律镜

2023-12-3

系统设计报告

AI+法律领域垂直搜索引擎

《软件工程管理》G04 小组

组长：徐毕颖

组员：章越、沈书豪、黄文杰、王梓轩

目录

系统设计报告

1.1 编写目的

本文档是2023年秋冬学期软件工程管理G04小组的系统设计报告，基于《项目计划书》和《需求规格说明书》进一步阐述系统结构、各项功能的实现方式和数据库设计方法等内容，以便为软件系统的开发提供清晰的方向。

本文档致力于阐明系统的整体结构，并为每个功能点和模块的开发提供具体指导。依照系统设计报告，团队成员能够更好地协同合作，确保整个系统在开发阶段能够高效而无缝地集成各个模块。

另外，该报告还旨在为软件系统的维护提供参考。通过详细描述各个模块的外部接口、内部接口以及用户接口，报告有助于未来维护人员更容易理解系统的结构和功能，从而更快速、更有效地进行维护和更新。

1.2 项目背景

随着法律领域的不断发展和法规的增多，对法律信息的准确、高效获取成为了法律从业者和公众的迫切需求。然而，传统搜索引擎在满足法律专业性和深度搜索需求方面存在一定的局限性，对于法律领域的从业者与学生来说，其检索结果的覆盖度与专业性都很难满足需求，而对于一般群众而言，受限于专业知识的匮乏以及对专业名词的陌生，也较难在关键词为导向的搜索引擎中获取权威的解答。因此，我们小组决定开发一款面向法律领域的垂直搜索引擎，并通过AI赋能搜索，旨在提供专门针对法律领域的深度、智能搜索服务。

1.3 术语与缩写解释

1. Elastic Search

Elasticsearch是一个开源的分布式搜索和分析引擎，用于在大规模数据集中进行快速、实时的搜索和分析。它基于Apache Lucene搜索引擎构建，通过提供简单的RESTful API和丰富的查询语言，使用户能够轻松地在各种类型的数据中执行复杂的搜索和分析操作。Elasticsearch被广泛用于构建实时应用程序，如日志分析、全文搜索、业务分析等，因其高性能、可扩展性和容错性而备受欢迎。

2. 垂直搜索引擎

垂直搜索引擎是一种专注于特定主题或领域的搜索引擎。与通用搜索引擎（如Google或Bing）不同，垂直搜索引擎致力于提供深度而专业化的搜索结果，以满足特定领域的用户需求。这些领域可以涵盖特定行业、主题、媒体类型或其他特定的信息范畴。通过专注于特定领域，垂直搜索引擎可以提供更准确、有针对性的搜索结果，帮助用户更有效地获取他们所需领域的信息。

3. 爬虫

爬虫（Spider）是一种程序或脚本，被设计用来自动地在互联网上获取信息。爬虫的主要任务是通过遍历网络上的页面并提取有用的数据，以便后续的分析、索引或展示。爬虫通常按照一定的规则和算法遍历网页，跟随链接，然后从页面中提取感兴趣的信息，如文本内容、链接、图像等。

爬虫在各种应用中都有广泛的用途，包括搜索引擎的索引过程、数据挖掘、舆情监测、价格比较、信息聚合等。然而，爬虫的使用也面临一些道德和法律问题，因为它们可能会对网站的性能产生负面影响，甚至触发反爬虫机制。因此，在使用爬虫时需要遵循网站的规定并尊重网络礼仪。

4. 知识图谱

知识图谱是一种用于表示、组织和推理知识的图状数据结构。它是一种语义网络，通过节点和边的连接关系表示实体之间的语义关联。知识图谱旨在捕捉现实世界中的实体和它们之间的关系，以便计算机能够更深入地理解和处理信息。

1.4 参考资料

《软件工程开发国家标准》

《软件工程项目开发文档范例》

《软件需求（第三版）》

《软件工程——实践者的研究方法》

1. 需求概述

2.1 功能需求

a. **搜索功能：**

- 用户能够通过关键字搜索法律领域的相关信息，包括法律条文、案例、法规等。

b. **高级搜索：**

- 提供高级搜索选项，包括时间范围、地区、法院等筛选条件，以满足用户更精确的检索需求。

c. **模糊搜索：**

- 实现模糊搜索功能，允许用户在输入关键字时考虑拼写错误或近义词，提高搜索结果的覆盖性。

d. **拼音搜索：**

- 支持拼音搜索，使用户可以使用拼音检索法律条文、案例等内容。

e. **数据分类与标签：**

- 将搜索结果进行分类与标签化，方便用户快速定位所需信息，提高检索效率。

f. **智能法律建议：**

- 基于用户搜索内容，使用AI助手为用户提供智能建议，解释法律问题和提供法律意见。

g. **首页推荐：**

- 基于当前热点和用户搜索历史记录为用户推送相关案例、法律等内容。

h. **知识图谱：**

- 将相关的法律内容联系起来，构建成一张知识图谱网，帮助用户更高效地搜索。

2.2 性能需求

a. **系统响应时间：**

- 律镜法律领域垂直搜索引擎对用户请求的响应时间应控制在2秒以内，以确保用户体验流畅。
- 高并发情况下，系统的响应时间不应超过5秒，以保证在用户访问量增加时仍能提供稳定的服务。

b. **并发用户量：**

- 系统应能支持最少1000名并发用户同时使用，以满足大量用户同时访问的需求。
- 在极端情况下，系统应有扩展性，支持最少2000名并发用户，以防止因用户数量激增而导致的系统性能下降。

c. **吞吐量：**

- 律镜法律领域垂直搜索引擎的吞吐量应能够处理每秒至少100个请求，以确保高效的信息检索服务。
- 在高峰期，系统吞吐量需达到每秒至少200个请求，以满足用户在特定时间段的集中查询需求。

d. **系统稳定性：**

- 系统运行连续时间应能够达到每月99.9%的可用性，确保系统在绝大多数时间内可供用户访问。
- 系统应具备快速恢复的能力，故障发生后系统可在30分钟内恢复正常运行。

e. **数据安全性：**

- 律镜法律领域垂直搜索引擎应采用加密技术，确保用户的检索数据传输过程中的机密性。
- 用户数据在系统中的存储应采取安全措施，防止未经授权的访问或数据泄露。

f. **系统日志：**

- 系统应记录用户的检索历史和操作日志，以支持系统运行状态的监控和故障排除。
- 日志记录应包括关键操作、异常事件、以及系统性能指标等信息，以便进行系统性能分析和优化。

g. **可扩展性：**

- 律镜法律领域垂直搜索引擎应具备良好的可扩展性，能够灵活地应对未来业务规模的扩大。
- 在系统升级或功能拓展时，应保证系统的稳定性，不影响已有功能的正常使用。

以上性能需求将确保律镜法律领域垂直搜索引擎在使用过程中能够高效、稳定地提供服务，满足用户的期望和需求。

2.3 安全需求

a. **用户隐私保护：**

- 系统将遵循《个人信息保护法》等相关法律法规，不得未经用户授权收集、使用、存储用户个人信息。
- 实施匿名化策略，将用户个人身份信息与搜索记录分离，确保用户隐私得到有效保护。

b. **防御性编码：**

- 在系统开发过程中采用防御性编码，包括输入验证、输出编码和安全配置等，以防范潜在的安全漏洞。
- 定期进行代码审查和安全漏洞扫描，及时修复潜在的安全问题。

c. **数据加密传输：**

- 采用安全的传输层协议（如TLS/SSL），对用户与系统之间的数据传输进行加密，保障信息在传输过程中的机密性。
- 确保数据库中的敏感信息（如用户密码）存储时采用适当的加密算法，防止敏感数据泄露。

d. **防护性需求详细措施：**

- 反垃圾机制：
  - 实施机器学习算法，检测搜索行为中的异常模式，防止恶意爬虫和刷点击等行为。
  - 设定阈值和规则，对搜索请求进行监控和过滤，及时发现异常行为。
- 验证码防恶意使用：
  - 引入验证码机制，特别是在用户频繁请求、操作时，要求用户输入验证码，有效防止爬虫等恶意使用。
- 用户身份认证：
  - 实施严格的用户身份认证机制，采用多因素认证，确保只有授权用户能够访问敏感信息和功能。
  - 引入双因素认证，如手机验证码、指纹识别等，提高身份验证的安全性。
- 安全审计日志：
  - 记录安全审计日志，包括用户登录、搜索记录等，以便在发生安全事件时进行追踪和分析。
- 定期安全培训：
  - 为系统开发人员、运维人员提供定期的安全培训，提高团队对安全问题的敏感性和应对能力。

这些详细措施将有助于确保律镜法律领域垂直搜索引擎在安全性方面具备高水平的保障，保护用户隐私，预防潜在的安全威胁。

2.4 可维护性需求

作为一个成熟的系统，在开发初期就应该充分考虑系统的可维护性。 对此，有以下几点要求：

a. 可读性。系统的代码必须易于阅读，理解和修改。使用一致的命名规则，注释和代码结构可以帮助代码更易于维护。

b. 可扩展性。系统的架构和设计必须支持新增功能的无缝集成。将系统模块化和组件化可以使系统更易于扩展。

c. 可测试性。系统的每个功能都必须可以被独立测试，测试用例和测试结果必须得到记录。这可以保证系统稳定性和可靠性。

d. 可维护性。系统的代码必须保持简洁，干净和有组织。使用版本控制和代码审查可以帮助开发人员快速定位问题和进行维护。

e. 文档化。系统的架构，设计，代码和运维过程必须都有文档化记录。这有助于更好地理解系统和快速定位问题。

f. 可配置性。系统的配置参数和环境变量必须可配置，这可以帮助快速部署和升级系统。同时，应提供易于管理的配置管理界面，方便对系统配置进行管理。

通过遵循上述可维护性需求，可以确保系统在开发后易于维护和修改。这对于长期维护和持续迭代系统至关重要 。

2. 总体设计

3.1 基本设计概念和流程处理

3.1.1 基本设计概念

我们的法律领域垂直搜索引擎 "律镜" 以以下基本设计概念为核心：

i. 全面性搜索：“律镜"致力于提供全面的法律信息搜索服务，覆盖法规、案例、法学文献等多种内容，满足用户广泛的法律信息需求。

ii. 智能搜索算法：引入先进的自然语言处理和机器学习技术，通过分析用户输入，理解搜索意图，提供高效、智能的搜索结果。

iii. 用户友好界面：设计直观、易用的用户界面，支持关键词搜索、类型选择、高级搜索等功能，使用户能够轻松、快速地获取所需法律信息。

iv. 数据可视化：提供数据可视化工具，通过知识图谱的形式帮用户分析该次搜索的相关词云，帮助用户更直观地理解和分析法律数据，更加快捷地搜索其他相关信息。

v. 用户反馈机制：提供用户反馈功能，以持续改进搜索算法和系统性能，确保用户体验的不断优化。

3.1.2 流程处理

用户搜索流程：

i. 用户输入：用户通过搜索框输入法律相关关键字或问题。

ii. 搜索词条模块处理：

- 关键词搜索： 用户输入一个或多个关键词，系统进行智能分析。
- 搜索类型选择： 用户选择搜索类型，如全文搜索、标题搜索等。
- 高级搜索： 用户进行高级搜索，限定时间范围、模糊匹配等。

iii. 搜索结果模块处理：

- 搜索结果排序： 根据相关性对结果排序，权衡用户输入的关键词、搜索方式等。
- 结果摘要： 生成摘要，展示相关文档片段，方便用户快速了解内容。
- 用户反馈： 允许用户对结果进行反馈，促进系统持续改进。

数据更新流程：

i. 数据源监控：实时监控法律数据源，检测数据变更或新数据发布。

ii. 搜索引擎索引更新：

- 数据同步： 定期同步数据，保持系统数据库与法律数据源一致。
- 数据清洗： 对同步数据进行清洗和格式化，确保数据质量。
- 索引更新： 更新搜索索引，保障搜索引擎准确性和效率。

系统监控流程：

i. 运行监控：实施系统运行监控，监测服务器状态、响应时间等关键指标。

ii. 异常检测：使用异常检测工具，及时发现系统异常，包括服务崩溃、高负载等。

iii. 警报通知：发现异常情况时，系统自动发送警报通知，通知运维团队。

iv. 自动修复：实施自动修复机制，对一些常见问题进行自动修复，降低系统停机时间。

AI 辅助模块处理：

i. AI 模块调用：

- 关键字提取： 通过调用相关 API 分析用户查询，提取关键字优化搜索算法。
- 文档总结： 阅读文档时，利用 AI 对搜索结果进行智能总结，生成简明扼要的摘要。

知识图谱模块处理：

i. 关键词图生成：

- 关键词图： 生成搜索结果的知识图谱，展示相关的关键词节点。
- 图谱转换与跳转搜索： 用户点击节点可将其作为中心词，触发新的搜索，提供更深层次的法律信息。

3.2 系统架构

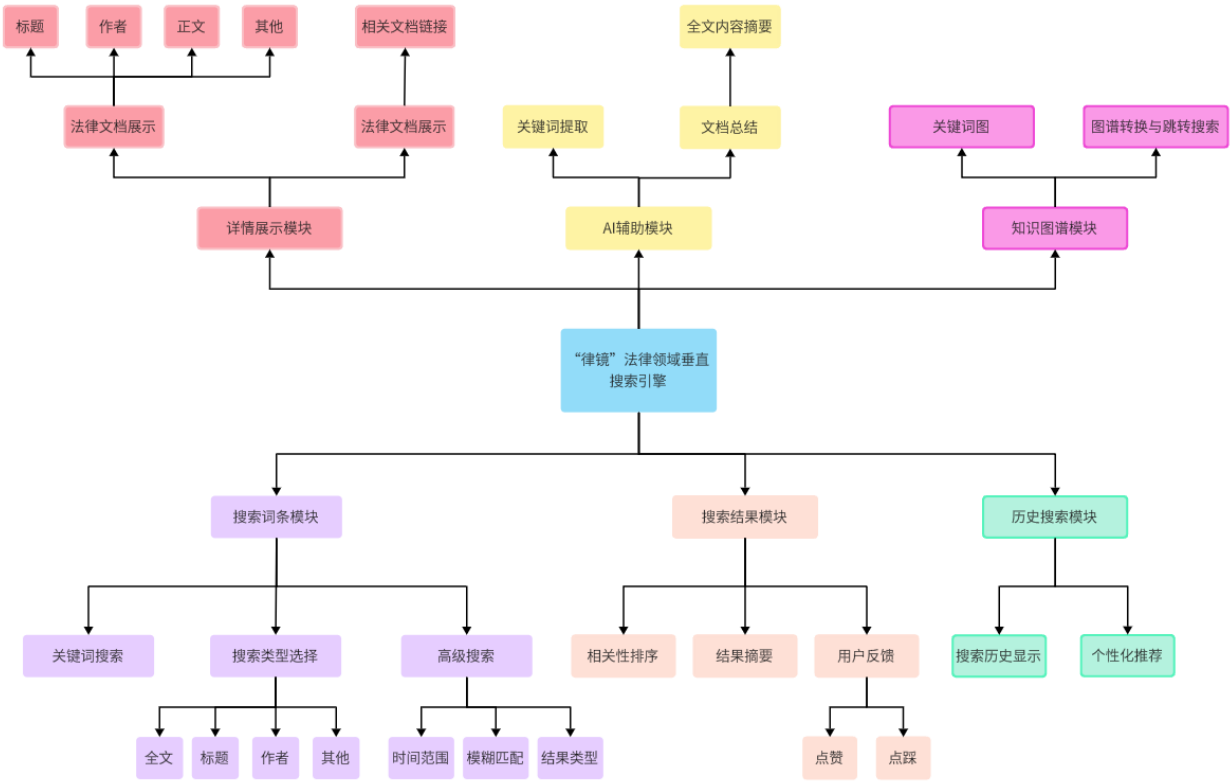

图3-2 系统模块架构图

3.3 功能设计

3.3.1 系统功能表

i. 关键词搜索

功能名称	关键词搜索	**所属模块**	搜索词条模块
相关用户/权限	所有用户可访问	**优先级**	高
功能描述	用户输入关键词或短语进行系统数据库搜索		
输入要求	用户提供关键词、短语或长文本		
处理/操作步骤	系统使用搜索算法匹配数据库中的文档		
输出结果	匹配搜索结果列表		

ii. 搜索类型选择

功能名称	搜索类型选择	所属模块	搜索词条模块
相关用户/权限	所有用户可访问	优先级	中
功能描述	用户可选择全文搜索、标题搜索等搜索类型		
输入要求	用户选择搜索类型		
处理/操作步骤	过滤或指定搜索范围和条件		
输出结果	根据指定类型的搜索结果列表		

iii. 高级搜索

功能名称	高级搜索	所属模块	搜索词条模块
相关用户/权限	所有用户可访问	优先级	中
功能描述	用户设定搜索方式，如时间范围、模糊匹配等		
输入要求	用户选择限定方式、期望的搜索结果类型		
处理/操作步骤	强化搜索算法以获得更精确的搜索结果		
输出结果	符合限定条件的搜索结果列表		

iv. 搜索结果排序

功能名称	搜索结果排序	所属模块	搜索结果模块
相关用户/权限	所有用户可访问	优先级	高
功能描述	根据相关性对搜索结果进行排序		
输入要求	用户指定排序方式		
处理/操作步骤	根据排序方式重新排列搜索结果		
输出结果	排序后的搜索结果列表		

v. 结果摘要

功能名称	结果摘要	所属模块	搜索结果模块
相关用户/权限	所有用户可访问	优先级	中
功能描述	为搜索结果生成摘要，包含相关文档片段		
输入要求	搜索结果文档内容		
处理/操作步骤	提取文档片段以生成摘要		
输出结果	每个搜索结果的摘要信息		

vi. 用户反馈

功能名称	用户反馈	所属模块	搜索结果模块
相关用户/权限	登录用户可访问	优先级	中
功能描述	记录用户对搜索结果的反馈（点赞、点踩等）		
输入要求	用户对搜索结果的反馈		
处理/操作步骤	先验证用户身份，再记录用户反馈		
输出结果	用户对搜索结果的反馈记录		

vii. 搜索历史显示

功能名称	搜索历史显示	所属模块	历史搜索模块
相关用户/权限	登录用户可访问	优先级	中
功能描述	展示用户的法律领域查询历史		
输入要求	1.用户的个人账户信息（如id） 2.用户的搜索记录		
处理/操作步骤	1.系统根据用户id搜索其对应的搜索历史 2.保存用户的搜索历史以供显示		
输出结果	用户的搜索历史记录列表		

viii. 个性化推荐

功能名称	个性化推荐	所属模块	历史搜索模块
相关用户/权限	登录用户可访问	优先级	中
功能描述	基于用户行为提供个性化推荐内容		
输入要求	用户的搜索历史、点击行为、偏好等信息		
处理/操作步骤	分析用户行为模式，生成相关推荐内容		
输出结果	个性化推荐的法律文档、案例、法规等内容列表		

ix. 法律文档展示

功能名称	法律文档展示	所属模块	详情展示模块
相关用户/权限	所有用户可访问	优先级	高
功能描述	展示选中搜索结果文档的详细内容		
输入要求	用户选中的搜索结果文档		
处理/操作步骤	根据文档内容呈现法律文档的具体内容和结构		
输出结果	选中文档的详细内容		

x. 相关文档推荐

功能名称	相关文档推荐	所属模块	详情展示模块
相关用户/权限	所有用户可访问	优先级	中
功能描述	基于当前文档内容和类型，推荐相关性较高的其他文档链接		
输入要求	当前文档的内容和类型		
处理/操作步骤	提供基于当前文档内容的其他相关文档链接		
输出结果	其他相关文档的链接列表		

xi. 关键词提取（AI 辅助）

功能名称	关键词提取（AI 辅助）	所属模块	AI 辅助模块

相关用户/权限	所有用户可访问	**优先级**	低
功能描述	通过调用 API（如chatGPT）提取用户查询中的关键词		
输入要求	用户输入的查询文本		
处理/操作步骤	利用相关 API （如chatGPT）提取查询中的关键词		
输出结果	提取的关键词列表		

xii. 文档总结（AI 辅助）

功能名称	文档总结	**所属模块**	AI 辅助模块
相关用户/权限	所有用户可访问	**优先级**	低
功能描述	使用 AI 对文档进行智能总结，生成简明扼要的摘要信息		
输入要求	用户阅读的文档内容		
处理/操作步骤	对用户阅读的文档进行智能总结		
输出结果	文档的智能总结摘要		

xiii. 生成关键词图谱

功能名称	生成关键词图谱	**所属模块**	知识图谱模块
相关用户/权限	所有用户可访问	**优先级**	中
功能描述	为每次搜索生成知识图谱形式的关键词图		
输入要求	用户输入的检索内容		
处理/操作步骤	提取关键词并构建知识图谱形式的关键词图		
输出结果	生成的关键词图谱		

xiv. 图谱转换与跳转搜索

功能名称	图谱转换与跳转搜索	**所属模块**	知识图谱模块
相关用户/权限	所有用户可访问	**优先级**	中

功能描述	点击词图中的节点进行转换为中心词，并允许以该节点为输入进行新的搜索
输入要求	点击的关键词图谱中的节点
处理/操作步骤	将点击的节点转换为中心词或进行动态转换
输出结果	转换后的中心词或新搜索节点的搜索结果列表

3.3.2 功能IPO图

i. 关键词搜索

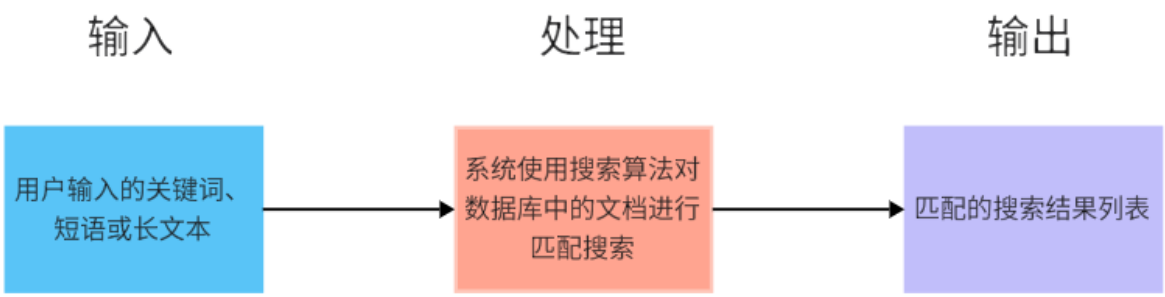

图3-3-2-1 关键词搜索

ii. 搜索类型选择

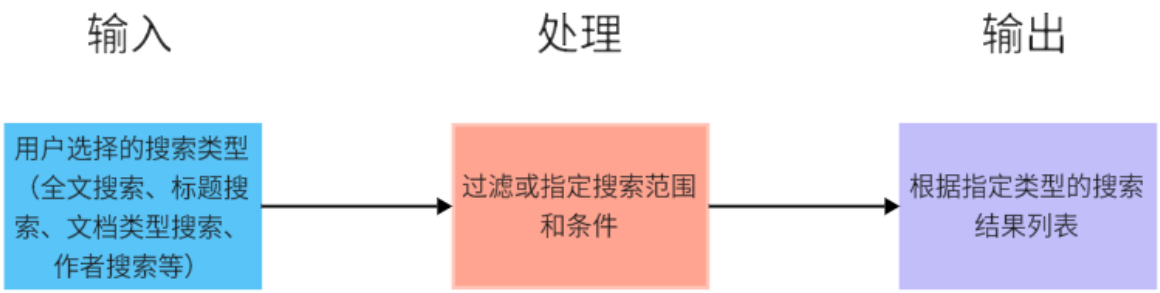

图3-3-2-1 搜索类型选择

iii. 高级搜索

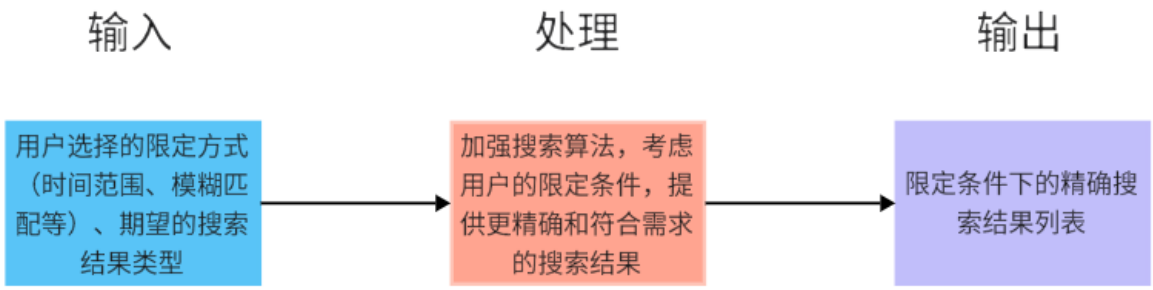

图3-3-2-3 高级搜索

iv. 搜索结果排序

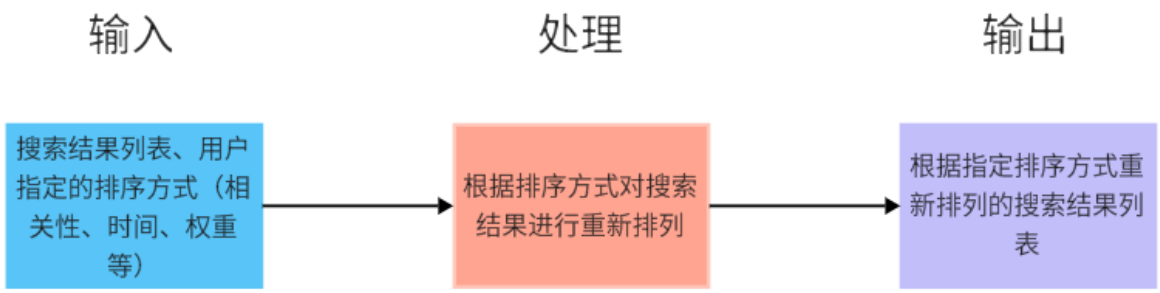

图3-3-2-4 搜索结果排序

v. 结果摘要

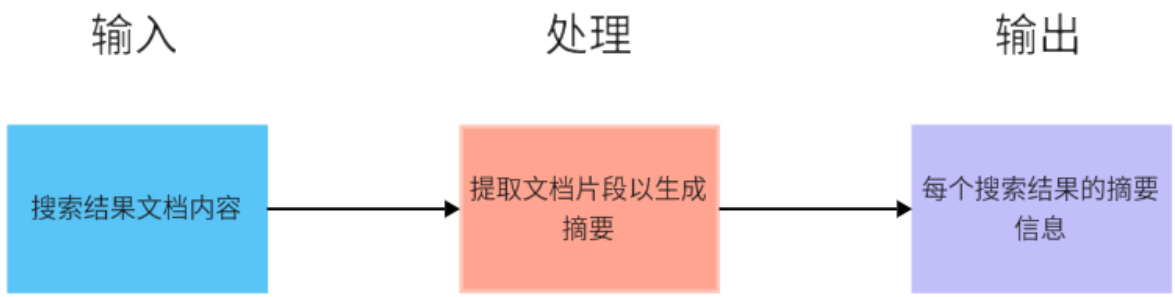

图3-3-2-5 结果摘要

vi. 用户反馈

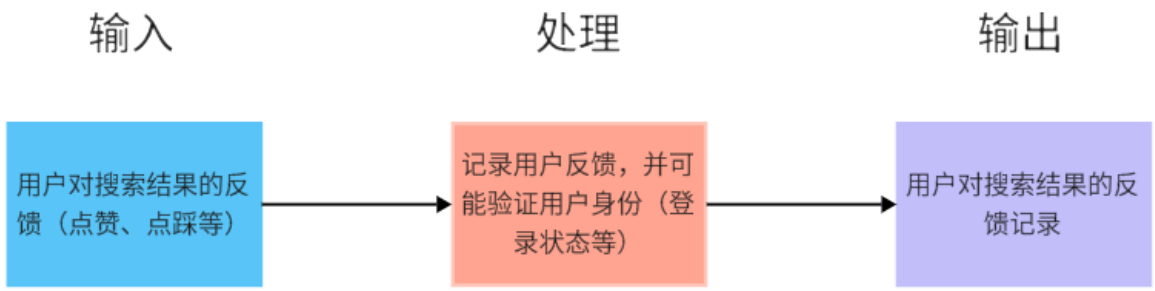

图3-3-2-6 用户反馈

vii. 搜索历史显示

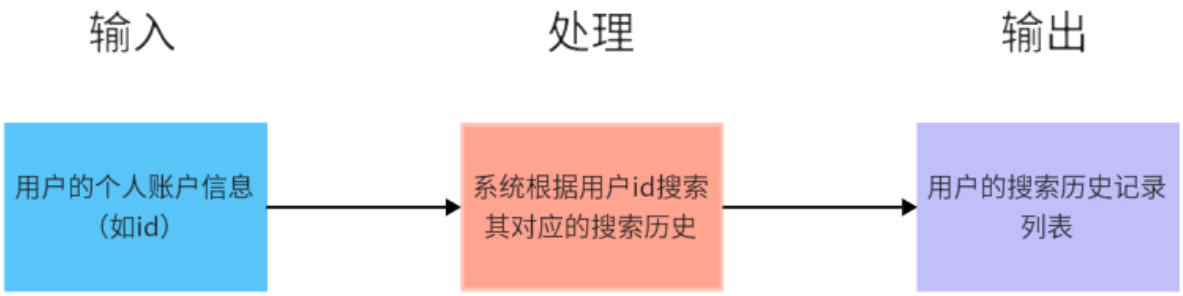

图3-3-2-7 搜索历史显示

viii. 个性化推荐

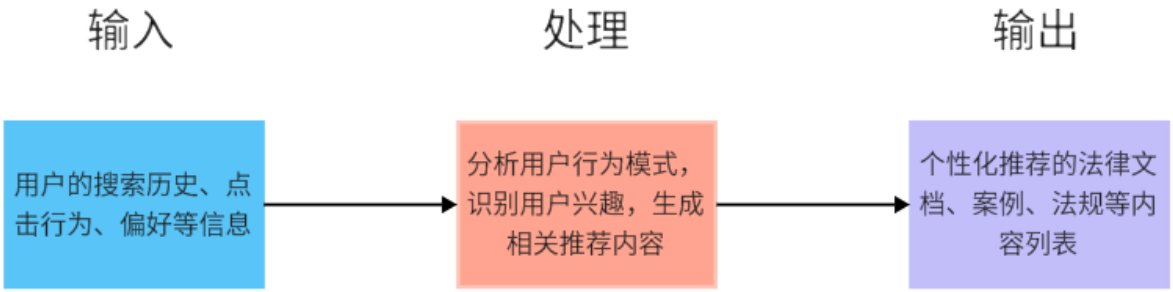

图3-3-2-8 个性化推荐

ix. 法律文档展示

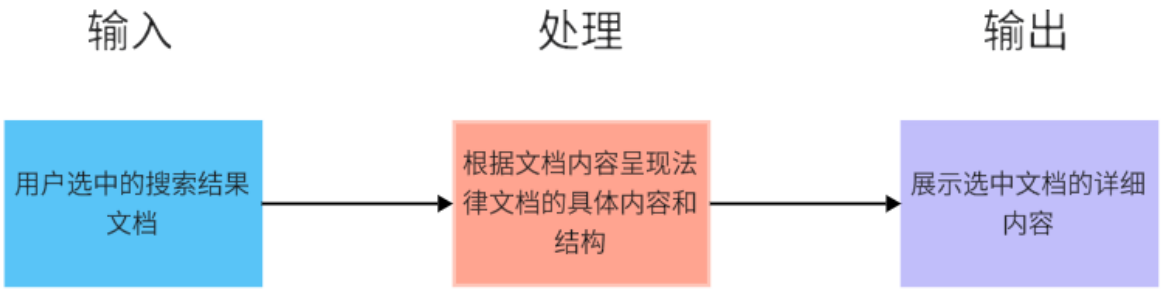

图3-3-2-9 法律文档展示

x. 相关文档推荐

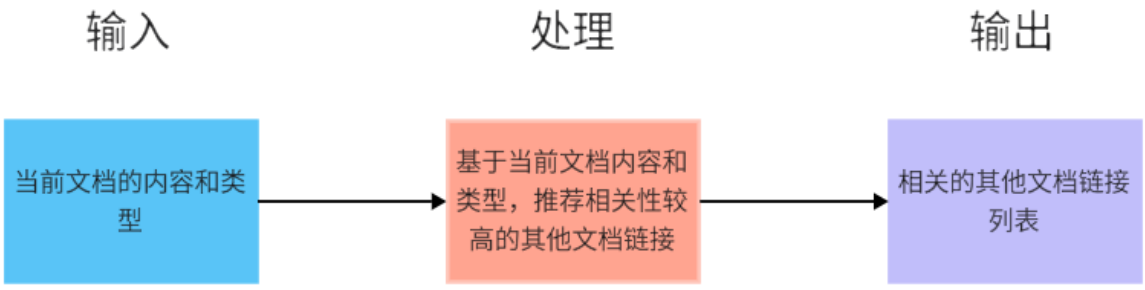

图3-3-2-10 相关文档推荐

xi. 关键词提取（AI 辅助）

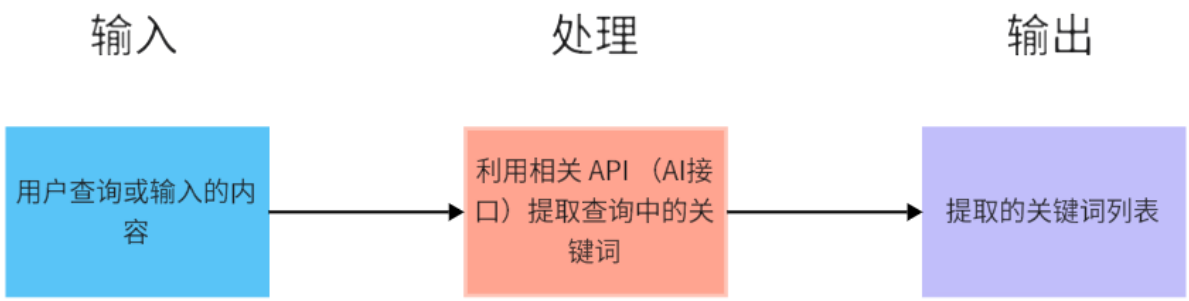

图3-3-2-11 关键词提取（AI辅助）

xii. 文档总结（AI 辅助）

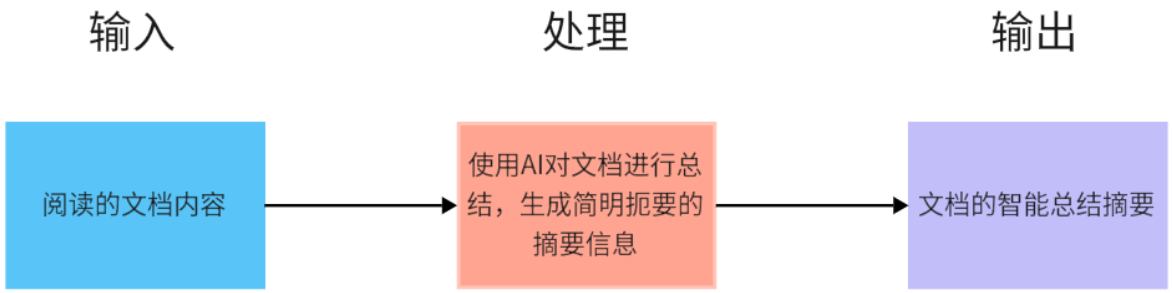

图3-3-2-12 文档总结（AI辅助）

xiii. 生成关键词图谱

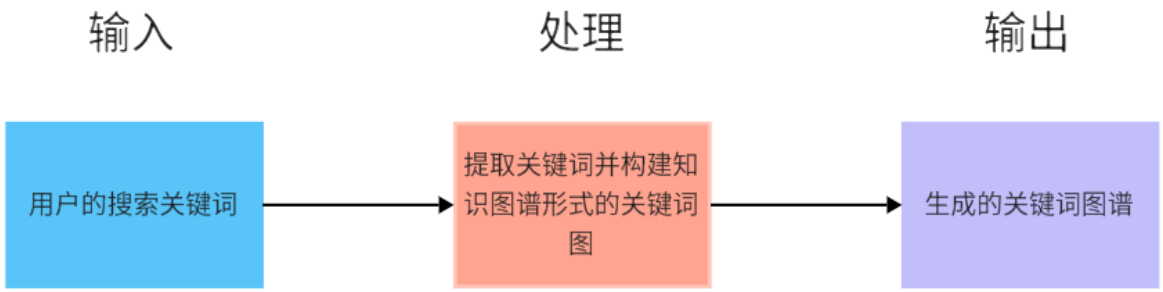

图3-3-2-13 生成关键词图谱

xiv. 图谱转换与跳转搜索

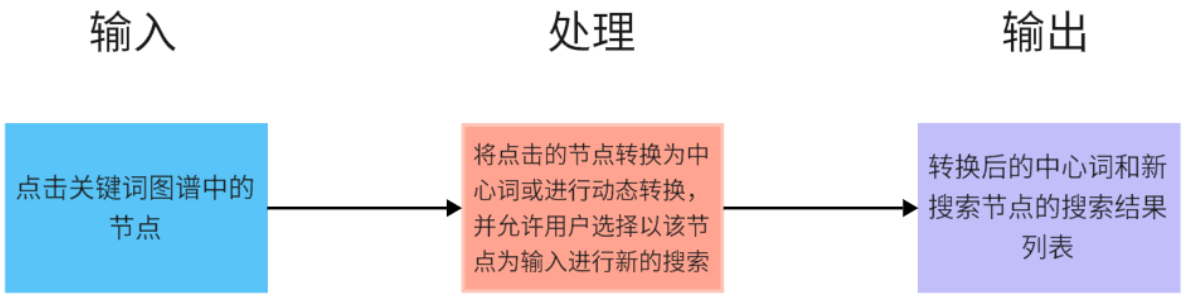

图3-3-2-14 图谱转换与跳转搜索

3.4 运行环境

3.4.1 软件运行环境

项目	名称	版本
操作系统	Windows 10 及 以 上 ，Linux，MacOS	/
网站服务器	Nginx	1.22.1以上
数据库服务器	Linux socket	/
数据库服务器类型	MySQL / MongoDb / Elasticsearch	8.0
浏览器	Chrome，Safari，Edge，Firefox等主流浏览器	/

3.4.2 硬件运行环境

项目	名称
操作系统	CPU:CORE i5 及以上
内存： 4G 及以上	/
硬盘：500G 及以上	/
应用服务器	内存： 2GB 及以上
数据库服务器邮件服务器文件服务器	硬盘：50G 及以上
通讯设备	网线 ： 具有良好的数据传输能力

3. 接口设计

4.1 内部接口

内部接口即指前后端交互所使用的请求接口，采用主流的基于 HTTP 协议的 RESTful 风格接口，使接口设计简洁明了。

4.1.0 统一规范

返回格式

前端给后端发送一个请求后，后端应返回一定格式的响应体，其中包含请求结果（成功与否）、错误码、错误信息、负载数据等必要信息，统一的返回格式可以方便前端统一处理请求结果。此外，由于除 `data` 外，其余属性的类型一致，因此下文中功能接口的 `response` 类型仅指 `data` 类型。

这里的统一返回格式采用如下设计：

```
export interface Response {
  success: boolean; // 请求是否成功处理并返回
  data: any; // 返回的数据
  errorCode: string; // 错误码 (0 if success)
  errorMessage: string; // 错误信息
}
```

接口规范

此外，为了最后部署的时候能够区分页面和接口，同时也是为了方便前端调试做接口的转发，后端接口路径统一添加 `/api` 的前缀。

类型规定

定义一些可复用的类型接口（仅代表前端处理时需要用到的数据类型，不代表数据库中的存储方式）：

用户信息

用户可以查看的与自己有关的信息。

```
export interface UserInfo{
  id: number; //唯一的用户id
  userName: string; //唯一的用户名
  email: string; //用户的注册邮箱
  history: string[];//最近20条搜索记录（输入值）
}
```

图谱节点

参考了 ant-design-charts 的辐射树图配置参数：辐射树图

```
export interface NodeInfo{
  id: string; //节点的唯一标识符
  value: string; //节点值（显示出来的）
  children?: NodeInfo[]; //连接的下一层子节点
}
```

搜索结果通用条目

这个类型的含义是：列表显示搜索结果时（没点进具体内容的时候），所有类型的搜索结果的通用显示形式。

```
export interface ResultItem{
  id: string; //唯一标识符
  title: string; //该条结果的标题
  description: string; //内容摘要
  date: string; //内容发布或更新时间，使用 ISO 8601 时间格式
  resultType: number;//该条结果的类型，0-法律法规，1-裁判文书...
  source: string; //来源网站或作者
  feedbackCnt: {
    likes: number; //点赞数量
    dislikes: number; //点踩数量
  }
}
```

4.1.1 用户管理

本模块用于用户状态管理，包括用户的注册、登录等功能，接口均以 `/user` 开头。

用户注册

新用户输入邮箱、用户名和密码，进行注册，规定所用邮箱在之前没有注册过账号，且用户名唯一，邮箱的格式要求在前端检查即可。成功完成注册后，后端返回完整的用户信息。

URL：`/user/register`

Method：POST

Request：

```
{
  email: string;
  username: string;
  password: string;
}
```

Response：`UserInfo`

对`errorCode`和`errorMessage`进行约定：

- 该邮箱已注册：`errorCode = 1, errorMessage = "This email is already registered!"`
- 用户名已存在：`errorCode = 2, errorMessage = "User name already exists!"`

获取用户信息

尝试向后端获取当前用户的信息，若拥有登录态则可以直接获取。

URL：`/user/info`

Method：GET

Request：`null`（登录态通过请求头的 Cookie 获取）

Response：`UserInfo`

用户登录

用户输入邮箱和密码进行登录，根据输入值的正确性返回不同的登录结果。

URL：`/user/login`

Method：POST

Request：

```
{
  email: string;
  password: string;
}
```

Response：`UserInfo`

对`errorCode`和`errorMessage`进行约定：

- 密码错误：`errorCode = 1, errorMessage = "Wrong password!"`
- 邮箱未注册：`errorCode = 2, errorMessage = "This email is not registered!"`

用户登出

用户退出登录，当前账号失去登录态。

URL：`/user/logout`

Method：POST

Request：`{id: number;}`

Response：`null`

4.1.2 搜索结果处理

本模块用于搜索结果的展示及对结果条目的相关操作，接口均以`/search`开头。

搜索结果列表展示

对于给定的输入量和筛选条件（包括搜索类型、高级搜索中的规定），返回以合适的顺序完成排序的搜索结果列表，为方便查看，采用分页展示。

URL：`/search/list`

Method：GET

Request：

```
{
  input: string; //用户在输入框中输入的内容
  //搜索类型，0-全文搜索,1-标题搜索,2-作者/出处搜索,3-主题搜索
  searchType: number;
  //筛选条件
  //搜索结果类型，0-法律法规，1-裁判文书，2-期刊论文
  //3-法律观点，4-资讯
  resultType: number[];
  //对发布日期的筛选，均以"YYYY-MM-DD"形式
  startTime: string; //起始时间
  endTime: string; //终止时间
  //后期涉及数据更新的话可以再规定一下更新时间
  //updateTime: string;
  pageSize: number; //每页条数
  pageNumber: number; //第x页，从0开始
}
```

Response：

```
{
  results: ResultItem[]; //搜索结果列表
```

```
  total: number; //相关搜索结果共x条
}
```

用户反馈

已登录的用户可以对某条搜索结果点赞/踩，该操作将被后台记录，影响该条结果权重用户之后的搜索结果。

URL：`/search/feedback`

Method：POST

Request：

```
//登录态和用户身份从cookie里拿
{
  id: number; //操作的文档id
  feedback: boolean; //true-点赞，false-点踩
}
```

Response：`null`

对`errorCode`和`errorMessage`进行约定：

- 未登录：`errorCode = 1, errorMessage = "Not logged in yet!"`

结果详情展示

用户点击搜索结果列表中的某一条后，即可进入详情页查看详细内容，这些内容应以合适的格式整理并呈现。

URL：`/search/detail`

Method：GET

Request：`{id:string;}//内容的id`

Response：

```
{
  title: string; //标题
  source: string; //文档出处/作者
  publishTime: string; //发布时间，使用 ISO 8601 时间格式
  feedbackCnt: {
    likes: number; //点赞数量
    dislikes: number; //点踩数量
  }
  content: string; //正文
```

```
  resultType: number; //文档类型，0-法律法规，1-裁判文书...
  link: string; //链接至原文
}
```

相关内容推荐

在文档详情页，基于该文档的内容与类型，提供与该篇文档内容相关性较高的其他文档的链接。

URL：`/search/relat`

Method：GET

Request：

```
{
  id: number; //文档id
  type: string; //文档类型，可以再进行规定
}
```

Response：

```
{
  links:{
    title: string; //相关内容的标题
    id: number; //跳转到详情页所需的该内容id
  }[];//一个相关内容数组
}
```

4.1.3 AI 辅助

本模块用于涉及 AI 辅助模块的操作，处理流程为前端向后端发送请求，后端对请求做一定处理（如添加prompts）后再将该请求转发给 GPT 调用接口。本模块接口均以 `/ai` 开头。

关键词提取

用户输入一段文字，返回这段文字中提取出的若干关键词。

URL：`/ai/extract`

Method：GET

Request：`{input: string;}`

Response：`{res: string;}`

文档总结

对当前文档内容进行总结，生成一段文字概要（考虑对不同类型的文章生成合适的概要形式）。

URL：`/ai/summarize`

Method：GET

Request：

```
{
  id: number; //文档id，方便后端寻找这篇文档的内容
  type: string; //文档类型，可以再进行规定
}
```

Response：`{res: string;}`

4.1.4 搜索历史

本模块提供基于用户历史搜索的个性化服务，接口均以`/personal`开头。

用户搜索历史显示

对于已登录状态的用户，可以查看自己的搜索历史。

URL：`/personal/history`

Method：GET

Request：

```
//登录态及用户身份通过请求头的 Cookie 获取，不在params里传
{
  reqNumber: number; //所需的历史条数
}
```

Response：`{historys: string[];}`

对`errorCode`和`errorMessage`进行约定：

- 未登录：`errorCode = 1, errorMessage = "Not logged in yet!"`

个性化推荐内容

用户在首页搜索框下方可以看到若干条推送内容：如果是已登录用户，这些推送内容将与其搜索历史相关；如果未登录，则随机推送。

URL：`/personal/recommends`

Method：GET

Request：`null`（登录态通过请求头的 Cookie 获取）

Response：

```
{//类似于相关内容推荐的res
  recommends:{
    title: string; //推送内容的标题
    id: number; //跳转到详情页所需的内容id
  }[];//一个推荐内容数组
}
```

4.1.5 知识图谱

获取知识图谱结构

用户每次进行搜索，或者点击切换知识图谱中的中心节点，都需要获取新的知识图谱结构，以渲染出新的知识图谱。其中，若为每次搜索后自动产生的知识图谱，则其中心词由输入内容分析得出。

URL：`/graph/get`

Method：GET

Request：

```
{
  input: string; //即搜索框中输入的内容，二次编辑图谱时为指定的中心词
  depth: number; //搜索深度
}
```

Response：

```
{
  center: NodeInfo;//中心词条（暂定只有一个中心词）
  desc: string; //对中心词条的描述
}
```

4.2 外部接口

4.2.1 GPT 调用接口

- **URL**：https://api.openai.com/v1/chat/completions
- **Method**：GET
- **Request**：

```
curl https://api.openai.com/v1/chat/completions \
-H"Content-Type: application/json"\
-H"Authorization: Bearer$OPENAI_API_KEY"\
-d'{
"model": "gpt-3.5-turbo",
"messages": [{"role": "user", "content": "Say this is a test!"}],
"temperature": 0.7
}'
```

- **Request Parameters**：

参数名	参数说明	备注
$OPENAI_API_KEY	Api key	
model	AI模型	
messages	消息	到目前为止包含对话的消息列表。
temperature	采样温度	取值在0到2之间。像0.8这样的较高值会使输出更加随机，而像0.2这样的较低值会使输出更加集中和确定。

- **Response**：

```
{
    "id": "chatcmpl-abc123",
    "object": "chat.completion",
    "created": 1677858242,
    "model": "gpt-3.5-turbo-1106",
    "usage": {
        "prompt_tokens": 13,
        "completion_tokens": 7,
        "total_tokens": 20
```

```
    },
    "choices": [
        {
            "message": {
                "role": "assistant",
                "content": "\n\nThis is a test!"
            },
            "finish_reason": "stop",
            "index": 0
        }
    ]
}
```

- **Response Parameters**：

参数名	参数说明	备注
id	会话id	
object	Object类型	恒为"chat.completion"
created	创建时间戳	
model	选用模型	
usage	Tokens使用情况	
choices	消息结果选择列表	可能有多条不同的返回消息

4.2.2 爬虫接口

4.2.2.1 国家法律法规数据库

URL：https://flk.npc.gov.cn/api (法律文档列表接口)

Method：GET

Request：

```
{
  "type": "flfg", /* {"法律法规": "flfg", "行政法规": "xzfg", "监察法规":
                "jcfg", "司法解释": "sfjs", "地方性法规" : "dfxfg"} */
  "searchType": "title:vague", //搜索类型
  "sort": true, //是否排序
  "page": 1, //起始页
  "size": 10, //搜索页数，最大为10
```

```
}
```

Response：

```
{
    "timestamp":1701599322736, //时间戳
    "success":true, //是否成功
    "message":"success",
    "code":200, //响应码
    "result":{ //返回结果
        "data":[{ //返回数据列表
            "id":"ZmY4M......Y%3D", //文档id
            "title":"中华人民共和国爱国主义教育法", //文档标题
            "office":"全国人民代表大会常务委员会", //发布组织
            "publish":"2023-10-24 00:00:00", //发布日期
            "expiry":"2024-01-01 00:00:00", //生效日期
            "type":"法律", //文档类型
            "url":"./detail2.html?ZmY4M......Y%3D" //文档详情页url
            }],
        }
}
```

URL：https://flk.npc.gov.cn/api/detail （法律文档详情接口）

Method：POST

Request：

```
{
  "id": "ZmY4M......Y%3D" //文档id
}
```

Response：

```
{
    "timestamp": 1701599815128, //时间戳
    "success": true, //是否成功
    "message": "success",
    "code": 200, //响应码
    "result": { //返回结果
        "title": "中华人民共和国爱国主义教育法", //文档标题
        "office": "全国人民代表大会常务委员会", //发布组织
```

```
        "publish": "2023-10-24 00:00:00", //发布日期
        "expiry": "2024-01-01 00:00:00", //生效日期
        "level": "法律", //法律效力
        "body": [ //资源列表
            {
                "type": "WORD", //资源类型
                "path": "/flfg/WORD/927......f5e.docx", //资源url
            }
        ],
        "otherFile": [] //相关文档列表
    }
}
```

4. 数据设计

5.1 MySQL 数据表设计

5.1.1 用户数据表

Property	Type	Note
id	bigint	PK
username	varchar	unique
email	varchar	unique
password	varchar	
history	blob	历史记录
setting	blob	用户设置
feedback	blob	用户反馈

5.1.2 知识图谱结点表

Property	Type	Note
id	bigint	PK
title	varchar	结点名称

group	int	所属组别

5.1.3 知识图谱连接表

Property	Type	Note
id	bigint	PK
source	bigint	FK
target	bigint	FK
weight	int	边权重

5.2 Elastic Search索引设计

5.2.1 法律法规索引设计

Property	Type	Note
_id		
title	text	标题
office	keyword	制定机关
publish	date	公布日期
expiry	date	施行日期
type	byte	类型
status	byte	时效性
content	text	内容
url	binary	链接
like	integer	点赞数量
dislike	integer	点踩数量

5.2.2 裁判文书索引设计

Property	Type	Note
_id		
title	text	标题
caseId	binary	案号
court	text	审理法院
type	byte	案件类型
brief	text	案由
proceeding	keyword	审理程序
date	date	裁判日期
party	binary	当事人
basis	text	法律依据
content	text	内容
url	binary	链接
like	integer	点赞数量
dislike	integer	点踩数量

5. 用户界面设计

用户界面设计是对需求文档中划分的界面原型的进一步细化和具体实现，相比于功能区域的划分，这一步会更注重配色、美观性、用户交互方面的考虑。

6.1 搜索引擎首页

图6-1-1 搜索引擎首页设计图（拉开高级搜索栏）

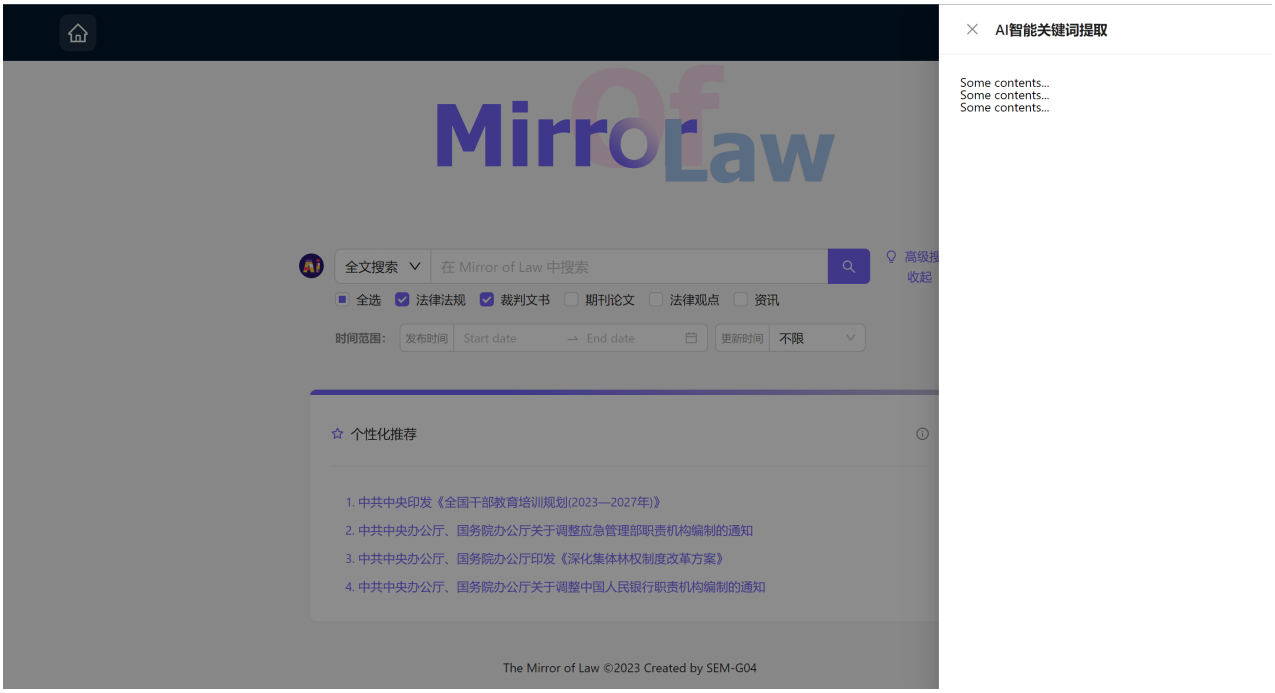

图6-1-2 搜索引擎首页设计图（拉开AI功能抽屉）

6.2 搜索结果列表页

图6-2 搜索引擎首页设计图（未拉开高级搜索栏）

6.3 详情页

图6-3-1 详情页整体布局图

图6-3-2 AI辅助功能

6. 系统出错设计

7.1 出错信息

出错类型	可能原因	处理方法
数据库连接失败	• 由于并发操作的用户数量很大，导致数据库访问读写率降低 • 网络故障：由于网络问题，系统无法与数据库建立连接。 • 数据库服务器宕机：数据库服务器崩溃或停止服务。 • 连接参数错误：数据库连接参数配置错误。	• 提供网络检测功能，引导用户检查网络连接，刷新页面或稍后再试。 • 实施自动化监控，及时发现数据库服务器宕机并自动重启。 • 实施监控机制，及时发现数据库连接问题。 • 记录错误日志，以便管理员可以查看和解决问题。
数据无法加载	• 数据源异常：数据源出现故障或无法访问。 • 数据格式错误：数据库中的数据格式不符合系统预期。 • 数据量过大：大量请求导致系统无法及时加载数据。	• 实施数据源监控，及时发现异常并自动切换至备用数据源。 • 在系统端实现数据格式验证，确保数据的一致性和有效性。 • 采用分页加载策略，确保系统可以逐步加载大量数据，避免系统崩溃。
系统服务异常	• 服务器故障： 可能由于硬件故障、网络问题或系统崩溃导致服务不可用。	• 部署负载均衡，确保服务器资源充足。

	• 系统维护： 系统正在进行维护和更新。 • 负载过高： 突然的大量请求导致服务器无法处理。	• 提供维护通知，告知用户系统即将进行维护，建议稍后再试。 • 实施自动化监控，及时发现并修复服务器故障，降低系统不可用时间。
搜索结果为空	• 关键字无匹配项： 用户输入的关键字在系统中没有匹配的内容。 • 搜索条件过于严格： 高级搜索或筛选条件导致搜索结果为空。 • 数据更新延迟： 系统数据未及时更新，导致搜索不到最新的信息。	• 提供相关搜索建议，引导用户修改关键字或放宽搜索条件。 • 显示搜索条件，让用户了解为何搜索结果为空。 • 加强数据更新机制，确保系统数据及时同步，减少搜索结果为空的情况。
用户输入错误	• 拼写错误： 用户可能因为拼写错误而输入了无效的关键字。 • 格式错误： 输入的关键字格式不符合系统要求，可能包含特殊字符或过长的字符串。	• 在前端进行格式验证，引导用户按照规定格式输入关键字 • 提供拼写建议，纠正用户拼写错误 • 提供友好的提示，指导用户如何正确输入关键字

7.2 预防&补救措施

a. **系统备份**

定期备份系统数据，当系统数据因不可抗拒力丢失时，可以启用备份数据。

b. **分布式部署**

将系统部署到不同计算机上，减小某台计算机硬件损坏造成的数据丢失的影响。

c. **异常监控**

部署系统监控工具，及时发现异常请求，防范潜在问题。

d. **日志记录**

记录系统运行日志，便于后续故障排查。

e. **定期维护**

定期进行系统维护，包括数据库优化、系统清理等。

7.3 系统维护设计

7.3.1 概述

i. 连接数据库时，需要在创建数据库连接、销毁数据库连接时使用 try catch 语句捕获异常，对不同的错误信息尽量区分输出。使用 MyBatis 中间件， 有效防止 SQL 注入攻击，保障网站的安全。

ii. 管理员有权对整个网站的状况进行控以防系统出现不可预计的错误防止系 统显示不合法信息。

iii. 系统维护人员每次维护后需要留下完备可读的系统维护日志便于管理员和其他维护人员查看。

iv. 做好代码版本管理和溯源的工作，发现异常和 bug 时及时排查并找到责任人进行修复。

7.3.2 检测点设计

7.3.2.1 搜索词条

- 关键词普通搜索
- 关键词与运算搜索
- 关键词或运算搜索
- 关键词非运算搜索
- 关键词混合逻辑搜索
- 关键词过滤高级搜索
- 关键词条件搜索
- 搜索过程中显示历史记录
- 搜索过程中进行智能补全
- 拼音搜索

7.3.2.2 查看搜索结果

- 搜索结果页面显示
- 搜索结果类型筛选
- 搜索结果跳转法规信息界面
- 搜索结果跳转案例新闻界面
- 搜索结果排序
- 相关内容个性化推荐
- 智能机器人问答

7.3.2.3 详情内容展示

- 法规、案例基本信息呈现

- 案例关键数据可视化呈现
- 案例知识图谱可视化呈现
- 法律新闻呈现

7.3.3 相关维护设计

i. 硬件资源维护：定期清理服务器硬盘垃圾，可根据网站实际需求选择升级服务器性能。

ii. 数据库维护：定期备份数据库文件。

iii. 系统功能升级：根据用户实际访问平台的需求，对于系统功能进行合理的更新。