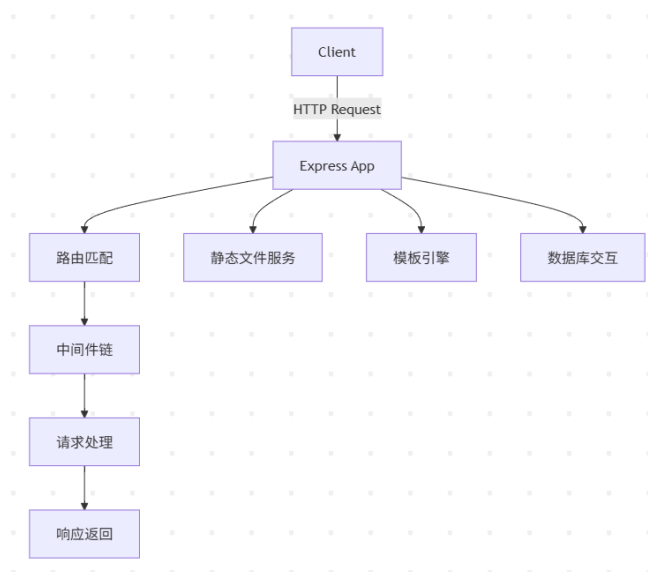


图 2-7 Express.js 的基本架构图

Figure 2-7 Basic architecture diagram of Express.js

Express.js 作为 Node.js 基金会孵化项目，GitHub 星标数超 60k，社区活跃度高，它通过提供灵活的路由、中间件机制和高效的请求处理能力，成为 Node.js 生态中最受欢迎的框架之一。无论是构建简单的静态网站，还是复杂的微服务架构，Express 都能提供高效的解决方案。通过合理的技术选型和开发实践，可以快速构建高性能、可扩展的

Web 应用。

2.2.5 FFmpeg

FFmpeg^[22]是一个开源的音视频处理工具集，支持多媒体文件的编解码、转码、流媒体传输、滤镜处理等功能。它基于 C/C++ 开发，跨平台兼容性强，广泛应用于音视频处理领域。它提供了录制、转换以及流化音视频的完整解决方案。它包含了非常先进的音频/视频编解码库 libavcodec，为了保证高可移植性和编解码质量，libavcodec 里很多 code 都是从头开发的。



图 2-8 FFmpeg 图标
Figure 2-8 FFmpeg icon

支持如 MP4、AVI、MKV、FLV、MP3、AAC 等数百种音视频格式；FFmpeg 可快速将音视频文件从一种格式转换为另一种格式，同时优化如分辨率、码率、帧率等编码参数还支持实时音视频流的推流和拉流。甚至可以通过 `filter_complex` 参数实现动态滤镜，如缩放、旋转、水印、字幕叠加。

FFmpeg 作为开源音视频处理领域的基石工具，凭借其强大的功能、灵活性和跨平台兼容性，成为多媒体开发的核心技术之一。

2.2.6 Graphviz

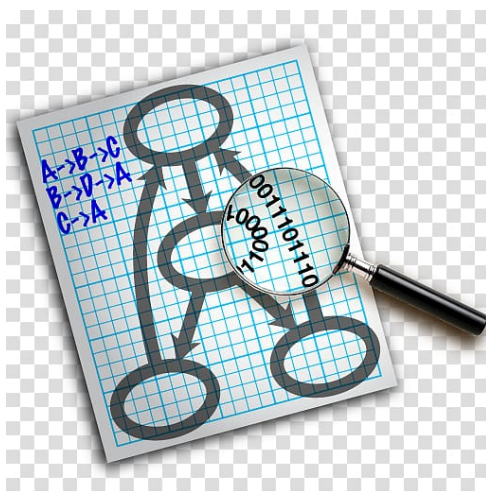


图 2-9 Graphviz 的概述图
Figure 2-9 Overview of Graphviz

Graphviz 是一个开源的图形可视化工具，专为创建和操作结构化图形，如流程图、网络图、组织结构图等而设计。它基于 DOT 语言，一种图形描述语言，通过自动布局算法生成高质量的图形。Graphviz 支持多种输出格式，如 PNG、SVG、PDF，广泛应用于学术研究、软件开发和系统设计等领域。

支持通过算法自动计算节点和边的最优位置，从而减少手动调整。还可通过 Python、Java 等编程语言调用 Graphviz 功能，支持多语言。同时支持 Windows、Linux 和 macOS 等系统，跨平台兼容性好。支持 PNG、SVG、PDF、HTML 等丰富的输出格式 [23]。

Graphviz 凭借其强大的自动布局功能和灵活的图形描述语言，现已成为学术研究和软件开发中的重要工具。无论是绘制复杂的数据结构图，还是分析系统的依赖关系，Graphviz 都能提供高效的解决方案。尽管存在一定的学习成本，但其在自动化和可扩展性方面的优势使其在数据可视化领域占据重要地位。随着技术的进步，Graphviz 在智能化和跨平台应用中的潜力将进一步释放。

2.2.7 BLOB

BLOB(Binary Large Object)，即二进制大对象，是一种专门设计用于存储大规模二

进制数据的容器。在计算领域中，BLOB 通常作为数据库中的一个字段类型，用来保存如图像、音频等二进制文件。这类文件由于其较大的体积，需要特定的技术手段来进行有效的管理，包括上传、下载以及数据库存储等方面。

Blob 接口提供了一种处理原始二进制数据的方式。通过这个接口，开发者可以利用 Blob 对象的 `slice()` 方法轻松访问和操作其中包含的数据片段。除此之外，每个 Blob 对象还包含了两个重要的属性：`size` 和 `type`，分别用于指示数据的大小和 MIME 类型。这些特性使得 Blob 在处理多媒体数据或者其他任何形式的大规模二进制信息时显得尤为有用。

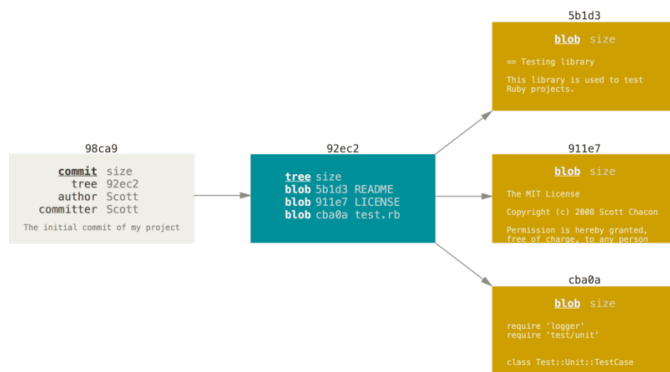


图 2-10 Blob 的概述图

Figure 2-10 Overview of Blob

在 HTML 5 中，开发者能够通过 XML HttpRequest 对象的 `send` 方法向服务器传输 Blob 对象，由于任何 File 实例本质上也是一个 Blob 实例，因此上传文件的过程同样可以通过发送 Blob 对象的方法来上传文件。

具体使用 XML HttpRequest 对象发送 Blob 对象时，在网页上配置一个“复制当前页面”的按钮与一个进度条控件 `progress` 元素，当用户单击“复制当前页面”按钮时脚本程序会将整个页面的内容，包括 HTML、CSS 及 JavaScript 代码，封装成一个 Blob 对象。然后，这个 Blob 对象指定为 XML HttpRequest 对象的 `send` 方法的参数值的方法向服务器端发送该 Blob 对象，服务器端接收到该 Blob 对象后将其保存为一个文件，文件名为“副本_”加上原页面文件名(含扩展名，不涉及路径信息)。在向服务器端发送 Blob 对象的同时，前端界面上的进度条也会实时更新，反映上传进度^[24]。

2.2.8 PDF.js 组件包



图 2-11 PDF.js 的图标

Figure 2-11 PDF.js icon

PDF.js 是由 Mozilla 开发并维护的一个基于 JavaScript 的开源项目，旨在使用 HTML5 技术实现对 PDF 文档的解析与渲染。它允许开发者在网页环境中直接查看 PDF^[25]文件，无需依赖任何第三方插件，如 Adobe Reader。PDF.js 主要用于浏览器端，提供了丰富的 API 接口，方便用户定制化需求。

目前已知的前端渲染 pdf 组件都是基于 pdf.js 进行封装。

2.2.9 微软 Office Online 服务

随着云计算的发展，越来越多的企业和个人选择在线办公解决方案来提高工作效率和协作能力。微软 Office Online 是基于云的服务，允许用户通过浏览器直接查看和编辑 Word、Excel、PowerPoint 等文档，无需安装本地软件。

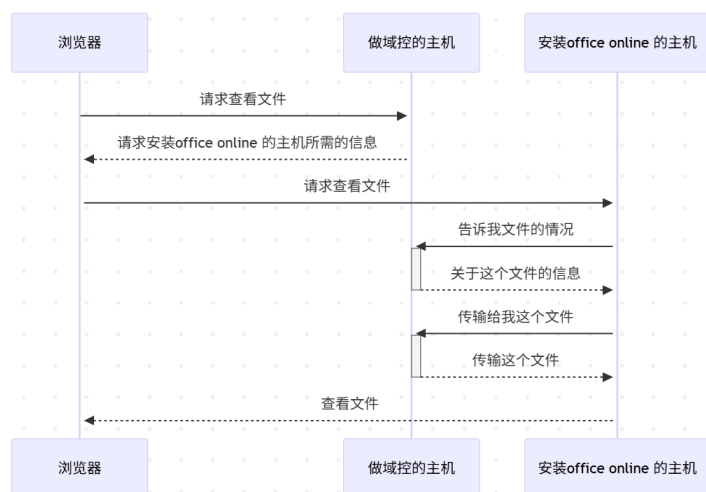


图 2-12 微软 Office Online 服务流程图

Figure 2-12 Microsoft Office Online Service Flow Chart

做域控的主机(WOPI Server), 安装 office online 的主机(WOPI Client)。

微软 Office Online 通过嵌入式框架, 如 <iframe> 标签将 Office 文档加载到网页中。首先需要将目标文档上传至 OneDrive 或 SharePoint, 或者提供可公开访问的 URL。再利用 Office Online 提供的 API 或手动配置生成用于嵌入网页的 URL。最后通过 <iframe>将生成的 URL 嵌入到目标网页中, 以实现在浏览器中查看和编辑文档。

2.3 本章小结

本章主要介绍了课程知识图谱系统项目中所使用的系统开发环境和其中的关键技术。

第3章 系统设计与实现

3.1 系统总体设计

3.1.1 系统模块功能

课程知识图谱系统根据任务书中提出的要求设计了四个主要功能模块，分别是：知识图谱操作模块、知识图谱可视化界面、多模态资源关联模块和后端存储模块。其中最核心的就是多模态资源关联模块这部分，如图 3-1 所示。

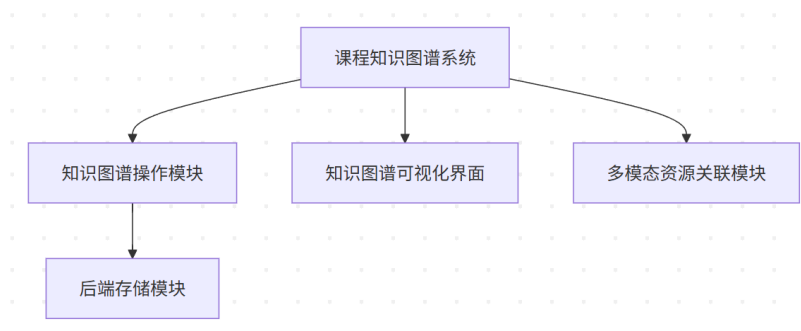


图 3-1 系统的功能模块组成

Figure 3-1 Composition of functional modules in the system

① 知识图谱操作模块：负责控制该知识图谱能否进行修改操作；对知识图谱中的节点进行添加、删除、修改、查询；一键导入、导出知识图谱；一键将知识图谱转换成学习路径；对知识图谱中的节点进行展开和收拢。

② 知识图谱可视化界面：展示数据库中存储的知识图谱；放大缩小知识图谱展示的视野大小；允许点击知识图谱中的节点进行更多的交互；同样也支持点击知识图谱内的按钮进行展开和收拢节点

③ 多模态资源关联模块：在点击了节点后会在旁边出现冒泡框，在这个冒泡框内支持存放与该节点有关的文件、视频、富文本内容。

④ 后端存储模块：建立前后端交互的桥梁，存储节点间的关系和个节点中冒泡框内的内容。

3.1.2 系统布局设计

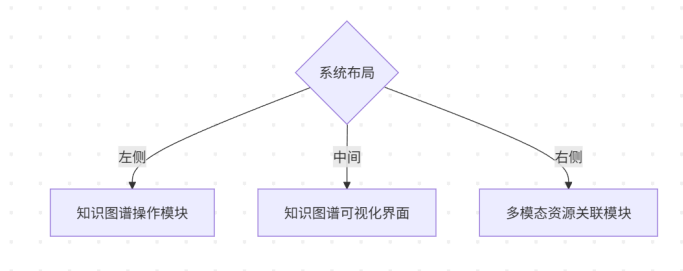


图 3-2 系统大致布局

Figure 3-2 General layout of the system

如图 3-2 所示，本知识图谱系统的网页界面是由三部分组成的，最左侧是知识图谱操作模块、中间部分是知识图谱可视化界面、右侧是多模态资源关联模块。Web 网页使用 Vue.js 框架进行搭建^[26]，通过 vue-router 建立 url 与页面组件的关联，并利用 axios 包^[27]进行 ajax 异步请求更新页面。

3.2 知识图谱操作模块

知识图谱操作模块是本系统的核心交互层，通过集成动态节点管理、搜索优化与数据迁移功能，支持用户对课程知识图谱进行高效编辑与维护。

3.2.1 功能设计

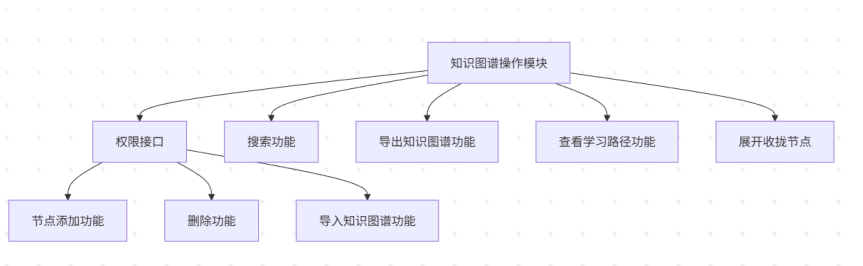


图 3-3 知识图谱操作模块功能示意

Figure 3-3 Function diagram of knowledge graph operation module

知识图谱操作模块旨在为教师与学生提供直观的操作界面，其具有的功能如图 3-3 所示。

通过工具栏内的按钮实现知识图谱知识点的实时增删；在搜索框内基于关键字匹配

节点标题或者节点冒泡框内富文本框的内容进行模糊搜索，高亮显示结果并支持跨层级跳转；支持从 Markdown 文件导入知识结构，或将现有图谱导出为标准 Markdown 格式，便于跨平台复用；提供“展开/折叠”按钮调节视图层级深度，适配不同复杂度的知识网络浏览需求。

3.2.2 技术实现

添加节点功能实现：用户选中父节点后，调用 `AddNode` 函数生成唯一节点 ID，通过 `jsmind.add_node` 方法动态渲染新节点；使用 `axios` 向后端发送 POST 请求 (`/api/mindmap`)，保存节点 ID、父节点 ID 及层级关系至 MySQL 数据库；选中的父节点是根节点还会使用 `balanceRootChildren` 函数平衡根节点子项分布，避免视图偏斜。

删除节点功能实现：调用 `jmind.remove_node` 移除节点及其子节点，同步发送 DELETE 请求 (`/api/mindmap/{nodeId}`) 清理数据库关联数据；

基于以上功能为知识图谱操作模块设计了一张数据库表，表结构的设计如表 3-1 所示。

表 3-1 知识图谱操作模块数据库存储表格
Table 3-1 Knowledge Graph Operation Module Database Storage Table

字段名	数据类型和约束	说明
id	INT NOT NULL AUTO_INCREMENT	无意义
parent_id	VARCHAR(255) CHARACTER SET UTF8MB4 COLLATE UTF8MB4_0900_AI_CI DEFAULT NULL	父节点id
topic	VARCHAR(255) CHARACTER SET UTF8MB4 COLLATE UTF8MB4_0900_AI_CI NOT NULL	节点名称
node_id	VARCHAR(255) CHARACTER SET UTF8MB4 COLLATE UTF8MB4_0900_AI_CI NOT NULL	节点id
level	INT DEFAULT NULL	节点层次

在上表中，`parent_id` 字段用于存储当前节点的父节点 id 信息；`topic` 字段用于存储节点名称信息；`node_id` 字段用于存储当前节点的 id 信息；`level` 字段用于存储当前节点的层次信息。

搜索框功能实现：实时监听搜索框输入，利用 `v-model` 绑定关键字 `searchKeyword`，再通过 `stripHtmlTags` 函数去除富文本标签后匹配节点内容；当关键字包含复杂逻辑时，发送 POST 请求 (`/api/mindmap/batch-get-content`) 批量获取节点文本框的文件，减少网络

请求次数；再通过 highlightNode 函数修改节点背景色



图 3-4 搜索功能流程图

Figure 3-4 Search function flowchart

导入导出思维导图功能实现：用户上传 markdown 文件后，parseMarkdownToMind 函数解析标题层级，按照节点前的#符号数与缩进层次来划分该节点属于的层级，再生成标准 node_array 的 jsmind 节点结构；然后调用/api/mindmap/replace 接口全量更新数据库，并调用函数重新初始化思维导图视图。导出逻辑则是遍历节点树，按 Markdown 语法拼接内容，利用 Blob 对象生成下载文件。

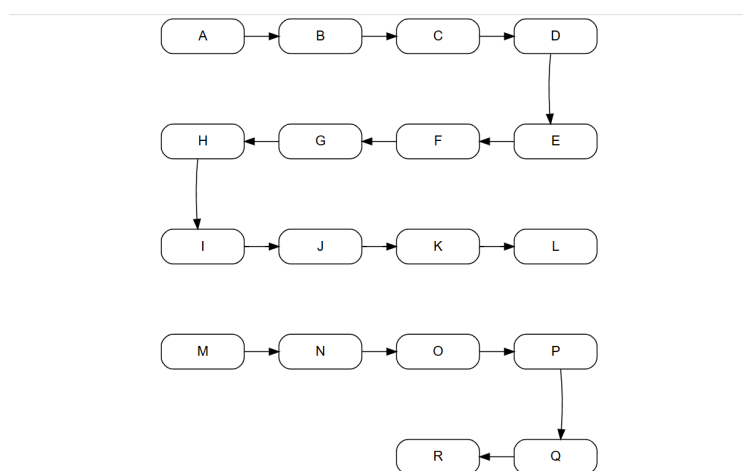


图 3-5 html 形式模拟下 Graphviz 兼容的 DOT 语言完成学习路径规划

Figure 3-5 Graphviz compatible DOT language completes learning path planning in HTML format simulation

转换成学习路径功能实现：先通过后端/api/mindmap/flowchart-data 接口返回节点数据，包含 node_id、topic、parent_id、route 及 importance 属性，再使用邻接表存储节点关系，通过递归遍历构建多层次路径链。再按照先前如图 3-4 所示用 html 网页模拟下固定数量的节点的学习路径的 DOT 语言编写方式，在 generateFlowchartDOT 函数中将邻接表中存储的节点关系转换为 DOT 语言格式的内容。然后动态加载 Viz.js 库，将 DOT 代码渲染为 SVG 矢量图从而实现学习路径展示，再通过 bindFlowchartClickEvents

函数绑定节点点击事件，实现与知识图谱的联动。



图 3-6 查看学习路径功能流程图

Figure 3-6 View the learning path function flowchart

知识图谱操作模块中所使用的函数如表 3-2 所示。

表 3-2 知识图谱操作模块函数列表

Table 3-2 List of knowledge graph operation module functions

函数名	参数	函数参数说明	函数说明
AddNode	无	无	添加节点函数
RemoveNode	无	无	删除节点函数
handleSearch	无	无	搜索函数
stripHtmlTags	html	html字符串	去除富文本标签
highlightNode	nodeId	节点id	对搜索到的节点进行变色
clearHighlight	无	无	清除变色节点的颜色
handleImportMindmap	file	文件对象	导入知识图谱函数
parseMarkdownToMind	mdContent	文件中的Markdown内容	Markdown内容转成node_array节点结构
initMindMap	无	无	重新初始化知识图谱
ExportMindmap	无	无	导出知识图谱函数
toggleFlowchart	无	无	转换成学习路径函数
bindFlowchartClickEvents	svgElement	学习路径中的节点	绑定学习路径中的节点点击事件
generateFlowchartDOT	allNodes	学习路径中的所有节点	将节点关系转换成DOT语言格式
cleanupFlowchart	无	无	关闭学习路径
collapse_all	无	无	收拢所有节点
expand	无	无	展开一层节点
expand_all	无	无	展开全部节点

3.3 知识图谱可视化界面功能项设计

知识图谱可视化界面通过集成动态渲染、交互操作与节点联动功能，实现课程知识结构的直观展示与高效探索。

3.3.1 功能设计

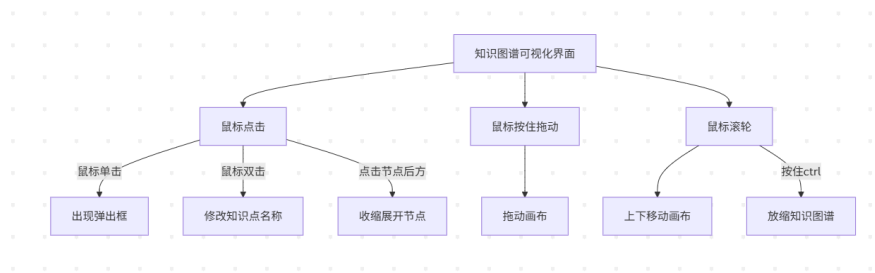


图 3-7 知识图谱可视化界面功能展示

Figure 3-7 Visual interface function display of knowledge graph

基于 jsMind 库实时渲染知识图谱，支持树状、网状等多种布局模式。通过背景色区分知识节点优先级，同时支持动态绑定。

允许用户按住鼠标拖拽画布进行自由的浏览复杂知识网络。

或者通过按住 ctrl 加鼠标滚轮实现视图缩放，方便用户查看整个或者部分知识图谱。

可以单击节点“+”图标展开子节点或者收拢节点，或通过工具栏一键展开或收拢全部层级。

可以单击节点来显示弹出框，进而呈现多模态资源关联模块。

可以双击节点来快速修改知识点的名称，并且会自动修改数据库中的相关字段。如图 3-5 所示。

3.3.2 技术实现

采用轻量级思维导图库 jsmind 框架来作为知识图谱的可视化引擎，在启用 draggable: true 后，用户可按住画布空白区域拖动。

通过 jmind.view.scale 方法动态调整缩放比例，绑定至滑块组件。

通过 bindNodeClick 函数绑定知识图谱中节点的点击事件，来触发弹窗与数据加载，为多模态资源关联模块作准备。

还允许根据各个节点在数据库中的 `importance` 字段来动态设置节点的背景色, 增强视觉层次。

知识图谱可视化界面中所使用的主要函数如表 3-3 所示。

表 3-3 知识图谱可视化界面函数列表

Table 3-3 List of Functions for Knowledge Graph Visualization Interface

函数名	参数	函数参数说明	函数说明
<code>bindNodeEvents</code>	无	无	事件绑定函数
<code>bindNodeClick</code>	<code>nodeId</code>	节点id	鼠标单击事件绑定

3.4 多模态资源关联模块

多模态资源关联模块是本系统的核心功能之一, 通过节点弹框实现知识点与文本、视频、文件等教学资源的动态绑定与管理。具备的功能如图 3-6 所示。

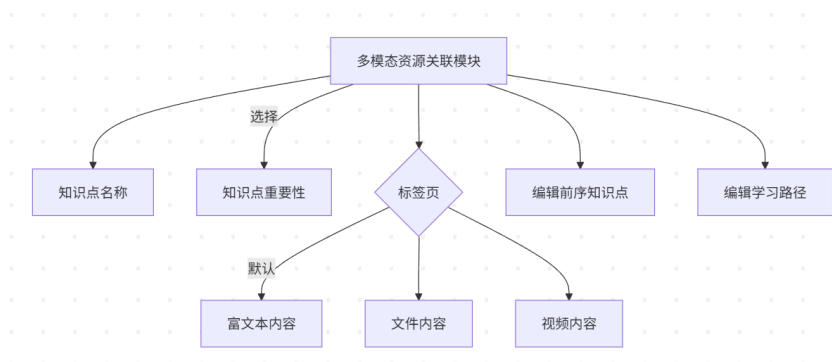


图 3-8 多模态资源关联模块功能展示

Figure 3-8 Function display of multimodal resource association module

3.4.1 功能设计

在节点的内容框中支持 Markdown 与 HTML 混合编辑, 实时预览图文内容; 并且内容修改后通过防抖机制, 300ms 延迟, 自动将数据保存至对应节点的 `html` 文件中; 同时也提供了显式保存按钮提供手动保存方式。

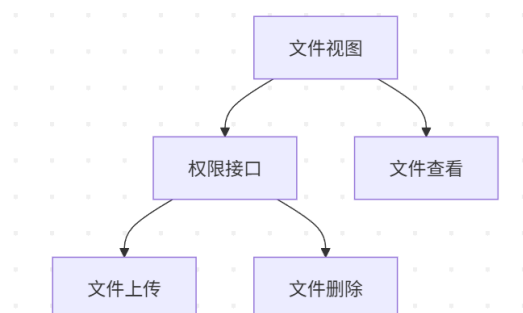


图 3-9 文件视图流程展示

Figure 3-9 Document View Process Display

如图 3-7 和图 3-8 所示，在多模态资源关联上，通过按钮来切换“文件”“视频”“编辑器”视图，对资源进行分类，优化信息展示密度，支持 PDF、PPT、Word 文档上传，提供在线浏览与下载；允许上传教学视频，集成 HTML5 播放器实现实时流畅播放；

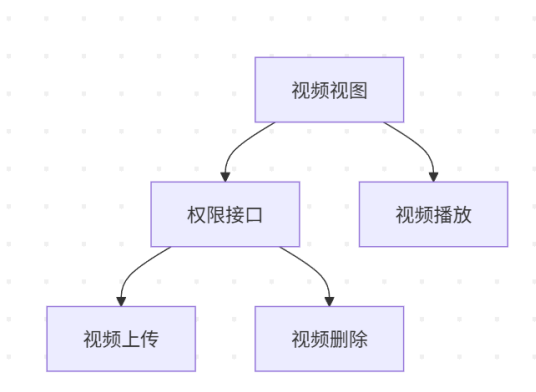


图 3-10 视频视图流程展示

Figure 3-10 Video View Process Display

前序知识点按钮是用来编辑或展示当前知识点的前序知识点，从而方便学生学习知识点。

编辑学习路径按钮是通过手动、自定义的方式来控制学习路径包含哪些知识点。

在表 3-4 中列出了与多模态资源关联模块相关的数据库表结构设计。

表 3-4 多模态资源关联模块数据库存储表格

Table 3-4 Multi modal resource association module database storage table

字段名	数据类型和约束	说明
content_file	VARCHAR(255) CHARACTER SET UTF8MB4 COLLATE UTF8MB4_0900_AI_CI DEFAULT NULL	存储富文本内容文件
files	TEXT CHARACTER SET UTF8MB4 COLLATE UTF8MB4_0900_AI_CI	文件列表
videos	text	视频列表
importance	VARCHAR(10) CHARACTER SET UTF8MB4 COLLATE UTF8MB4_0900_AI_CI DEFAULT NULL	知识点的“重要性”
knowledge	VARCHAR(255) CHARACTER SET UTF8MB4 COLLATE UTF8MB4_0900_AI_CI DEFAULT NULL	前序知识点
route	VARCHAR(100) CHARACTER SET UTF8MB4 COLLATE UTF8MB4_0900_AI_CI DEFAULT NULL	学习路径

在上表中 content_file 字段用于存储富文本内容文件在后端的存储地址；files 字段用于存储文件的 URL；videos 字段用于存储视频的 URL；importance 字段用于存储知识点的“重要性”属性；knowledge 字段用于存储知识点的前序知识点的节点 id；route 字段用于存储该知识点在学习路径中的位置信息。

3.4.2 技术实现

在弹窗组件的架构上，使用的是 Element Plus 库中的<el-popover>组件来实现浮动弹窗，通过设置 v-model:visible 属性来控制显隐状态。

当节点被点击时加载关联资源，如 loadNodeContent、loadFilesForNode、loadVideosForNode，和 Vue 响应式变量，如 currentFiles、currentVideos，来驱动视图更新。

通过集成@vueup/vue-quill 库，来实现富文本编辑的功能。

文件管理部分，上传文件时会校验文件的类型与命名规范，仅允许中文、英文、数字及下划线，防止文件存储出现问题，再将文件的 URL 等一系列的数据与当前的节点 id 一同传入后端进行后续处理。浏览时通过调用 pdf.js 库来实现浏览器内的 PDF 文件预览，和通过微软 Office Online 服务在线渲染 Word 和 PPT 文件以实现浏览。

视频管理中是使用 Element Plus 库里的<el-upload>组件调用后端接口，同文件资源的处理方式将视频存入后端，再返回视频 URL 与缩略图存入数据库中。

在禁用编辑，学生视角下查看前序知识点时采用的是遍历所有节点的 knowledge 属

性的方式，来匹配是否等于当前知识点的节点 id。

多模态资源关联模块中所使用的函数如表 3-5 和表 3-6 所示。

表 3-5 多模态资源关联模块函数列表

Table 3-5 List of multimodal resource association module functions

函数名	参数	函数参数说明	函数说明
handleImportanceChange	value	“重要性”值	选择知识点的“重要性”
loadNodeImport	nodeId	节点id	加载知识点的“重要性”
imageHandler	无	无	插入图片到富文本框
handleManualSave	无	无	手动保存富文本内容
loadNodeContent	nodeId	节点id	加载节点富文本内容
handleFileUpload	response	响应数据	处理文件上传
saveFilesToNode	无	无	保存文件到当前节点
loadFilesForNode	nodeId	节点id	加载节点文件
removeFile	file	文件对象	删除文件
handleFileClick	file	文件对象	查看文件
handleVideoUpload	response	响应数据	视频上传
removeVideo	video	视频对象	删除视频
saveVideosToNode	无	无	保存视频到当前节点
loadVideosForNode	nodeId	节点id	初始化加载视频
handleKnowledgeButton	无	无	编辑前序知识点
viewPreviousKnowledge	无	无	查看前序知识点
handleRouteButton	无	无	编辑学习路径
updateNodeTopic	nodeId, newTopic	节点id和节点名称	更新知识点名称
updateNodeKnowledge	nodeId, knowledgeId	节点id和前知识点id	更新前序知识点
updateNodeRoute	nodeId, Route	节点id和学习路径节点id	更新学习路径

3.5 后端存储模块

3.5.1 技术实现

添加节点时,在数据库中创建一条新的空白的数据,将当前选定的节点的节点 id 存入新节点的 `parent_id` 下,再将前端生成的节点 id 和节点名称存入对应的位置中。删除节点时根据当前选定的节点的节点 id 查询数据库中 `node_id` 或 `parent_id` 相同的数据,然后删除所有匹配的点。

导入思维导图时,将前端生成标准 `node_array` 的 `jsmind` 节点结构,进行全量更新数据库。导出时遍历数据库数据然后传入前端函数中进行处理。



图 3-11 富文本框内容存储过程

Figure 3-11 Rich Text Box Content Storage Procedure

如图 3-9 所示,富文本内容以 HTML 文件形式存储在后端 `contents` 目录下,数据库仅记录文件路径。其中的图片则通过正则表达式提取内容中的图片 URL 进行存储,同时能够自动清理未引用的旧图片。以文件形式存储的方式来避免当富文本内容的内容过大时触碰到数据库中单条记录的存储上限,并且读取时直接返回 HTML 文件内容,避免数据库大字段查询性能损耗。

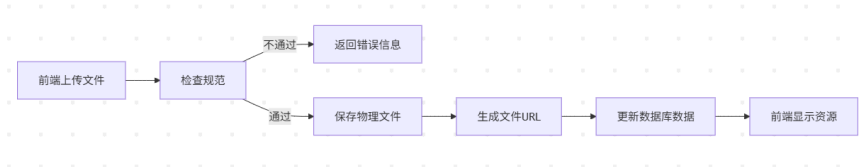


图 3-12 文件及视频存储示意图

Figure 3-12 Schematic diagram of file and video storage

文件及视频存储过程如图 3-10 所示,采用 `latin1` 中转+前端 `decodeURIComponent` 解码的方式来解决前端显示文件名时出现的中文文件名乱码问题。将文件对象数组序列化为字符串存储在数据库中,单条数据同时包含 `url`、`name`、`type`、`size` 等元数据,在删除文件时能够同步清理数据库记录和物理文件。

利用 FFmpeg 在视频第 1 秒截取关键帧，规格化为 320x180 的 JPG 图片，再将视频文件与缩略图分别存储在 uploads/videos 和 uploads/thumbnails 目录下。

共设计了如下的服务 API 端点：

- (1) /api/mindmap/export (GET): 用于导出思维导图
- (2) /api/mindmap/flowchart-data (GET): 用于转换成流程图
- (3) /api/mindmap/:nodeId (GET): 用于获取单个节点
- (4) /api/mindmap (GET): 用于获取所有节点
- (5) /api/mindmap (POST): 添加节点
- (6) /api/mindmap/:id (DELETE): 删除节点
- (7) /api/mindmap/replace (POST): 导入知识图谱
- (8) /api/mindmap/update (POST): 更新知识点名称
- (9) /api/mindmap/update/importance (POST): 更新知识点的“重要性”
- (10) /api/mindmap/update/knowledge (POST): 更新节点前序知识点字段
- (11) /api/mindmap/update/route (POST): 更新学习路径
- (12) /api/mindmap/save-content (POST): 保存富文本内容到文件
- (13) /api/mindmap/get-content/:nodeId (GET): 读取知识点的富文本内容文件
- (14) /api/mindmap/batch-get-content (POST): 搜索时批量读取富文本内容文件
- (15) /api/upload/image (POST): 图片上传
- (16) /api/upload/video (POST): 视频上传
- (17) /api/mindmap/update/videos (POST): 更新知识点视频列表
- (18) /api/mindmap/videos/:nodeId (GET): 获取知识点视频列表
- (19) /api/mindmap/videos/:nodeId/:videoId (DELETE): 删除视频
- (20) /api/upload/file (POST): 文件上传
- (21) /api/mindmap/update/files (POST): 更新文件列表
- (22) /api/mindmap/files/:nodeId (GET): 获取知识点文件列表
- (23) /api/mindmap/files/:nodeId/:fileName (DELETE): 删除文件

3.6 本章小结

本章主要介绍了课程知识图谱系统中各个模块的功能设计及技术实现。

第 4 章 系统功能展示

4.1 知识图谱可视化界面功能展示

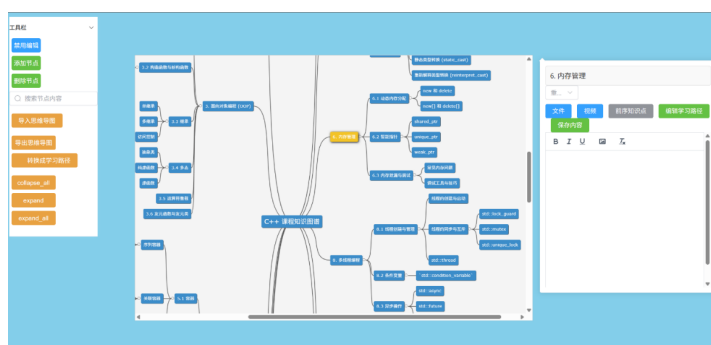


图 4-1 系统整体界面展示

Figure 4-1 Overall System Interface Display

如图 4-1 所示，本知识图谱系统的网页界面是由三部分组成的，最左侧是知识图谱操作模块、中间部分是知识图谱可视化界面、右侧是多模态资源关联模块。

如图 4-2 所示，当点击知识图谱中的节点时，右边会出现弹出框，弹出框里最上方是该知识点的名称，名称下方是当前知识点的“重要性”。之后一排是文件资源、视频资源按钮和查看前序知识点按钮和编辑学习路径按钮

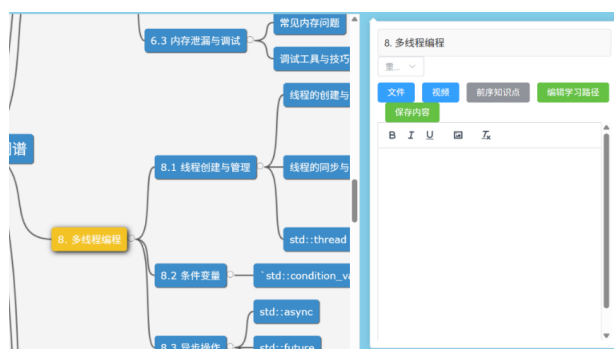


图 4-2 点击节点交互展示

Figure 4-2 Click Node Interaction Display

如图 4-3 所示，双击节点时能够直接修改知识点的名称。

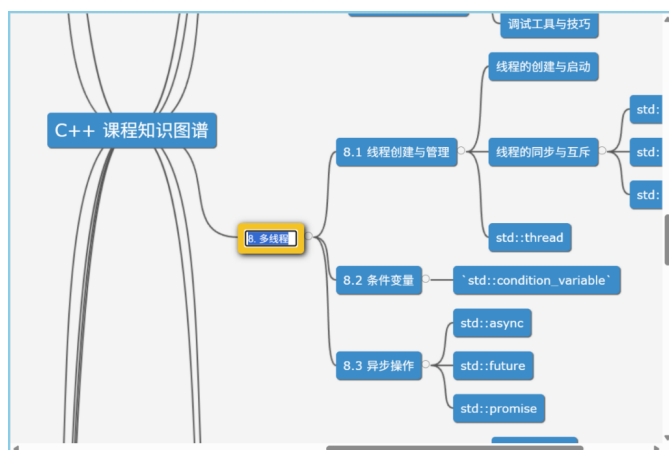


图 4-3 双击节点交互展示

Figure 4-3 Double click Node Interaction Display

按住 **ctrl** 再鼠标进行滚轮时可以缩放视图大小，如图 4-4 所示。可以用鼠标拖动画布。

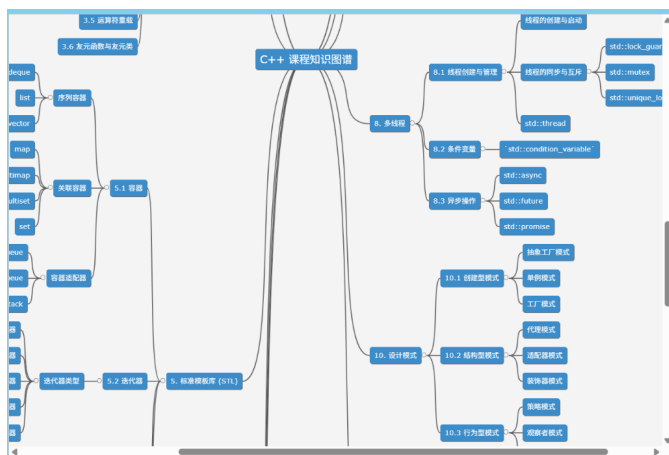


图 4-4 缩放视图展示

Figure 4-4 Zoom View Display

4.2 知识图谱操作模块功能实现

如图 4-5 所示，左边的工具栏中有着编辑权限的开关，添加节点、删除节点按钮，搜索节点名称和富文本内容的搜索框，导入思维导图和导出思维导图按钮，转换成学习路径按钮和与一键展开收拢知识图谱所有节点功能的按钮。

当想要手动添加节点时，需要先用鼠标点击知识图谱内想要添加的节点，然后再点击左边工具栏中的添加节点按钮。新的节点会自动生成在选定的节点的后方，同时数据库中也会存入新节点。

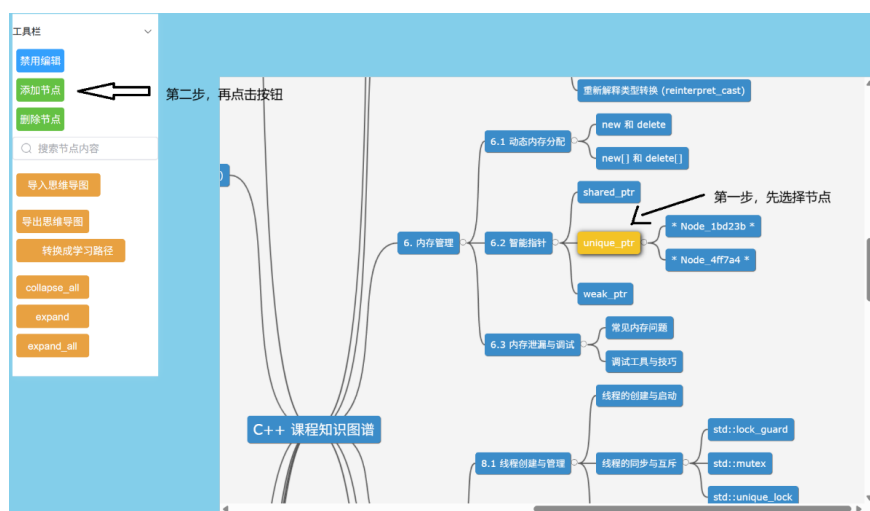


图 4-5 添加节点展示

Figure 4-5 Add Node Display

如图 4-6 所示，当想要手动删除某个节点时，需要先用鼠标点击知识图谱内的想要删除的节点，后再点击左边工具栏中的删除节点按钮，节点就会自动删除，同时对数据库中的数据也会删除。

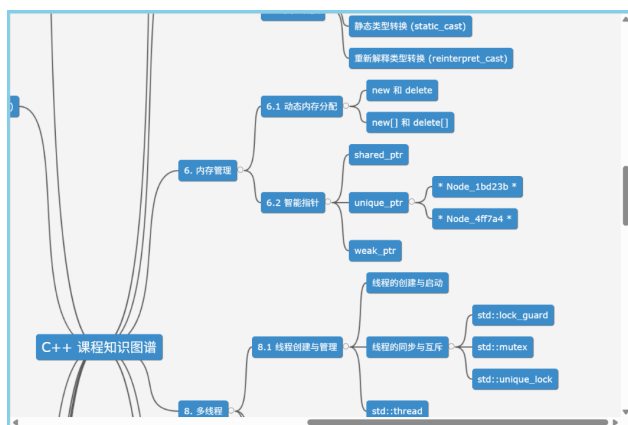


图 4-6 删除节点展示

Figure 4-6 Display of deleted nodes

如图 4-7 所示，在搜索栏中进行搜索时支持同时搜索知识点标题和知识点中的内容来进行匹配，以此来方便师生快速找到想要的内容。

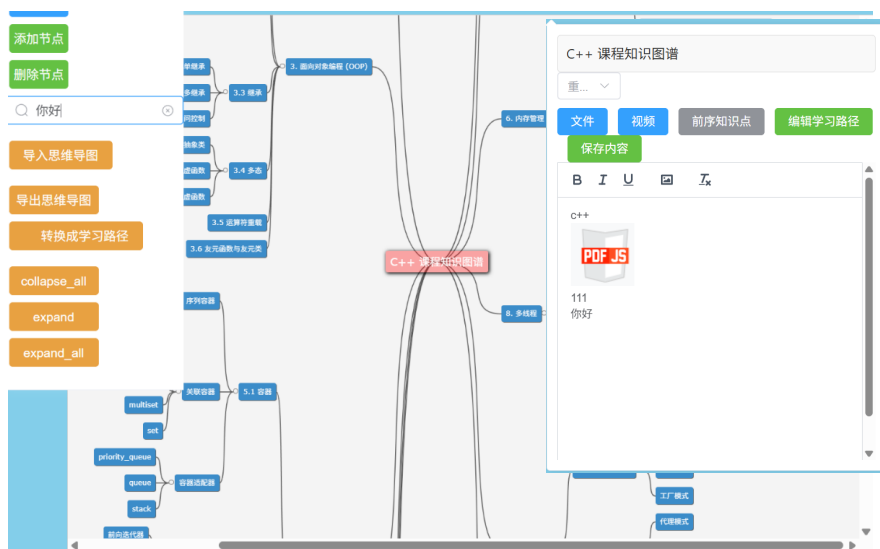


图 4-7 搜索展示

Figure 4-7 Search Display

点按导入思维导图按钮，再从电脑上选择想要导入的思维导图 markdown 文件即可一键完成思维导图导入。点按导出思维导图按钮，同样能够一键导出当前展示的思维导图

图的 markdown 形式。

点按转换成学习路径按钮就能自动生成默认的学习路径，并且能够自动进行弯折，学习路径中的节点具有和知识图谱中的节点一致的功能，同样能够进行点击交互，如图 4-8 所示。

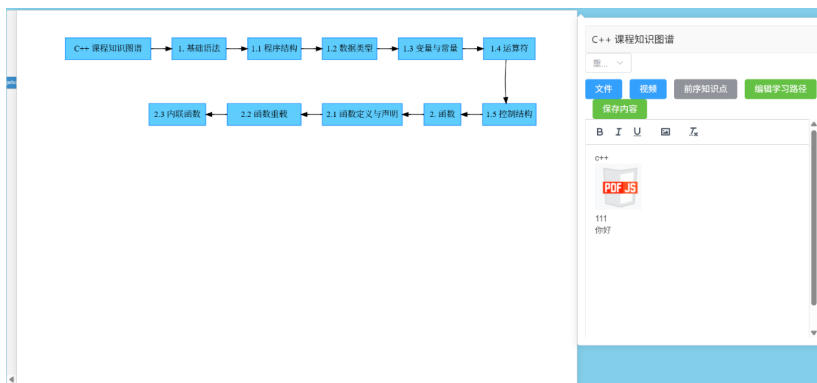


图 4-8 生成学习路径展示

Figure 4-8 Display of generated learning path

如图 4-9 所示，支持一键收拢所有节点和分层次展开所有节点，以及单独展开某个节点。

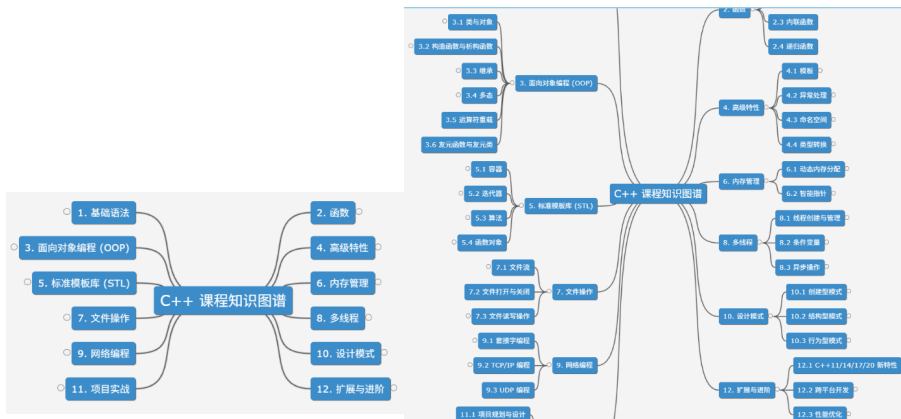


图 4-9 知识图谱收拢、层次展开展示

Figure 4-9 Knowledge Graph Collapsed and Hierarchical Unfolding Display

4.3 多模态资源关联模块功能实现

在点击了知识节点后的弹出框中可进行多模态的资源关联。

可以在知识点的标题下方设置该知识点的“重要性”：必学、选学和可学。选择后在知识图谱中就会呈现不同的颜色进行标识，如图 4-10 所示。



图 4-10 知识点的“重要性”效果展示

Figure 4-10 Display of Importance Effect of Knowledge Points

如图 4-11 所示，在点击了文件按钮后，弹出框就会变成文件列表视图，此时就可以对该节点进行上传文件、删除文件，也能点击文件进行查看文件，如图 4-12 所示。



图 4-11 文件功能展示

Figure 4-11 File Function Display

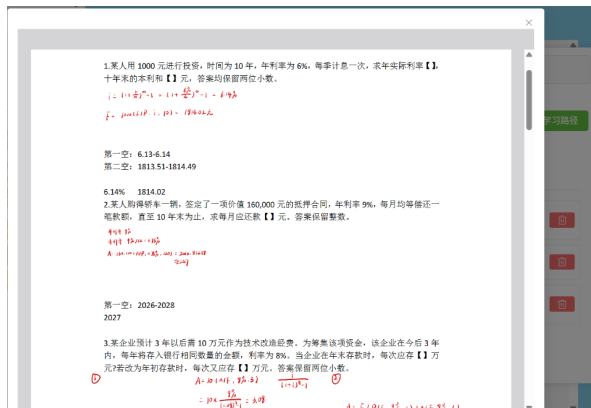


图 4-12 文件浏览展示

Figure 4-12 File browsing display

如图 4-13 所示，点击视频按钮后，弹出框就会变成视频列表视图，此时就可以对该节点进行上传视频、删除视频，也能够播放视频。点击上传视频按钮再选择想要上传的视频文件就能添加视频到知识点中了。点按删除键就能移除视频。



图 4-13 视频浏览展示

Figure 4-13 Video browsing display

点击前序知识点按钮后再点击知识图谱中的节点即可完成对该知识点的前序知识点的添加。此时进入查看视图即点按了禁用编辑按钮后再点击前序知识点按钮，会在知识图谱中将之前添加的前序知识点进行变色处理，如图 4-14 所示。

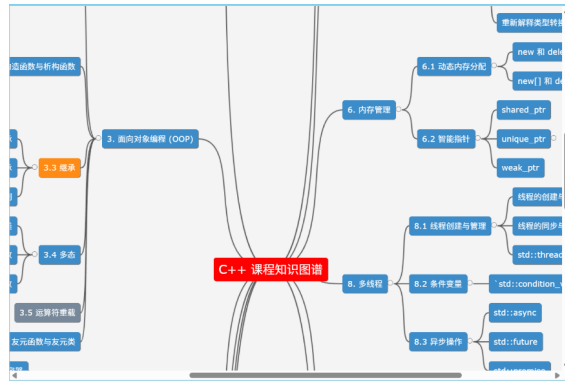


图 4-14 前序知识点查看展示

Figure 4-14 Preface Knowledge Point View Display

点击编辑学习路径按钮后再点击知识图谱中的节点即可完成对学习路径的编辑。同时设置了当漏选节点时出现的断链情况,通过同时显示两条学习路径来快速定位哪个知识点编辑学习路径时编辑错了,如图 4-15 所示。

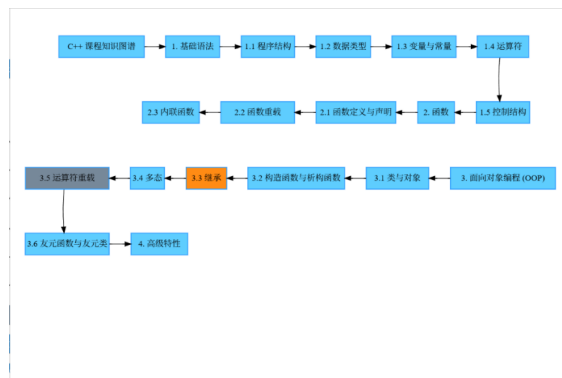


图 4-15 编辑学习路径展示

Figure 4-15 Display of Editing Learning Path

4.4 本章小结

本章主要介绍了课程知识图谱系统项目中前端操作的三个模块所具有的功能以及相关的操作方式。

第五章 总 结

本研究成功设计并实现了课程知识图谱系统，通过结构化知识建模与动态可视化，显著降低知识管理的复杂度，助力师生快速构建与理解课程体系。同时打破了传统资源孤岛，实现知识点与多样化教学材料的深度关联，为个性化学习提供数据基础。未来可通过引入自然语言处理技术实现知识自动抽取，结合推荐算法优化资源推送策略，进一步提升系统的智能化与实用性。

当前系统还存在着一些缺陷，例如学习路径上，目前只设定了出现双链的情况，只能手动复制相关代码来增加能够存在的学习路径上限，以及学习路径的布局上还欠缺优化。富文本编辑上还有提升的空间，支持的功能还不够多，不能完全实现 markdown 编辑。想要使用微软 Office Online 服务需要进行内网穿透^[28-29]，达到可以在公网访问系统界面，从而查看 Word 文件和 PPT 文件。以及为系统添加登入功能^[30]，区分学生账号和教师账号。

参 考 文 献

- [1] 熊言林, 周倩. 导教图: 一种思维疏导的教学设计工具[J]. 化学教育, 2013, 34(9): 37-41.
- [2] Lehmann J, José Varzinczak I, Bundy A. Reasoning with context in the semantic web[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2012, 12: 1-2.
- [3] Jiang K, Wu D, Jiang H. FreebaseQA: A new factoid QA data set matching trivia-style question answer pairs with Freebase[C]. //Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics. Minneapolis, Minnesota: Human Language Technologies, 2019: 318-323.
- [4] Han J, Pei J, Tong H. Data Mining: Concepts and Techniques[M]. San Francisco: Morgan Kaufmann, 2022: 55-70.
- [5] Peng S, Cao L, Zhou Y, et al. A survey on deep learning for textual emotion analysis in social networks[J]. Digital Communications and Networks, 2022, 8(5): 745-762.
- [6] Sadeeq M J, Zeebaree S R M. Semantic Search Engine Optimisation (SSEO) for dynamic websites: A review[J]. International Journal of Science and Business, 2021, 5(3): 148-158.
- [7] 《中国智慧教育蓝皮书》: 智慧教育是数字时代的教育新形态[EB/OL]. (2023-02-13). [2025-05-30]. http://www.moe.gov.cn/jyb_xwfb/xw_zt/moe_357/2023/2023_zt01/mtbd/202302/t20230214_1044497.html.
- [8] 教育部关于发布智慧教育平台系列两项教育行业标准的通知[EB/OL]. (2022-11-30). [2025-05-30]. https://www.gov.cn/zhengce/zhengceku/2023-02/15/content_5741574.htm.
- [9] 数字化引领教育变革新风向——一年来国家教育数字化战略行动发展观察[EB/OL]. (2024-01-27). [2025-05-30]. http://www.moe.gov.cn/jyb_xwfb/s5147/202401/t20240129_1113155.html.
- [10] 顾亚丽, 曾志宏. 融合人工智能图像特征的知识图谱表示学习研究[J]. 兰州文理学院学报, 2024, 38(01): 44-49.
- [11] Shi H, Liu X, Shi G, et al. Research on medical automatic Question answering model based on knowledge graph[C]. //2023 35th Chinese Control and Decision Conference (CCDC). Dalian, China: IEEE, 2023: 1778-1782.
- [12] 李振, 周东岱. 教育知识图谱的概念模型与构建方法研究[J]. 电化教育研究, 2019,

删除了:,

删除了:

40(08): 78 - 86.

- [13] 李艳燕, 张香玲, 李新, 等. 面向智慧教育的学科知识图谱构建与创新应用[J]. 电化教育研究, 2019, 40(08): 60 - 69.
- [14] 钟卓, 唐烨伟, 钟绍春, 等. 人工智能支持下教育知识图谱模型构建研究[J]. 电化教育研究, 2020, 41(04): 62 - 70.
- [15] 黄焕, 元帅, 何婷婷, 等. 面向适应性学习系统的课程知识图谱构建研究——以“Java 程序设计基础”课程为例[J]. 现代教育技术, 2019, 29(12): 89 - 95.
- [16] Jiang T, Zhao T, Qin B, et al. The role of “condition” a novel scientific knowledge graph representation and construction model [C]. //Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. NY: Association for Computing Machinery, 2019: 1634-1642.
- [17] Ye Z, Yuting J, Luoyi F U, et al. AceMap academic map and AceKG academic knowledge graph for academic data visualization[J]. Journal of Shanghai Jiaotong University, 2018, 52: 1357-1362.
- [18] Jones A, Bomash I. Validating mastery learning: assessing the impact of adaptive learning objective mastery in Knewton Alta[C]. //Artificial Intelligence in Education: 19th International Conference, AIED 2018. London, UK: Springer International Publishing, 2018: 433-437.
- [19] Alshalabi I A, Hamada S, Elleithy K. Automated adaptive learning using smart shortest path algorithm for course units[C]. //2015 Long Island Systems, Applications and Technology. NY: IEEE, 2015: 1-5.
- [20] Jackowska-Strumiłło L, Bieniecki W, Saad M B. A web system for assessment of students' knowledge[C]. //2015 8th International Conference on Human System Interaction (HSI). NY: IEEE, 2015: 20-26.
- [21] 聂富鹏. 视频转码传输系统的设计与实现[D]. 西安: 西安电子科技大学, 2014.
- [22] 马敬奇, 雷欢, 陈敏翼. 基于 AlphaPose 优化模型的老人跌倒行为检测算法[J]. 计算机应用, 2022, 42(01): 294-301.
- [23] Van Der Aalst W. Process mining: Overview and opportunities[J]. ACM Transactions on Management Information Systems (TMIS), 2012, 3(2): 1-17.
- [24] 陆凌牛. HTML 5 与 CSS 3 权威指南[M]. 北京: 机械工业出版社, 2014: 271-272.
- [25] PDF Reference. Adobe Portable Document Format[EB/OL]. (2021-10-26). [2025-05-30]. <https://helpx.adobe.com/tw/incopy/using/pdf.html>.

- [26] Uzayr S B. Mastering Vue.js: A Beginner's Guide[M]. Taylor & Francis Group, 2022: 37-43.
- [27] 陈光柳, 王洁, 王博, 等. 一种基于 axios 的封装注册全局拦截方法及装置[P]. 北京: CN112905160A. 2021-06-04.
- [28] 黄瀚宇, 陈焯辉, 肖梓勤, 等. 基于深度学习的手写数字图像识别模型研究及其浏览器服务平台搭建[J]. 科学技术创新, 2022(01): 97-100.
- [29] 韩星越. 基于树莓派的内网穿透技术研究与应用[J]. 网络安全技术与应用. 2021(10): 22-25.
- [30] 陆江东. 面向用户登陆管理模块的 Web 前端系统设计[J]. 信息技术, 2018(01): 144-148.

致 谢

感谢钱衍翰班长在大学期间对我无论是生活上还是学习上各种问题的耐心解答，感谢通信工程 2104 班的大家，没有大家的热心就没有班级其乐融融的氛围。感谢方老师在我毕业设计的过程中的悉心指导，感谢在论文写作过程中提供的帮助。感谢周围的每一个人。