

律

镜

2023-11-1

项目计划书

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

1. 引言	3
1.1 编写目的	3
1.2 软件系统名称	3
1.3 相关人员及项目支持	3
1.3.1 任务提出者	3
1.3.2 开发者	3
1.3.3 用户	4
1.3.4 实现该软件的计算机网络	4
1.4 读者对象	4
2. 项目概述	4
2.1 项目背景	4
2.2 项目介绍	5
2.3 项目的整体目标和意义	5
2.4 可交付成果	6
2.4.1 产品	6
2.4.2 手册	6
2.4.3 服务	6
3. 选题的整体方案	6
3.1 项目的主要内容	7
3.2 功能单元	7
3.3 数据采集和处理策略	7
3.3.1 数据来源	8
3.3.2 数据采集	8
3.3.3 数据存储	8
3.3.4 数据分类	8
3.4 用户界面设计与体验	8
3.5 技术架构和工具选择	9
3.5.1 前端技术栈	9
3.5.2 后端技术栈	10
3.5.3 其他辅助工具	10
4. 要突破的技术点	10
4.1 项目中的难点和挑战	10
4.2 技术创新点	10
4.3 预期的解决方案和方法	11
5. 项目计划	11
5.1 项目阶段与里程碑	11
5.2 项目时间表	12
5.3 团队成员和角色分配	12
5.4 持续改进和监测策略	13
6. 项目成本分析	14
6.1 软硬件资源说明	14
6.2 成本分析	15
6.2.1 人力资源成本	15
6.2.2 硬件成本	15
6.2.3 软件成本	15
6.2.4 API 成本	15
6.2.5 数据成本	15

7. 专题计划要点	16
7.1 开发人员培训计划	16
7.2 质量保证计划	18
7.3 测试计划	19
7.3.1 静态测试	19
7.3.2 单元测试	20
7.3.3 系统测试	20
8. 参考文献	20



项目计划书

1. 引言

1.1 编写目的

本项目计划书是2023年秋冬学期软件工程管理G04小组的项目总体开发计划书，旨在说明整个法律领域信息垂直搜索引擎项目的实施过程中的开发技术栈、开发进度安排、角色分配、软硬件资源、测试约定、工作规范、工作要点等内容。通过明确这些方面的内容，来有效地管控项目的进展，推动项目的顺利进行。此外，我们还将运用项目管理策略与方法，详细说明如何计划、组织和实施项目开发工作，以确保项目的成功实施。

1.2 软件系统名称

律镜 —— 法律领域垂直搜索引擎

1.3 相关人员及项目支持

1.3.1 任务提出者

浙江大学软件工程管理课程任课老师：邵健、邹强

1.3.2 开发者

浙江大学软件工程管理课程小组第四小组

组长：徐毕颖

组员：章越、沈书豪、王梓轩、黄文杰

1.3.3 用户

1. **法律专业学生**：这个项目可以为法学院的学生提供一个有用的研究工具，帮助他们查找法律案例、法规、判例法和其他相关资料。
2. **法律从业者**：律师、法律顾问和法律研究员可能会使用该搜索引擎来查找相关法律信息，以支持他们的法律研究和案件准备工作。
3. **一般公众**：人们可能需要法律信息来了解其权利、义务或解决法律问题，这个搜索引擎可以为他们提供可靠的法律信息来源。

4. **科研人员：**社会科学和政策研究人员可以使用该搜索引擎来获取法律数据和统计信息，以进行学术研究和政策分析。

1.3.4 实现该软件的计算机网络

基于公网 ip 的服务器与数据库

1.4 读者对象

- 软件客户代表
- 项目经理
- 项目开发人员
- 项目测试人员
- 项目监督管理和质量保证人员
- 其它所有想要了解本项目质量管理总体情况的人员

2. 项目概述

2.1 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。

2.2 项目介绍

我们的项目旨在开发一款专注于法律领域的垂直搜索引擎，为用户提供全面、准确、高效的法律信息检索服务。该搜索引擎将覆盖法律条例、法律案例、法律判决等多种法律信息资源，并引入了AI技术进行赋能，使系统能够针对用户提供的案例描述自动提取关键词并进行检索。

通过结合人工智能技术和搜索引擎框架优质的检索算法，我们的搜索引擎将能够理解和解析复杂的法律语言，提供精准的搜索结果。用户可以通过输入关键词、案例描述或特定法律问题来获取相关法律信息，包括相关法规、相关案例、相关判决等。

2.3 项目的整体目标和意义

本项目的整体目标是开发一款面向法律领域的垂直搜索引擎，满足用户对法律信息深度搜索的需求，并提供高效、准确的搜索结果。具体而言，我们的目标包括：

- 提供全面的法律信息覆盖：整合法律条例、法律案例、法律判决等多种法律信息资源，为用户提供全面的法律信息覆盖，帮助用户准确理解和应用法律知识。
- 引入人工智能技术提升搜索体验：通过引入AI技术，使用户能够根据案例描述自动提取关键词进行检索，提供更智能、个性化的搜索体验，帮助用户快速定位所需信息。
- 提供高质量的搜索结果：通过数据源筛选与搜索引擎框架的选则，对法律信息进行精准解析和分类，提供高质量、高效率的搜索结果，帮助用户快速找到最相关的法律资料。

该项目的意义在于：

- 提升法律从业者和公众获取法律信息的效率和准确性，降低信息搜索的时间成本和学习成本。
- 为律师、法官、学者等法律专业人士提供一个可靠的工具，帮助他们更好地进行法律研究和决策支持。
- 促进法律知识的传播和普及，使更多的人能够了解和应用法律，增强法律意识和法治观念。

我们希望能够为用户提供一种高效、准确的法律信息搜索方式，推动法律领域的数字化转型，为法律行业的发展和社会的进步做出贡献。

2.4 可交付成果

2.4.1 产品

最终交付的产品是一个基于法律领域深度搜索引擎的综合信息平台——“律镜”，其具体组成部分包括：

- 完整的前后端开发代码
- 数据库创建文件
- 相关开发文档
- 网站所需资源文件

2.4.2 手册

- 验收报告及交接手册
- 项目技术说明

向系统管理员、部署人员、运维人员或者有购买该项目意愿的客户群体提供该项目所设计的技术，包括但不限于前后端开发环境配置、前后端开发IDE推荐、项目架构、可能设计的技术问题及可供参考的解决方案。

- 操作文档

提供给使用此搜索引擎网站的用户。对于系统的界面、功能以及操作说明进行描述，使用户对于网站的操作可以快速上手。

- 功能介绍

通过思维导图、树形结构图等方式向用户展现各个功能模块的关联关系，并辅以文字说明和效果展示图，让用户更加明晰该项目所能提供的价值。

- 维护手册

主要包括搜索引擎网站系统说明、程序模块说明、操作环境、支持软件的说明、维护过程的说明，便于维护人员对于系统的维护。

2.4.3 服务

提供使用教程以及咨询服务，使解决用户在使用网站过程中的困难。提供后续一定年限的维护服务，当软件出现问题时提供技术支持。

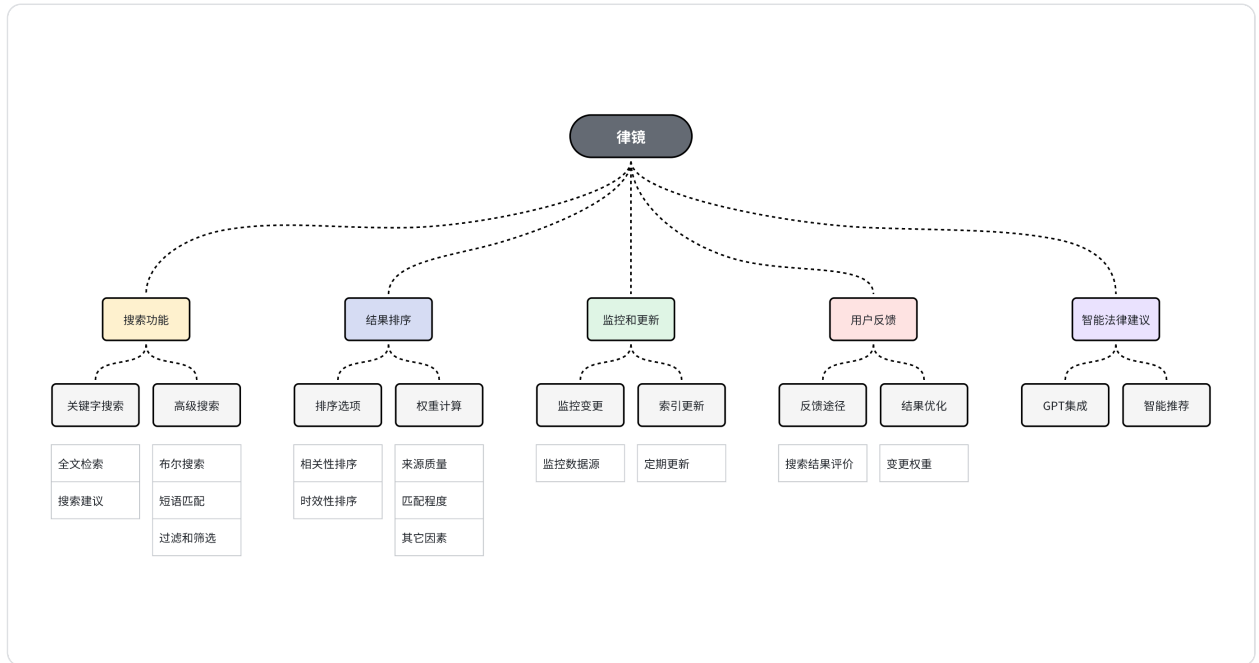
3. 选题的整体方案

3.1 项目的主要内容

- a. **数据采集与索引**：该项目的首要任务是整合各种法律数据库和资源，包括法律文本、司法解释、法律评论、案例研究、法学期刊等，并将其索引，建立一个庞大的法律文档数据库，从而为用户提供多样化的有深度的法律资源。
- b. **搜索引擎**：项目包含一个强大的搜索引擎，支持全文搜索、关键词搜索、高级搜索和过滤功能等，为用户提供便利的搜索功能。

3.2 功能单元

对于本项目的搜索引擎部分，即核心功能模块，其组成可参照下图：



搜索引擎部分功能模块示意图

- a. **搜索功能**：全文检索法律文档，支持布尔搜索、短语匹配和高级搜索选项。
- b. **相关性排序**：对搜索结果进行排序，确保最相关的文档排在前面。
- c. **监控和更新**：定期监控法律文档的变化，确保数据库和搜索结果的及时更新。
- d. **用户反馈和改进**：收集用户反馈，改进搜索引擎性能。
- e. **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。

同时，为满足用户多样化和更灵活的需求，我们还设计了以下附加功能：

- a. **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。
- b. **知识图谱**：将相关的法律内容联系起来，构建一张知识图谱网，帮助用户更高效地搜索。

3.3 数据采集和处理策略

3.3.1 数据来源

- 确定数据来源，包括法律文件、司法解释、裁判文书、法律博客、新闻、政府公告、法律网站等。
- 了解数据的版权、许可和使用限制。

3.3.2 数据采集

- 开发网络爬虫或API集成，以自动获取数据。
- 遵守法律和网站的使用政策。

- 定期更新数据以确保信息的新鲜性。

3.3.3 数据存储

- 清理和规范数据，包括去除HTML标记、格式化文本、去除重复项等。
- 制定规则和算法以将非结构化数据（如法律文档）存储至数据库中，并创建索引。

3.3.4 数据分类

- 标注法律文件和案例以确定关键信息，如案号、法院、判决日期、当事人等。
- 对数据进行分类，如刑法、民法、知识产权等领域，以便用户进行更精确的检索。

3.4 用户界面设计与体验

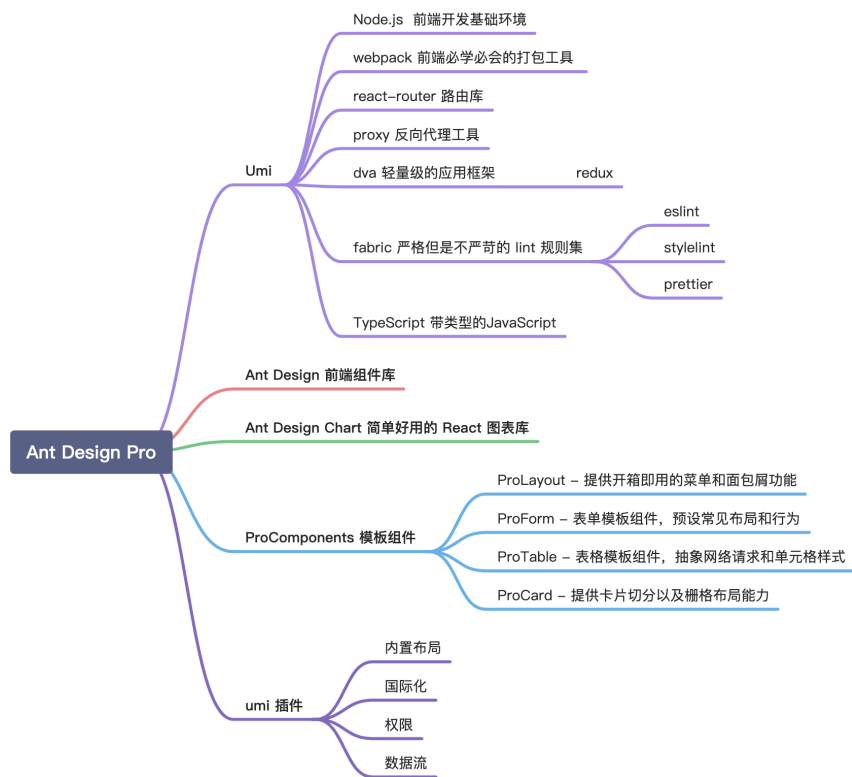
该部分仅列出本项目在用户界面与体验设计方面的几项原则，具体的设计方案（如可能有的Figma原型）将在后续设计文档及相关记录中给出和完善。

- 信息分类和过滤：**法律领域的搜索引擎通常需要处理大量的法律文件、案例和相关信息。界面设计应该提供有效的信息分类和过滤功能，使用户能够按照法律条款、案例、法规、法院等进行精确的搜索和筛选。
- 法律术语和语言：**法律领域具有特定的术语和语言规范。界面设计应该考虑到用户对这些术语的理解和使用，提供术语解释、法律词典或相关参考资料的链接，可以以关联图形式可视化呈现术语、条款间的联系，以帮助用户更好地理解和应用法律信息、更高效地利用本搜索引擎。
- 搜索结果的权威性和可信度：**法律领域的搜索引擎需要提供高质量、权威性的搜索结果。界面设计应该明确标示搜索结果的来源和可信度，并使得可行度与权威性与搜索结果排名相关，以帮助用户评估信息的可靠性和适用性。
- 案例关键词提取：**在搜索引擎的实现中，完整的案例输入可能因为叙述所需过多无关词的存在而影响结果质量，法律领域搜索引擎应提供案例关键词提取功能，为用户案例提供可选的智能关键词提取，帮助用户更准确地搜索。
- 法律文档的阅读：**界面设计应该支持法律文档的方便阅读、查询和标注，提供舒适的文档阅读体验，帮助用户在文档中快速定位和标记重要信息。

3.5 技术架构和工具选择

3.5.1 前端技术栈

- 前端框架：React
- CSS组件库：Ant Design
- 脚手架：Ant Design Pro



脚手架提供的框架资源，引自Ant Design Pro官网

3.5.2 后端技术栈

- 后端框架：Spring Boot
- 数据库：MySQL+MongoDB
- ORM框架：MyBatis
- 爬虫语言：Python
- 搜索引擎框架：ElasticSearch

3.5.3 其他辅助工具

- 版本控制：Git
- 文档管理/会议记录：飞书
- API调试工具：Apifox
- 低/高保真图绘制：Figma

4. 要突破的技术点

4.1 项目中的难点和挑战

- a. **多样化的法律信息源整合：**法律领域的信息来源广泛，包括法律条例、法律案例、法律判决等多种类型的文本。整合和处理这些多样化的法律信息源是一个挑战，需要建立有效的数据抓取、清洗和整合机制。
- b. **搜索结果的准确性和排序：**针对垂直搜索引擎而言，搜索结果的准确性和排序是关键问题。需要设计和优化搜索算法，使得相关性高的法律信息能够排在前面，并可能需要考虑到法律文本的特殊性，如法律条款的权威性和适用性。
- c. **AI技术的引入和应用：**引入AI技术实现根据案例描述自动提取关键词检索的能力是该项目的创新点之一。在应用人工智能技术时，需要解决模型接口的调用以及prompts设计的问题，确保生成的关键词准确性和相关性。

4.2 技术创新点

- a. **专注于法律领域的垂直搜索引擎：**与传统搜索引擎相比，我们的搜索引擎专注于法律领域，提供深度搜索和精准结果，满足法律从业者和公众对法律信息的特定需求。
- b. **引入人工智能技术实现智能搜索：**通过引入AI技术，用户可以根据案例描述自动提取关键词进行检索，提供更智能、个性化的搜索体验，帮助用户快速定位所需信息。
- c. **多种法律信息源的整合：**本项目能够整合法律条例、法律案例、法律判决等多种法律信息源，为用户提供全面的法律信息覆盖，帮助用户准确理解和应用法律知识。

4.3 预期的解决方案和方法

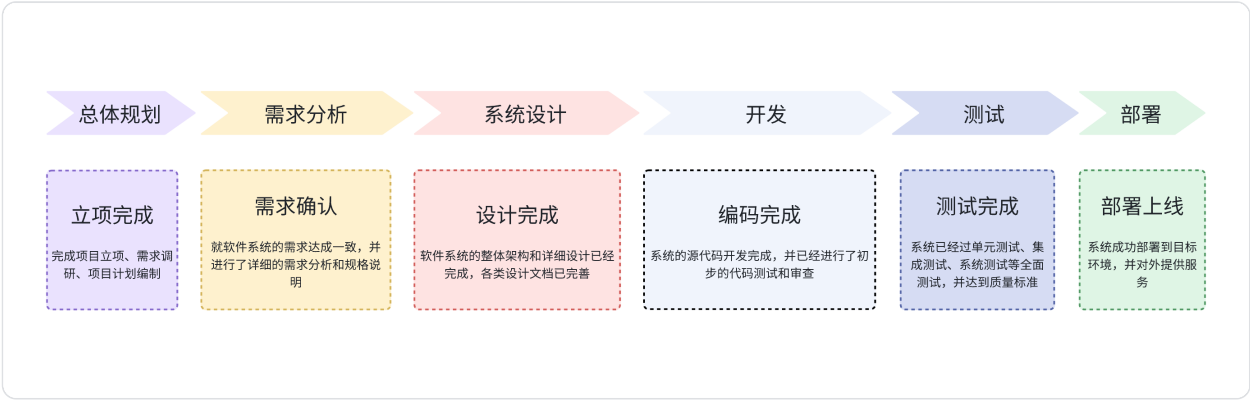
- a. **数据抓取和整合的自动化：**设计和开发高效的数据抓取和整合系统，自动从各种法律信息源中提取和整合数据。可以利用爬虫技术、数据清洗和结构化方法，结合本地数据库系统，来处理多样化的法律信息源。
- b. **搜索算法的设计和优化：**应用合适的搜索算法和排序策略，考虑法律领域的特点和用户需求，提高搜索结果的准确性和相关性。可以使用搜索引擎框架提供的高质量、高效率信息检索服务，结合用户反馈和评估指标进行系统的优化和调整。
- c. **优化AI模型的prompts预设：**为使系统能够调取ChatGPT等大语言模型接口，并借此实现关键词的提取进行辅助检索，需要设置合适的prompts预设，使得AI模型能够精确地理解项目需求，理解并生成高质量的与法律相关的检索关键词。

5. 项目计划

5.1 项目阶段与里程碑

参照软件开发的一般流程，我们对整个项目的实现阶段进行总体划分。为更好地测量每个阶段的实施与完成情况，进而把握总体项目进度，我们为每个阶段制定了“里程碑”。通过与预定的“里程碑”进行比较，可以评估项目是否按计划进行，并识别潜在的进度偏差。

为了更直观地表现项目阶段划分与里程碑，我们用以下的示意图展示：



项目阶段划分&里程碑示意图


在之后的整个项目开发过程中，我们将以此作为进度安排的总依据。

5.2 项目时间表

根据上面的项目总体阶段规划，我们列出如下的更详细的项目进度计划。下表记录了我们当前预期的各阶段事项及预期时间安排，后续会根据实际开发过程进行调整与维护。

事项	预期成果	预期开始时间	预期结束时间	备注
第一阶段：总体规划				
撰写项目计划书	产出项目计划书	2023/10/27	2023/10/30	 项目计划书
立项报告展示	完成立项展示ppt	2023/10/28	2023/10/31	
第二阶段：需求分析				
市场调研	调研报告	2023/10/31	2023/11/03	
用户意见收集	用户意见报告	2023/11/03	2023/11/05	
形成需求文档	需求文档	2023/11/06	2023/11/08	
明确定义系统功能和特性	功能文档	2023/11/09	2023/11/12	
第三阶段：系统设计				
设计系统架构	系统架构设计文档	2023/11/13	2023/11/16	
数据库设计	数据库设计文档	2023/11/16	2023/11/18	
后端接口设计	API文档	2023/11/18	2023/11/20	
前端界面设计	前端设计文档	2023/11/17	2023/11/20	
第四阶段：开发				
前端代码编写	前端子系统	2023/11/20	2023/12/13	
后端代码编写	后端子系统	2023/11/20	2023/12/13	
前后端整合	项目初步模型	2023/12/14	2023/12/16	
第五阶段：测试				
单元测试&集成测试	单元测试报告&集成测试报告	2023/12/16	2023/12/17	
功能测试	功能测试报告	2023/12/17	2023/12/18	
性能测试&安全测试	性能测试报告&安全测试报告	2023/12/18	2023/12/19	
用户体验测试	用户体验测试报告	2023/12/19	2023/12/20	
第六阶段：部署				
将项目部署到生产环境	部署文档	项目完成优化之后	项目完成优化之后	
验收测试&性能优化	验收测试报告	部署上线之后	部署上线之后	

5.3 团队成员和角色分配

序号	角色	职责	人员
1	项目经理	负责项目的总体规划、组织、协调和控制，按实际情况分配和安排各人员工作，确保项目按时、按质量要求完成。	 徐毕颖

2	业务分析师	负责市场调研和收集用户需求，将业务需求转化为可执行的软件需求，确定软件功能的具体范围。	 王梓轩
3	设计总监	建立系统框架，数据库设计，概要设计，技术评审	 章越
4	测试经理	组织编写测试计划和测试方案，组织系统测试；参加技术评审	 沈书豪
5	质量经理	带领质量监督组成员制定质量保证计划，对监督组反应的质量问题进行汇总，和产品经理以及项目经理进行交流沟通	 黄文杰
6	开发人员	负责进行编码工作和单元测试，进行系统集成，及时解决测试的时候出现的问题	全体人员
7	测试人员	编写测试方案和测试用例，进行系统测试，向开发组反馈 bug	全体人员
8	软件质量监督	实时对质量经理以及项目经理提供项目进度与项目实际开发时候的差异提出报告，指出差异原因和改进方法	全体人员

5.4 持续改进和监测策略

a. 代码审查和质量保证

采取尽量短的代码检查周期（如4天~1周），定期进行线下代码审查和质量保证活动，以确保代码的质量和组内成员对系统理解的一致性。通过检查代码、标识潜在问题和提供反馈，可以及早发现和解决质量问题，并且有利于不断优化代码结构。

b. 持续性能监测

实施持续性能监测，使用性能测试工具（如Apache JMeter/Tsung）和监测工具（如New Relic）对软件系统的性能进行监测和评估。这样可以及时发现性能瓶颈和问题，并采取相应的优化措施。

c. 用户反馈和用户测试

在系统开发的不同阶段，例如单个功能模块完成时，可适当邀请开发人员以外的用户进行功能试用，积极收集和分析用户的反馈（可采取问卷形式），包括用户体验、功能需求和问题报告。在此过程中，用户测试和用户参与可以提供开发人员以外的另一个视角，帮助发现软件系统的改进点，并针对用户需求进行调整。

d. 度量和指标

使用度量和指标来监测软件开发过程和产品质量。例如，追踪缺陷率、代码覆盖率、发布频率等指标，以评估项目的健康状况和改进方向。

e. 团队讨论和持续学习

开发过程中，定期进行团队讨论会议，回顾过去的工作和项目经验，识别问题和改进机会，并制定相应的行动计划。鼓励团队成员进行持续学习，寻找更优的技术实现手段，以提高开发能力和效率。

6. 项目成本分析

6.1 软硬件资源说明

软硬件资源说明	级别	详细配置	获取时间或方式	使用说明
开发应用服务器	普通	CPU&内存：2核 4GB SSD系统盘：100GB 带宽：7Mbps 流量包：1000GB/月	腾讯云购买	部署阶段使用
域名备案	普通	无	腾讯云购买	部署阶段使用
GPT API	关键	GPT-4.0	OpenAI官网购买	AI赋能功能需求
Visual Studio Code, IntelliJ IDEA	关键	符合主流版本要求	官网下载	开发阶段使用的IDE
Git	关键	版本：2.40.1以上	官网下载	代码托管&版本控制
浏览器&其他测试阶段需要用到的软件	普通	无	官网下载	前端效果预览&测试阶段使用

6.2 成本分析

6.2.1 人力资源成本

该垂直搜索引擎是一个B/S体系项目，用户端主要为网站。考虑到采用开源框架进行开发，项目工作量不多，适合小团体开发。本组相关人员有5人，已满足项目开发的需求。因人力成本的工资难以确定，我们在此以**工时**来确定人力资源成本。我们预计将有18个小时用来进行前期的培训和学习，30个工时用来进行代码的编写（开发），10个工时用来进行整合对接，8个工时用来进行沟通协调。

6.2.2 硬件成本

我们组预计将使用腾讯云服务器进行部署，配置要求为CPU（2核）、内存（4GB）、SSD系统盘（100GB）、带宽（7Mbps），参考了腾讯云官网的价格，该配置的服务器需要100元/月，购买域名（需要解析一个子域名：admin.）以及域名备案预计需要几十元。除了部署阶段的服务器开销以外，本项目暂无其他硬件方面的开销。

6.2.3 软件成本

本项目使用的所有软件均为开源且免费，都可以用学生身份体验到免费版本，所有本项目暂无软件成本的开销。

6.2.4 API成本

本项目预计将包含AI赋能功能，故需要调用GPT-4.0的API，参考官网的价格，对于8K context的模型，input（\$0.03/1k tokens），output（\$0.06/1k tokens）。考虑到开发&测试阶段可能使用的token数量以及部署上线后用户可能使用的token数量，该方面的开销预计为45元/月。

6.2.5 数据成本

该搜索引擎所涵盖的数据源将采用爬虫的方式从网络上获取，预计不会产生费用的开销。

7. 专题计划要点

7.1 开发人员培训计划

前端开发培训（React框架）：

- a. React基础知识
 - 了解React的概念和组件化开发。
 - JSX语法和元素渲染。
- b. 组件与状态管理
 - 创建和管理React组件。
 - 使用React状态管理工具，如Redux或Context。
- c. 路由和导航
 - React Router库的使用。
 - 前端路由和导航的最佳实践。
- d. 异步数据获取
 - 使用Fetch或Axios处理数据请求。

- 处理异步数据的组件渲染。

后端开发培训（Spring Boot框架）：

a. Spring Boot基础

- Spring Boot概述和特点。
- 项目创建和配置。

b. RESTful API开发

- 创建RESTful API端点。
- 请求处理和响应。

c. 数据库连接

- 使用MyBatis（MySQL）或Spring Data MongoDB。
- 数据库模型和操作。

d. 安全性和验证

- 使用Spring Security进行身份验证和授权。
- 数据验证和输入过滤。

数据库培训（MySQL或MongoDB框架）：

a. 数据库基础

- 数据库管理系统（DBMS）概述。
- 数据库设计原则。

b. MySQL或MongoDB

- MySQL：关系数据库管理系统。
- MongoDB：文档数据库管理系统。
- 数据库的创建、读取、更新和删除（CRUD）操作。

c. 数据库连接和ORM

- 连接数据库。
- 使用MyBatis（MySQL）或Spring Data MongoDB。

MyBatis框架：

a. MyBatis基础知识

- MyBatis的简介和优势。

- 配置文件和映射文件。
- b. 数据访问层设计
 - 如何设计数据访问层（DAO）。
 - MyBatis映射和数据操作。
- c. MyBatis配置
 - MyBatis配置文件的详细介绍。
 - 数据源配置。
- d. SQL映射
 - SQL语句的编写和映射到Java方法。
 - 参数传递和结果映射。
- e. 高级MyBatis功能
 - 动态SQL：使用if、choose、foreach等。
 - 缓存：MyBatis的缓存机制。

整合MyBatis到Spring Boot:

- a. Spring Boot与MyBatis
 - 如何在Spring Boot中整合MyBatis。
 - 配置数据源和事务管理。
- b. DAO层整合
 - 创建DAO接口。
 - 配置MyBatis的Mapper扫描器。
- c. 事务管理
 - Spring Boot中的事务管理。
 - 使用@Transactional注解。

7.2 质量保证计划

a. 质量目标

明确项目的质量目标，以确保项目能够满足用户需求和预期。

b. 质量标准

定义项目的质量标准和验收标准，以便进行检查和验证。

c. 质量控制措施

制定质量控制措施，以确保项目在开发过程中遵循最佳实践和标准。

- 代码审查：定期进行代码审查，以确保代码的质量、可读性和性能。
- 单元测试：编写全面的单元测试，以验证代码的正确性。
- 集成测试：进行集成测试，确保不同模块之间的协作无误。
- 功能测试：执行功能测试，验证系统的功能性。
- 性能测试：测试系统的性能，确保它可以承受实际负载。

d. 缺陷管理

建立缺陷管理流程，以跟踪、报告和修复项目中的问题。

- 问题跟踪系统：使用问题跟踪系统来记录和管理问题。
- 优先级分配：为问题分配优先级，以确保高优先级问题得到及时处理。
- 问题解决流程：定义问题解决流程，包括分配、修复、验证和关闭问题的步骤。

e. 文档管理

确保项目文档的完整性和准确性，以便项目成员和维护团队能够理解和维护系统。

- 文档编写：编写详细的技术文档，包括代码注释、使用手册和系统架构文档。
- 版本控制：使用版本控制系统，以管理和维护文档的版本。

f. 培训和知识传递

为项目团队提供必要的培训和知识传递，以确保他们了解项目的架构和技术。

- 培训计划：制定培训计划，涵盖项目中使用的所有关键技术和工具。
- 知识传递：确保知识传递，使新成员能够快速融入项目团队。

g. 审查和改进

定期审查项目的质量和流程，以持续改进项目的质量保证方法。

- 定期审查：制定定期审查计划，以评估项目的质量和流程。
- 改进措施：根据审查结果，制定改进措施，并持续优化项目的质量保证方法。

h. 验收和交付

在项目结束时，执行最终验收和交付，确保项目满足用户需求和质量标准。

- 验收测试：执行最终验收测试，确保项目达到预期目标。
- 用户培训：为最终用户提供必要的培训和支持。

i. 风险管理

管理项目的风险，以减轻潜在的问题对项目质量的影响。

- 风险评估：识别潜在的风险，并为其制定应对策略。

- 风险监控：定期监控风险，以确保它们不会影响项目的质量。

7.3 测试计划

7.3.1 静态测试

静态测试是通过分析或检查软件的语法、结构、过程、接口等来检查程序的正确性，可以发现参数不匹配、循环嵌套、未使用变量、空指针引用等问题。

在项目代码编写与测试过程中，前端逻辑部分采用 Typescript 编写，提供静态类型检查；同时将定期进行代码质量检查，检查组内各成员所编写的代码是否存在结构或者逻辑上的不合理之处，并对可能的问题进行及时讨论（如有必要）和修正，以减少后期测试与修改bug的成本。

7.3.2 单元测试

单元测试是对软件中最小可测试单元（通常是函数、方法或类）进行测试的过程。它旨在验证单元的行为是否符合预期，并检测潜在的错误或缺陷。

单元测试将主要在本项目的代码编写阶段由开发人员完成，使用测试框架和工具进行测试，以尽早发现和纠正单元级别的错误和缺陷，确保单元的独立功能正确性；同时，测试人员在测试阶段进一步完善单元测试，寻找可能遗漏的 bug。

7.3.3 系统测试

系统测试是将已经测试过的子系统组装成一个完整系统进行测试，以验证系统能否按照系统设计的要求提供指定的功能，确保最终软件系统达到了产品需求。

我们将系统测试放在测试阶段完成，此时主体代码已经编写完成，可以由测试人员使用不同的测试技术和方法，例如功能测试、性能测试、安全测试等，模拟真实的用户环境，对整个软件系统的功能、性能和稳定性进行全面测试。

8. 参考文献

- [1]包来军,白志秀,周丽.河北省法律类网站建设研究——以河北省司法厅官网、河北长安网、河北法制网为例[J].产业与科技论坛,2018,17(06):229-230.
- [2]南嘉鱼.2017法律咨询服务平台TOP30[J].互联网周刊,2017(19):50.
- [3]姜宇航.法律网站服务功能的分析与探究[J].法制博览(中旬刊),2014(12):265-266.
- [4]陈暄.网络环境下法律信息资源的检索策略与利用[J].科技信息,2010(23):46+45.
- [5]吴志鸿.美国法律信息资源发展现状与启示[J].现代情报,2009,29(09):211-215.
- [6]罗晗骞,石月鑫.人工智能(AI)技术在搜索引擎中的应用[J].电子技术与软件工程, 2020(21):2.
- [7]余淼,杨丹,赵俊芹.垂直搜索引擎的关键技术研究[J].软件导刊, 2007(12):3.

[8] 《软件设计文档国家标准》

[9] 《Software Requirements》Edition 2