

NPULearn 需求文档

版本: v0.2.0 | 更新日期: 2025 年 4 月
适用平台: Windows、macOS、Linux、Android

目录

1. 项目概述	3
1.1. 产品定位	3
1.2. 产品愿景	3
1.3. 目标用户	3
2. 功能需求	3
2.1. 核心功能模块	3
2.1.1. AI 聊天助手	3
2.1.1.1. 功能描述	3
2.1.1.2. 具体需求	3
2.1.2. 富文本渲染系统	4
2.1.2.1. 功能描述	4
2.1.2.2. 具体需求	4
2.1.3. 智能工具协作系统	4
2.1.3.1. 功能描述	4
2.1.3.2. 具体需求	4
2.1.4. 文档处理系统	5
2.1.4.1. 功能描述	5
2.1.4.2. 具体需求	5
2.1.5. 个性化学习系统	5
2.1.5.1. 功能描述	5
2.1.5.2. 具体需求	5
2.2. 辅助功能模块	6
2.2.1. 主题系统	6
2.2.1.1. 功能描述	6
2.2.1.2. 具体需求	6
2.2.2. 历史记录管理	6
2.2.2.1. 功能描述	6
2.2.2.2. 具体需求	6
2.2.3. 设置管理系统	7
2.2.3.1. 功能描述	7
2.2.3.2. 具体需求	7
3. 非功能性需求	8
3.1. 性能需求	8
3.1.1. 响应时间	8
3.1.2. 资源占用	8
3.1.3. 并发处理	8
3.2. 可用性需求	8
3.2.1. 易用性	8
3.2.2. 无障碍性	8
3.2.3. 多语言支持	8
3.3. 兼容性需求	8
3.3.1. 操作系统	8

3.3.2. 硬件要求	9
3.4. 安全性需求	9
3.4.1. 数据安全	9
3.4.2. 隐私保护	9
3.4.3. 网络安全	9
3.5. 可维护性需求	9
3.5.1. 代码质量	9
3.5.2. 文档要求	9
3.5.3. 更新机制	9
4. 质量属性	10
4.1. 可靠性	10
4.2. 可扩展性	10
4.3. 用户体验	10
4.4. 技术创新	10
5. 约束条件	10
5.1. 技术约束	10
5.2. 法律约束	10
5.3. 商业约束	10
5.4. 时间约束	11
6. 验收标准	11
6.1. 功能验收	11
6.2. 性能验收	11
6.3. 用户体验验收	11
6.4. 兼容性验收	11
7. 项目里程碑	11
7.1. 第一阶段：核心功能开发	11
7.2. 第二阶段：功能完善	11
7.3. 第三阶段：智能优化	12
7.4. 第四阶段：生态建设	12
8. 风险评估	12
8.1. 技术风险	12
8.2. 市场风险	12
8.3. 资源风险	12
8.4. 缓解策略	12
9. 总结	12

1. 项目概述

1.1. 产品定位

NPULearn 是一个基于人工智能的智能学习助手桌面应用，旨在为用户提供全面的学习支持和知识获取体验。通过集成多种 AI 模型和学习工具，为不同层次的学习者创造个性化、高效且易用的智能学习环境。

1.2. 产品愿景

致力于成为教育技术领域的创新引领者，通过 AI 技术打破知识获取的壁垒，让高质量的教育资源惠及更多人群，推动知识民主化和个性化学习的深度实践。

1.3. 目标用户

- **学生群体：**大中小学生、研究生等各阶段学习者
- **教育工作者：**教师、培训师、教育内容创作者
- **自学者：**终身学习者、职业技能提升者
- **科研人员：**需要快速查询和分析学术资料的研究者
- **专业人士：**需要持续学习新知识的各行业从业者

2. 功能需求

2.1. 核心功能模块

2.1.1. AI 聊天助手

2.1.1.1. 功能描述

提供智能对话交互功能，支持多种 AI 模型，为用户提供个性化学习指导和问题解答。

2.1.1.2. 具体需求

1. 多 AI 模型支持

- 支持 DeepSeek、Gemini、Claude 等主流 AI 模型
- 根据查询类型智能推荐最适合的 AI 模型
- 支持用户手动切换 AI 模型

2. 流式响应处理

- 实时显示 AI 回复内容，如同真人对话
- 支持中断正在进行的对话生成
- 流式响应过程中的错误恢复机制

3. COT（思维链）功能

- 可视化 AI 推理过程，显示思考步骤
- 支持折叠/展开思维链内容
- 提供透明的决策过程展示

4. 上下文管理

- 智能维护对话历史和上下文
- 支持长对话场景的上下文压缩
- 保留重要信息的同时控制上下文长度

5. API 密钥管理

- 支持多个 API 密钥配置
- 自动轮换机制提高可用性
- 安全的密钥存储和加密

2.1.2. 富文本渲染系统

2.1.2.1. 功能描述

提供强大的富文本内容渲染能力，支持多种格式的内容展示和交互。

2.1.2.2. 具体需求

1. Markdown 渲染

- 完整的 GitHub 风格 Markdown 支持
- 代码语法高亮 (highlight.js)
- 自定义 CSS 样式支持

2. 数学公式渲染

- MathJax 支持 LaTeX 数学公式渲染
- KaTeX 支持快速数学表达式渲染
- 行内和块级公式显示

3. 图表渲染

- Mermaid 图表支持 (流程图、时序图等)
- Pintora 图表渲染
- 动态图表生成和交互

4. 文档排版

- Typst 现代化文档排版系统
- 高质量数学排版
- 多语言文档支持

5. 渲染优化

- 异步渲染避免界面阻塞
- 批量处理多个渲染请求
- 缓存机制避免重复渲染
- 主题适配 (亮/暗主题)

2.1.3. 智能工具协作系统

2.1.3.1. 功能描述

集成多种专业工具，通过智能编排为用户提供一站式问题解决方案。

2.1.3.2. 具体需求

1. Wolfram Alpha 集成

- 支持复杂数学计算和科学计算
- 自动格式化计算结果展示
- 相关查询推荐和一键查询

2. 工具链编排

- 自动识别用户需求并选择合适工具
- 支持并行工具执行提升效率
- 智能合并多个工具的输出结果

3. Python 代码执行

- 支持在线 Python 代码执行
- 安全的代码执行环境
- 结果可视化展示

4. 知识库查询

- 快速检索相关知识内容
- 智能推荐相关主题
- 多源知识整合

2.1.4. 文档处理系统

2.1.4.1. 功能描述

智能处理多种格式的文档，提供文档分析、总结和问答功能。

2.1.4.2. 具体需求

1. 文档格式支持

- PDF 文件读取和解析
- Word 文档（.docx）处理
- 文本文件（.txt, .md 等）处理

2. 文档上传方式

- 拖拽上传文档文件
- 点击按钮选择文件
- 支持批量文档处理

3. 智能内容分析

- 自动提取文档关键信息
- 生成文档摘要和要点
- 识别文档结构和主题
- 评估内容复杂度

4. 文档问答

- 基于文档内容回答问题
- 引用原文支持答案
- 多文档关联分析

2.1.5. 个性化学习系统

2.1.5.1. 功能描述

根据用户学习行为和偏好，提供个性化的学习内容和建议。

2.1.5.2. 具体需求

1. 学习风格识别

- 识别视觉、听觉、动手实践等学习偏好

- 分析用户学习行为模式
- 动态调整内容呈现方式
- 2. **知识水平评估**
 - 评估用户在不同领域的知识水平
 - 识别知识空隙和薄弱环节
 - 提供针对性学习建议
- 3. **内容个性化**
 - 根据学习风格调整内容难度
 - 生成个性化练习和测试
 - 推荐相关学习资源
- 4. **学习路径规划**
 - 制定个性化学习计划
 - 动态调整学习目标
 - 跟踪学习进度和效果

2.2. 辅助功能模块

2.2.1. 主题系统

2.2.1.1. 功能描述

提供多样化的界面主题选择，适应不同用户偏好和使用场景。

2.2.1.2. 具体需求

1. 主题切换

- 亮色主题和暗色主题
- 自动跟随系统主题设置
- 手动切换主题选项

2. 界面适配

- 所有渲染内容自动适配主题
- 保持界面一致性和美观性
- 护眼模式支持

3. 个性化设置

- 自定义主题颜色
- 字体大小调节
- 界面密度设置

2.2.2. 历史记录管理

2.2.2.1. 功能描述

智能管理用户的对话历史和学习记录，提供便捷的查找和回顾功能。

2.2.2.2. 具体需求

1. 历史记录存储

- 自动保存所有对话内容

- 本地安全存储用户数据
- 支持数据导出和备份
- 2. **智能分组**
 - 按时间自动分组历史记录
 - 按主题智能分类对话
 - 标签系统支持自定义分类
- 3. **快速检索**
 - 全文搜索历史对话
 - 关键词高亮显示
 - 智能推荐相关历史
- 4. **记录管理**
 - 删除不需要的历史记录
 - 收藏重要对话内容
 - 批量操作支持

2.2.3. 设置管理系统

2.2.3.1. 功能描述

提供全面的应用设置管理，包括 API 配置、界面设置和用户偏好。

2.2.3.2. 具体需求

1. **API 密钥配置**
 - 多个 AI 服务商密钥管理
 - 密钥有效性验证
 - 安全的加密存储
2. **界面设置**
 - 主题偏好设置
 - 字体大小和类型选择
 - 界面布局自定义
3. **功能配置**
 - 默认 AI 模型选择
 - COT 功能开关
 - 自动保存设置
4. **数据管理**
 - 设置导入导出
 - 恢复默认设置
 - 清理应用数据

3. 非功能性需求

3.1. 性能需求

3.1.1. 响应时间

- AI 对话响应延迟 < 3 秒（不含 AI 模型处理时间）
- 文档渲染时间 < 2 秒
- 界面操作响应时间 < 500ms
- 应用启动时间 < 5 秒

3.1.2. 资源占用

- 内存占用 < 500MB（正常使用）
- CPU 占用 < 20%（空闲状态）
- 磁盘空间占用 < 200MB（安装包）

3.1.3. 并发处理

- 支持多个 AI 请求并发处理
- 异步渲染不阻塞用户界面
- 流式响应实时更新

3.2. 可用性需求

3.2.1. 易用性

- 直观的用户界面设计
- 渐进式功能揭示
- 一键完成复杂操作
- 智能默认设置

3.2.2. 无障碍性

- 支持键盘导航
- 屏幕阅读器兼容
- 高对比度模式
- 大字体显示选项

3.2.3. 多语言支持

- 中文界面（简体/繁体）
- 英文界面支持
- 国际化框架准备

3.3. 兼容性需求

3.3.1. 操作系统

- Windows 10/11 (x64)
- macOS 10.15+ (Intel/Apple Silicon)
- Linux（主流发行版）
- Android 8.0+（移动端）

3.3.2. 硬件要求

- 最低内存：4GB RAM
- 推荐内存：8GB RAM
- 硬盘空间：1GB 可用空间
- 网络连接：稳定的互联网连接

3.4. 安全性需求

3.4.1. 数据安全

- API 密钥 AES 加密存储
- 用户数据本地化存储
- 敏感信息不上传云端
- 定期安全漏洞扫描

3.4.2. 隐私保护

- 明确的数据使用政策
- 用户数据控制权
- 可选的数据匿名化
- 遵循数据保护法规

3.4.3. 网络安全

- HTTPS 加密通信
- 安全的 API 调用
- 输入内容安全验证
- 防护恶意攻击

3.5. 可维护性需求

3.5.1. 代码质量

- 模块化架构设计
- 完整的单元测试覆盖
- 代码静态分析
- 持续集成/部署

3.5.2. 文档要求

- 完整的 API 文档
- 用户使用手册
- 开发者指南
- 问题排查文档

3.5.3. 更新机制

- 自动检查更新
- 增量更新支持
- 版本回滚机制
- 平滑升级体验

4. 质量属性

4.1. 可靠性

- 系统可用性 $\geq 99.5\%$
- 故障恢复时间 < 1 分钟
- 数据一致性保证
- 优雅的错误处理

4.2. 可扩展性

- 插件系统架构
- 新 AI 模型接入支持
- 第三方工具集成
- 功能模块热插拔

4.3. 用户体验

- 认知负荷最小化
- 直观的交互设计
- 及时的操作反馈
- 个性化适配能力

4.4. 技术创新

- 多 AI 模型融合策略
- 智能工具协作机制
- CoT 思维链技术
- 自适应 UI 系统

5. 约束条件

5.1. 技术约束

- 基于 Tauri + Vue 3 + TypeScript 架构
- Rust 后端处理核心逻辑
- 跨平台原生应用要求
- 开源组件优先选择

5.2. 法律约束

- 遵循各 AI 服务商使用条款
- 符合开源许可证要求
- 满足数据保护法规
- 知识产权合规使用

5.3. 商业约束

- 免费提供核心功能
- 可持续的开发模式
- 社区驱动的发展方向
- 开源生态贡献

5.4. 时间约束

- MVP 版本开发周期：3 个月
- 主要功能迭代周期：1 个月
- 稳定版本发布周期：6 个月
- 长期支持版本：1 年

6. 验收标准

6.1. 功能验收

- 所有核心功能正常运行
- AI 模型切换和调用成功
- 文档处理和渲染正确
- 用户设置保存和恢复

6.2. 性能验收

- 满足响应时间要求
- 资源占用在可接受范围
- 并发处理能力达标
- 稳定性测试通过

6.3. 用户体验验收

- 用户满意度评分 $\geq 4.0/5.0$
- 任务完成率 $\geq 90\%$
- 学习曲线平缓
- 错误率 $< 5\%$

6.4. 兼容性验收

- 主流操作系统正常运行
- 不同硬件配置适配
- 多分辨率屏幕支持
- 无障碍功能可用

7. 项目里程碑

7.1. 第一阶段：核心功能开发

- AI 聊天助手基础功能
- 基本富文本渲染
- 简单文档处理
- 基础设置管理

7.2. 第二阶段：功能完善

- CoT 思维链集成
- Wolfram Alpha 工具集成
- 完整渲染系统
- 主题系统实现

7.3. 第三阶段：智能优化

- 个性化学习系统
- 智能工具协作
- 性能优化
- 用户体验优化

7.4. 第四阶段：生态建设

- 插件系统开发
- 社区功能建设
- 文档完善
- 多平台适配

8. 风险评估

8.1. 技术风险

- AI 模型 API 稳定性风险
- 跨平台兼容性问题
- 性能优化挑战
- 第三方依赖风险

8.2. 市场风险

- 竞品技术快速发展
- 用户需求变化
- AI 技术发展方向变化
- 法律法规变化

8.3. 资源风险

- 开发团队规模限制
- API 调用成本控制
- 服务器和基础设施成本
- 长期维护资源需求

8.4. 缓解策略

- 建立技术储备和备选方案
- 持续用户反馈收集
- 灵活的架构设计
- 开源社区生态建设

9. 总结

NPULearn 作为智能学习助手，致力于通过 AI 技术创新为用户提供高质量的学习体验。本需求文档详细阐述了产品的功能需求、非功能性需求、质量属性和约束条件，为项目开发提供了清晰的指导方向。

通过模块化设计、用户中心的产品理念和开放的技术架构，NPULearn 将成为教育技术领域的创新产品，推动个性化学习和知识民主化的发展。