

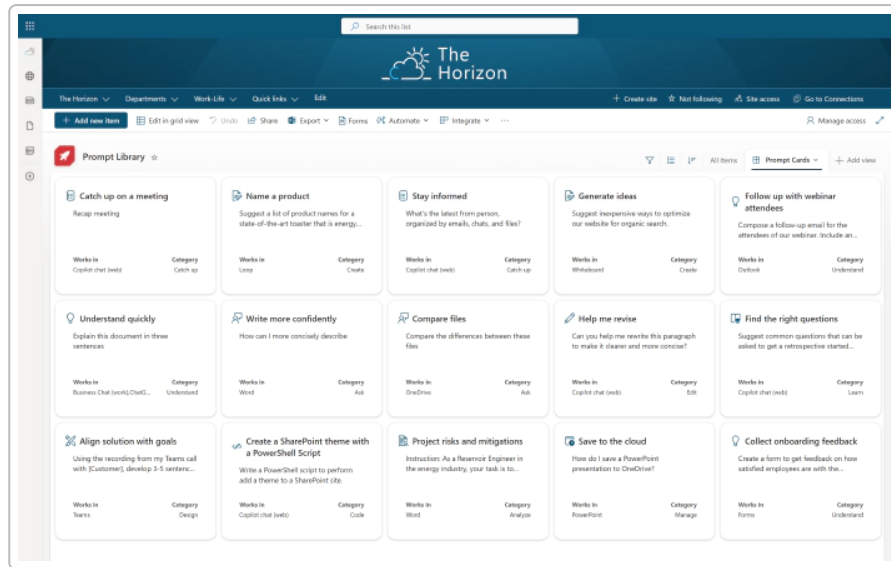
精简版提示词收集管理平台方案

为应对提示词管理的挑战，本方案针对仅收集、存储和管理提示词的需求进行设计。正如 Agenta 所述，将提示词散落在代码或多份文档中，会导致无法跟踪历史版本、难以统一管理¹。为此，我们建立一个**集中式提示词库**，支持按照项目、团队或主题分组管理，并为每条提示词记录必要信息。每个提示词条目可包含**标题、完整提示文本、描述/备注**，以及**分类标签和适用模型/场景**等元数据²。例如，可以定义“提示类别”和“适用场景”字段，帮助用户对提示词进行分类和筛选。此外，建议提供**示例输入输出**或补充说明，帮助用户理解提示用途³。

- **集中式提示词库**：集中管理所有提示词条目，避免分散存储。提示词库支持按项目、团队和主题创建不同列表或目录，便于协作与权限划分。每个提示词记录包括标题、正文、标签、作者、创建时间等关键信息²。如微软指南所示，可为提示词添加标题、完整文本及分类标签，以实现可视化和便捷的管理²。用户还可自定义关键字段（如备注、示例）以丰富提示内容³。
- **版本控制**：对每次提示词编辑都生成版本记录，形成线性历史，支持查看修改日志和文本差异，以及必要时回退到旧版本⁴⁵。提示词的版本管理类似软件开发中的版本控制，通过语义化版本号等方式（例如 X.Y.Z 格式）清晰标识重大改动、功能新增或修复⁶。系统应提供差异比较功能，直观显示各版本间的变化⁷，并允许用户恢复到某个历史版本，保证提示词库的安全性和可追溯性⁴⁵。
- **标签与分类**：支持用户为提示词添加自定义标签（如场景、模型、功能等），并提供多维度的过滤和搜索功能。借鉴 Langfuse 数据模型，每个提示词可绑定多个标签，所有版本共享这些标签，便于在界面或 API 中按标签快速筛选⁸。同时应支持基于关键词、创建者、日期等条件的组合检索。通过标签分类，用户可以快速定位满足特定场景或模型要求的提示词，提高检索效率⁸。
- **协作与权限**：引入角色和审批流程以保证内容质量。系统定义**管理员、编辑者、查看者**等角色，不同角色拥有不同操作权限。只有管理员或通过审核的编辑者才能发布新提示词或修改现有提示词，以避免未经验证的更改造成风险。如 Agenta 所述，分工合作是最佳实践：存储和版本化提示词可以产生完整的审计日志和回滚能力⁹。每次新增、修改或删除操作都应记录日志，记录变更内容、时间和执行者，便于后续审计和问题排查¹⁰⁹。这种基于角色的访问控制和审批流程能够确保只有经过审核的提示词更新才会发布，维护提示库的一致性和可靠性¹⁰⁹。
- **导入与导出**：提供批量导入导出功能，以便在不同环境之间迁移或进行备份。系统应支持从 CSV/JSON 文件导入提示词，也支持将提示库导出为 JSON/CSV 格式以供版本控制和离线归档¹¹。例如，用户可以将当前提示库快照导出为文件并托管到 Git 仓库，实现备份和审计¹¹。导入功能则可加速在新项目或环境下初始化提示库，提高灵活性。

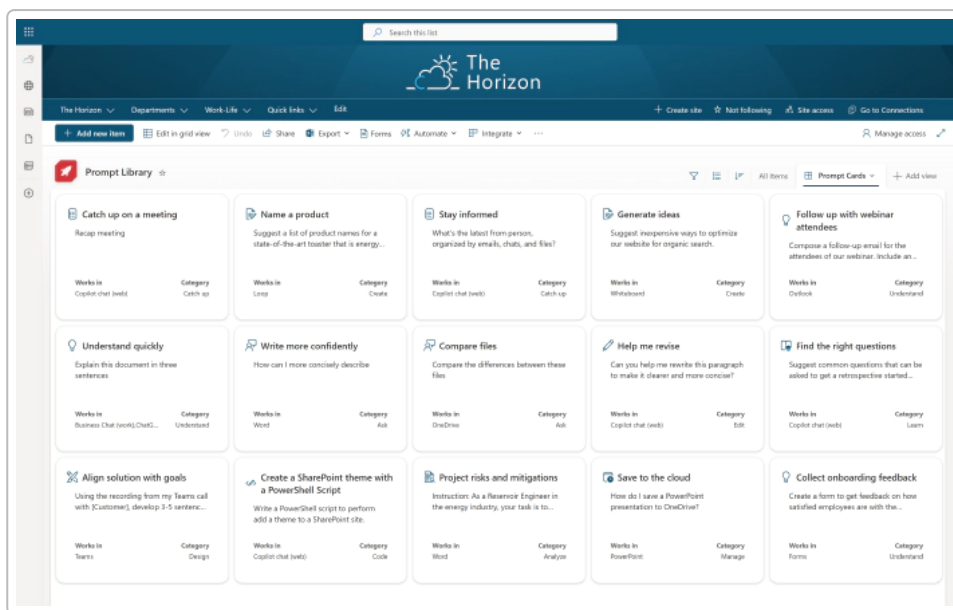
技术架构建议

- **前端**：建议采用 React 或 Vue 等现代前端框架，构建简洁、响应式的界面。界面布局可使用列表和卡片相结合的方式展示提示词数据。例如可以为提示词列表创建“提示卡片”视图，使用画廊模式将提示词以卡片形式展示¹²。如图所示，每个提示词卡片可包含标题、分类、适用场景等信息，用户可在卡片上快速浏览并点击查看详情



。前端还应支持表单输入、标签筛选、全文检索等功能，提供良好的用户体验¹²。

- **后端**：可选用 Node.js + Express 或 Python + Flask 等轻量级框架，实现 RESTful API 接口。后端负责处理用户请求、执行业务逻辑、与数据库交互等。对于版本控制，可在数据库层面为提示词更新生成新的记录，或使用专门的版本控制库实现这一逻辑。在设计时需考虑并发安全和事务处理，确保同时编辑时数据一致。
- **数据库**：PostgreSQL 和 MongoDB 都是合适的选择。PostgreSQL 支持 JSON/JSONB 字段，可用于存储提示词内容及其历史版本，同时提供全文搜索功能。MongoDB 则更灵活地存储文档型数据，适合存储多变的标签或元数据结构。无论哪种数据库，应设计两大核心表（或集合）：**Prompt**（存放提示词元数据和最新版本引用）与**PromptVersion**（存放每个版本的详细内容）。此外，需有标签表和用户角色表以支持过滤和权限管理。
- **存储**：对于提示词示例输入/输出文件、Markdown 文档等附件，可使用对象存储（如 AWS S3）或数据库二进制字段进行存储。需要时也可保留在数据库中，但推荐使用文件存储提高扩展性。
- **接口**：采用 RESTful 风格的 API 提供增删改查及批量操作能力。例如，GET /prompts 列出提示词，POST /prompts/import 批量导入，GET /prompts/{id}/versions 查看历史版本等。批量导入导出接口应支持 CSV/JSON 格式，并提供验证与错误提示。



示例界面使用卡片布局展示提示词元素，用户可在每张卡片上查看提示词的标题、类别等信息，提高浏览效率¹²。前端 UI 应力求简洁大方，采用响应式设计适配不同终端。可以结合现成的 UI 库（如 Material-UI、Ant Design）来加速开发和提升美观度。

实施步骤

- 需求梳理与数据模型设计：**分析提示词管理的需求，设计数据库模型。确定**提示词**实体（包括标题、文本、标签、创建者、创建时间等字段）以及关联的**版本记录表**（包括版本号、内容、修改者、修改时间等）。同时设计标签表、用户表及权限表，明确它们之间的关系。根据业务需求定义角色权限和工作流规则。
- 基础 CRUD 与版本管理：**实现提示词的增删改查接口。当创建新提示词时，插入初始版本；每次编辑提示词时，在版本表中插入新记录，同时更新提示词表的最新版本引用⁴。后端逻辑需确保版本号自动递增，并保存修改日志。提供查看历史版本和比较版本差异的功能，便于用户审阅编辑记录⁴⁵。
- 搜索、过滤与标签：**实现全文检索和多条件过滤功能。利用数据库的全文搜索或集成搜索引擎（如 Elasticsearch）支持按关键词搜索提示文本。标签系统应允许用户为提示词打上多个标签（场景、模型等），前端提供复选过滤器。参考 Langfuse 的做法，对提示词进行标签分类，用户可通过标签快速筛选符合条件的提示词⁸。界面应支持按标签、作者、创建日期等条件组合查询。
- 权限与审批流程：**实现基于角色的访问控制（RBAC）。在后端检查用户角色，只有具有编辑权限的用户才能提交更改，且可以配置审核流程：新提示词或编辑需要管理员审核通过后才正式发布。所有操作（创建、修改、删除）都需记录操作日志，包括执行者、时间和变更内容，形成完整的审计轨迹¹⁰⁹。如遇冲突修改，可通过版本回退或管理员手动干预解决，以保证系统稳定。
- 导入导出功能：**实现 CSV/JSON 格式的批量导入导出接口。用户可以通过导出功能将当前提示库的所有条目下载为文件（CSV 或 JSON），作为版本快照或备份¹¹。导入功能则根据文件内容创建或更新提示词，有助于跨项目迁移或初始化提示库。此功能也便于将提示词置于代码版本控制系统，从而实现额外备份和变更管理¹¹。

6. **测试与部署**：为保证质量，应编写单元测试和集成测试，覆盖提示词管理的各项功能。测试包括接口测试、权限验证以及导入导出等场景。部署方面，可以使用 Docker 容器化应用，以便于部署和扩展。采用 CI/CD 管道自动化构建、测试和部署，提高开发效率和发布稳定性。

使用场景

该平台定位于**个人和小团队**使用，旨在帮助用户快速收集并管理有效提示词。个人用户可将平时遇到或整理的优秀提示词记录其中，便于日后查找。团队项目中，成员可共享通用提示词并协同编辑，以提升工作效率。由于目标用户注重简单快速的体验，界面设计应美观直观，避免复杂冗余的功能模块。该平台无需先进的训练或评测功能，而是提供轻量级的提示词版本控制与分享机制，使用户能够专注于提示设计本身。

参考文献：如 [Latitude 博客](#) 所示，清晰的版本管理和变更记录对于提示词协同至关重要⁵；Agenta 强调集中管理提示词可带来回滚与审计能力⁹；Langfuse 指出通过标签对提示词分类，可大幅提升检索效率⁸；Langfuse 社区讨论也建议支持 CSV/JSON 导入导出，以便备份和迁移¹¹。上述实践和工具都为本方案设计提供了有益参考。

¹ ⁷ ⁹ The Definitive Guide to Prompt Management Systems

<https://agenta.ai/blog/the-definitive-guide-to-prompt-management-systems>

² ¹² Build a prompt library with Microsoft Lists

<https://pnp.github.io/blog/post/build-a-prompt-library-with-microsoft-lists/>

³ ⁵ ⁶ ¹⁰ Prompt Versioning: Best Practices

<https://latitude-blog.ghost.io/blog/prompt-versioning-best-practices/>

⁴ Tag a prompt | Phoenix

<https://arize.com/docs/phoenix/prompt-engineering/how-to-prompts/tag-a-prompt>

⁸ Langfuse Prompts Data Model - Langfuse

<https://langfuse.com/docs/prompt-management/data-model>

¹¹ Allow for export and import of prompts · langfuse · Discussion #6894 · GitHub

<https://github.com/orgs/langfuse/discussions/6894>