

项目策划案：Project WOK (Words to Knowledge)

副标题：AI 时代的个性化知识炼金枢纽

摘要：文字 ➡ 知识图谱

组织AI回答中的知识（虽然下面主要说的是AI浏览器但也应该支持对所有文字的储存）

1. 执行摘要 (Executive Summary)

当今世界，我们正被“AI 答案”的洪流所淹没。以 AI 浏览器为代表的新一代工具，能为我们提供即时、精炼的回答，但这也带来了新的、更深层次的问题：我们如何将这些“一次性”的答案，转化为“永久性”的个人知识？

现有的知识管理工具，要么像“数字仓库”（如 Obsidian），需要用户花费大量精力（复制、粘贴、手动分类、建立链接），“入库摩擦力”极大；要么像“云端剪报”（如 Readwise），将用户锁定在昂贵的订阅服务中，且更侧重于“复习”而非“创造”。

Project WOK 是一个创新的桌面端知识管理解决方案。 它由一个智能浏览器插件和一个本地知识库应用组成。

WOK 的核心使命是：彻底消除“信息消费”与“知识构筑”之间的鸿沟。

它允许用户在浏览 AI 答案时，通过**一步操作**（选中文字 → 右键导入），将最小的“知识元”（一个词、一句话）精准捕获到本地。随后，应用内置的 **AI 管家**（基于 Python 构建）将自动为这条知识智能添加标签、**建立关联**，并将其保存在用户自己的电脑上，最终以“知识图谱”的形式，将用户所有的零散智慧“连接成网”。

Project WOK 不是另一个笔记软件，它是一个“AI 驱动的知识自动化引擎”，旨在成为用户在 AI 时代中不可或缺的“第二大脑”。

2. 我们面临的问题：AI 时代的“知识鸿沟”

在 AI 浏览器成为主流的今天，一个典型用户的“知识内化”流程是断裂且低效的。

场景复现（问题所在）：

1. **获得答案：** 用户向 AI 浏览器提问（例如：“介绍一下 2013 年的‘缩减恐慌’”），并得到了一个完美的总结。
2. **发现价值：** 用户在答案中发现了一个关键概念（例如：“美联储主席伯南克释放结束债购计划的信号”）。
3. **“知识搬运”的痛苦（鸿沟出现）：**
 - **方法A (手动复制)：** 用户必须：选中 → 复制 (Ctrl+C) → 切换到笔记软件 (如 Obsidian) → 创建新笔记 → 粘贴 (Ctrl+V) → 手动思考并输入标签 (如 `#财经` `#美联储`) → 思考它和别的笔记有什么联系 → 手动添加 `[[链接]]` ...
 - **方法B (网页剪藏)：** 用户使用传统“剪藏”插件，将整个网页保存下来。
4. **恶果：**
 - 在方法 A 中，高昂的“**入库摩擦力**”导致用户最终放弃了记录，宝贵的灵感瞬间流失。
 - 在方法 B 中，用户的知识库变成了“**数字垃圾场**”，堆满了未经提炼的网页，知识无法连接，最终“死”在仓库里。

核心痛点： 我们缺少一个工具，能以极低的摩擦力、极高的精度，捕获最小单位的知识，并自动化后续的整理与连接工作。

3. 我们的解决方案：Project WOK

Project WOK 通过一个“两件套”的组合，完美解决了上述痛点。

- **部件一：“魔法捕手”（浏览器插件）**
 - 它是一个轻量级的浏览器插件。当用户在任何网页（尤其是 AI 浏览器的答案页）上选中一段文字时，只需单击右键，在二级菜单中选择“导入到 WOK”。
 - (zcanic 强调) 仅此一步。用户的任务到此结束，可以继续专注浏览。
- **部件二：“知识炼金室”（Tauri 桌面应用）**
 - 这是用户在本地电脑上安装的“知识大本营”。
 - 当“魔法捕手”发射信号后，“炼金室”会在后台自动完成以下所有工作：
 1. **接收：** 立即接收到那段文字。
 2. **理解：** （核心功能）应用内置的 AI 管家（由 Python 驱动）会立刻“阅读”这段文字。
 3. **加工：** AI 会自动为它推荐并打上标签。（例如，捕获“伯南克释放结束债购信号”，AI 自动添加 `#美联储` `#缩减恐慌` `#2013年`）
 4. **存储：** 将这条知识连同标签，保存为本地电脑上的一个干净文件（我们采用开放的 Markdown 格式，数据永远属于用户）。
 5. **连接：** AI 会检查这个新知识点与你已有的知识库存在哪些共同标签或概念，并自动建立它们之间的“连接”。

4. 核心功能与用户体验

我们将为用户提供三个维度的革命性体验：

功能一：“零摩擦”的知识捕获

- **体验：** 如丝般顺滑。用户在阅读时的“心流”不会被打断。
- **对比：** 传统笔记软件的“入库摩擦力”如果需要 60 秒，Project WOK 将其降到了 1 秒。这会带来知识捕获数量和质量的指数级增长。

功能二：“AI 驱动”的自动整理

- **体验：** 智能且贴心。用户不再是“档案管理员”，而是“知识鉴赏家”。
- **对比：** 传统工具需要用户“手动分类”，这是一种滞后的、高认知负担的工作。Project WOK 提供的“AI 自动标签”是即时的、智能的，它甚至能发现用户自己都未曾察觉的分类维度。

功能三：“可视化”的知识图谱

- **体验：** 极具启发性。我们最核心的界面不是“笔记列表”，而是“知识星图”。
- **展示：** 用户的每一条知识都是一颗“星星”。当用户捕获的新知识（新星星）飞入时，它会因为 AI 自动打上的标签，而自动与已有的星星（旧知识）连接成“星座”。
- **价值：** 用户可以直观地看到“缩减恐慌”和自己昨天保存的“AI 芯片产业”笔记，因为它们都连接到了 #财经 这个标签上。**WOK** 的核心价值不是“记录”，而是“启发”——帮助用户连接看似无关的知识，从而创造新价值。

5. 市场定位与竞品分析

Project WOK 的生态位是独一无二的，它不与任何人正面竞争，而是填补了市场中最关键的那个空白。

- **AI 浏览器 (如 Perplexity) - [上游]**
 - 它们是“矿井”，负责生产信息。但它们不负责提炼和保存。
- **Obsidian / Logseq - [平行]**

- 它们是“本地金库”，提供了强大的知识连接功能。但它们没有“AI 自动入库”的能力，用户体验极度繁琐，是一个需要“DIY 拼装”的“平台”。
- **Readwise / Reader - [邻近]**
 - 它们是“云端仓库”，捕获体验很好。但它们是昂贵的订阅制 (SaaS)，数据不在本地，且核心是“复习”，而非“图谱连接”。

Project WOK 的精准定位：

我们是连接“矿井（AI 浏览器）”和“金库（知识库）”的“AI 自动化精炼厂”。

- **我们 VS Obsidian：**我们是“开箱即用”的产品，而不是“需要配置”的平台。我们用 AI 自动化了 Obsidian 用户 80% 的体力劳动（捕获与打标签）。
- **我们 VS Readwise：**我们是“本地优先、买断制”的，数据 100% 属于用户。我们是“AI 驱动的创新工具”，而不仅仅是“复习工具”。

6. 项目可行性与团队配置

这是一个高可行性的创新项目，其成功建立在一个革命性的“人机协同”开发模型之上。

1. 技术选型（大脑）：

- 我们将使用 **Pytauri** 架构。这意味着我们的核心“AI 大脑”将使用 **Python** 构建。
- **优势：**这是我们团队的核心强项。我们可以利用 Python 无与伦比的 AI 生态，构建出业界领先的“自动标签”和“语义连接”引擎。

2. 技术选型（外观）：

- 应用外壳将使用 **Tauri**，确保应用体积小、速度快、跨平台 (Windows/Mac)。前端界面将使用成熟的 Web 技术（如 React,

Milkdown) 构建。

3. 团队与“AI 生产线”（我们的秘密武器）：

- 本项目的“前端开发”（界面和插件）对我们（假设只有 Python 经验）来说本是挑战。但我们将采用“AI 赋能”的流水线模式：
 - **AI 开发工具 (Claude/Codex)：** 作为我们的“初级程序员”，它们负责接收“微型任务包”并生成高质量的前端代码。
-

7. 总结与愿景

Project WOK 不仅仅是一个“效率工具”，它是一个“认知增强器”。

在信息爆炸的 AI 时代，胜利不属于“知道最多”的人，而属于“连接最快”的人。Project WOK 的愿景，就是成为用户最贴身的“知识炼金师”，自动地、智能地将每天流入的庞杂信息，提炼、重组、连接为只属于用户自己的、闪闪发光的智慧网络。