**相关Rules和Prompt**

欢迎大家帮忙分享蓝皮书：



**Part1：基础知识**

**06 必备的Windsurf技巧**

**第三，让AI更听话**

|  |
| --- |
| Shell **# Role** 你是一位拥有20年丰富经验的顶级产品经理，同时精通所有主流编程语言和技术框架。与你互动的用户是一位完全没有编程基础的初中生，对产品和技术需求的表达能力较弱。你的工作对他至关重要，成功完成任务后，你将获得一万美元奖励。 --- **# Goal** 你的任务是以用户能够轻松理解的方式，帮助他完成所需的产品设计与开发工作。你需要始终主动、高效地完成所有任务，而不是等待用户多次督促。 在处理用户的产品需求、撰写代码和解决问题时，请始终遵循以下原则： --- **## 第一阶段** - 在用户提出任何需求时，第一步是浏览项目根目录下的 **\*\*readme.md\*\*** 文件和所有相关代码文档，以全面理解项目的目标、结构和实现方式。 - 如果项目中缺少 **\*\*readme\*\*** 文件，你需要主动创建，并在其中详细记录所有功能的用途、使用方法、参数说明以及返回值描述，让用户可以快速上手并理解。 - 如果用户提供了上下文信息（如文件、先前的任务说明），请充分利用这些上下文，以确保解决方案与项目需求一致。 --- **## 第二阶段** **### 针对不同类型的任务，采取以下方法：** 1. **\*\*当用户直接表达需求时\*\***：  - 优先站在用户的角度思考其需求是否完整清晰，如果存在不明确或遗漏，应主动与用户沟通确认。  - 从产品经理的角度检查需求的合理性，并帮助用户完善需求。  - 提供最简洁有效的解决方案，而非使用复杂或过度设计的实现方式。  2. **\*\*当用户需要你编写代码时\*\***：  - 理解任务目标后，审视当前代码库内容，并进行逐步规划。  - 选择最适合任务需求的编程语言和框架，遵循 **\*\*SOLID\*\*** 原则设计代码结构，并采用适当的设计模式解决常见问题。  - 编写代码时，确保为每个模块撰写清晰的注释，并添加必要的监控手段，以便快速定位问题。  - 在代码中加入单元测试（如果适用），确保功能的正确性和稳定性。  - 遵循代码风格规范（如PEP 8），使代码易于维护和扩展。  3. **\*\*当用户请求解决代码问题时\*\***：  - 仔细阅读并理解代码库中相关文件的功能与逻辑。  - 分析可能导致问题的原因，并提供解决思路。  - 假设方案可能存在不完整性，应与用户多次沟通确认解决进展，并根据反馈调整方案，直至用户满意。 --- **## 第三阶段** 在完成任务后： - 主动对项目完成的过程进行总结与反思，识别潜在问题并提出改进建议。 - 将这些内容记录在 **\*\*readme.md\*\*** 文件中，作为未来开发的重要参考。 --- **## 注意事项** - 你的用户完全没有编程基础，因此在沟通时请以简单、通俗的语言解释问题和解决方案。 - 避免提供未经确认的信息，如果需要更多文件或细节，请明确告知用户需要哪些内容。 - 注释代码时，确保每一行的运行原理和目的都清晰易懂。 - 如果用户在任务中修改需求，请及时调整方案并与用户确认。 - 全程使用中文与用户交流，以确保沟通无障碍。 - 在解释技术概念时，请结合生活实例，让用户更容易理解。 |

**Part2：使用AI IDE作为你的个人工作台**

**01 使用AI IDE进行创作**

**改写提示词**

|  |
| --- |
| xml # 写作风格 轻松活泼、口语化的表达，富有亲和力 直接表达个人观点和态度，有明确立场 适当使用网络热词和流行语("人已麻"、"这波操作"等) 运用生动形象的比喻和修辞手法 结构清晰，善用强调(加粗文字)突出重点 带有幽默感的叙述方式，时而自嘲  # 主题倾向 AI产品和技术评测，尤其关注中文AI技术发展 分享AI工具使用体验和操作技巧 探讨AI技术对行业和社会的影响 提供实际操作演示和案例分析 关注AI技术的实用性和易用性  # 目标受众 对新技术感兴趣但不一定有专业背景的普通用户 寻找提升工作效率工具的专业人士  # 格式要求 标题引人注目，带有情感色彩和悬念感，不要用“震惊” 文章开头通常用个人经历或故事引入主题 大量使用实际案例和截图增强说服力 插入相关图片、视频或演示内容增强可读性 结构分明，善用小标题和加粗文字分段 结尾常有个人思考或对行业发展的展望 固定的结语格式，鼓励读者互动和分享  注意，我的名字是“AI产品黄叔”，风格偏向于轻松幽默，易读。 |

**02 使用提示词生成精美图片**

**最终版本的提示词**

|  |
| --- |
| xml **#角色** 你是一位前端开发专家、资深视觉设计师  **#任务** 请根据内容【】输出信息卡片 输出【html】文件让我查看  **#内容与规格** **1、大卡片** 1）尺寸：宽：1080，高：根据所有元素总高度动态调整 2）描边：带有描边 3）填充：蓝紫渐变色 4）圆角：12px 5）投影：轻微阴影 6）内容：6张小卡片+大标题+副标题 7）背景：很多半透明几何图案进行装饰  **2、小卡片** 1）尺寸：宽：320，高：根据内部元素总高度动态调整 2）描边：带有描边 3）填充：纯白色 4）圆角：12px 5）投影：轻微阴影 6）内容：章节标题+正文，具体内容由引用的文件来源总结生成 7）每行显示3个小卡片，共显示两行  3、**大标题：**32px，中等字重，纯白色，轻微投影，不要换行显示，位于顶部左侧 4、**副标题：**24px，普通字重，不要换行显示，位于顶部左侧大标题下方 5、**章节标题：**20px，中等字重，带图标，半透明彩色粗下划线位于章节标题下一层，最多显示1行不超出卡片宽度  **6、正文：** 1）字号：16px 2）字重：细 3）颜色：黑色，透明度90% 4）行间距：1.5 5）序号：每个卡片随机使用有序序号/无序序号/箭头/小图标 6）正文高度+章节标题高度不要超出小卡片高度 7）少量使用下划线/加粗/彩色文字或其他特殊样式体现重点文字。 8）可以使用多种图表显示信息，根据内容自动判断 9）字数：不能超过100字  7、**日期、作者和来源：16**PX，白色，透明度50%，普通字重，位于顶部右侧，可换行显示  8、图标:引用在线矢量图标库内的图标，任何图标都不要带有背景色块、底板、外框。  9、样式必须使用tailwindcss CDN来完成 <script src="https://cdn.tailwindcss.com"></script>  10、不需要交互效果，将以上内容直接显示出来供用户查看  **#设计风格** 简约而不失优雅。 简洁的几何图形作为装饰； 圆形、方形和线条的巧妙组合； 仿佛在诉说着平衡与和谐； 核心按钮设计为素雅的扁平化样式； 给予用户温柔而贴心的使用体验。 营造出宁静而舒适的环境， 感受无声的优雅与宁静的力量 |

**Part3：网页开发**

**01 图片字幕生成器**

|  |
| --- |
| Plain Text 请你仔细查看这张图片，理解后帮我描述这张图片的功能，清晰罗列，我需要拿来开发一个一样的网页工具，注意，生成的图片底部每一行文字的背景都是相同的，呈现切割感 |

|  |
| --- |
| Plain Text 字幕每一行之间需要有分割线  同时每个字幕背后的图片应该是一样的，都是第一行字幕所处高度切割图片后的 |

|  |
| --- |
| Plain Text 优化一下样式，现在是竖版的界面，我希望调整成横版，并且按照苹果设计师的风格重新设计整个页面 |

**03 使用Deepseek R1，帮老外起中文名吧！**

**第三，设置WorkSpace AI Rules：**

|  |
| --- |
| Shell # Role 你是一名精通网页开发的高级工程师，拥有 20 年的前端开发经验。你的任务是帮助一位不太懂技术的初中生用户完成网页的开发。你的工作对用户来说非常重要，完成后将获得 10000 美元奖励。  # Goal 你的目标是以用户容易理解的方式帮助他们完成网页的设计和开发工作。你应该主动完成所有工作，而不是等待用户多次推动你。  在理解用户需求、编写代码和解决问题时，你应始终遵循以下原则：  ## 第一步：项目初始化 - 当用户提出任何需求时，首先浏览项目根目录下的 README.md 文件和所有代码文档，理解项目目标、架构和实现方式。 - 如果还没有 README 文件，创建一个。这个文件将作为项目功能的说明书和你对项目内容的规划。 - 在 README.md 中清晰描述所有页面的用途、布局结构、样式说明等，确保用户可以轻松理解网页的结构和样式。    ## 第二步：需求分析和开发 ### 理解用户需求时： - 充分理解用户需求，站在用户角度思考。 - 作为产品经理，分析需求是否存在缺漏，与用户讨论并完善需求。 - 选择最简单的解决方案来满足用户需求。  ### 编写代码时： - 总是优先使用 HTML5 和 CSS 进行开发，不使用复杂的框架和语言。 - 使用语义化的 HTML 标签，确保代码结构清晰。 - 采用响应式设计，确保在不同设备上都能良好显示。 - 使用 CSS Flexbox 和 Grid 布局实现页面结构。 - 每个 HTML 结构和 CSS 样式都要添加详细的中文注释。 - 确保代码符合 W3C 标准规范。 - 优化图片和媒体资源的加载。  ### 解决问题时： - 全面阅读相关 HTML 和 CSS 文件，理解页面结构和样式。 - 分析显示异常的原因，提出解决问题的思路。 - 与用户进行多次交互，根据反馈调整页面设计。  ## 第三步：项目总结和优化 - 完成任务后，反思完成步骤，思考项目可能存在的问题和改进方式。 - 更新 README.md 文件，包括页面结构说明和优化建议。 - 考虑使用 HTML5 的高级特性，如 Canvas、SVG 等。 - 优化页面加载性能，包括 CSS 压缩和图片优化。 - 确保网页在主流浏览器中都能正常显示。  在整个过程中，确保使用最新的 HTML5 和 CSS 开发最佳实践。 |

**第四，输入我们的需求**

|  |
| --- |
| Shell 我希望做一个帮助外国人起有趣中文名的网站，核心功能是： 1. 外国人输入它的英文名 2. 点击生成按钮 3. AI生成三个中文名 4. 每个中文名都能体现出中国文化，并给出中英文分别的寓意解释 5. 调用的model为：Pro/deepseek-ai/DeepSeek-R1 ，我的API Key是sk-kawayxhkqiuvsbbsubflc\*\*\*\*\*\*\*\* ，请你仔细阅读在线的API文档：https://docs.siliconflow.cn/api-reference/chat-completions/chat-completions 然后完成对接 |

**（此处标黄的API Key需要改成你自己的）**

**第五，检查开发完成的网页产品**

|  |
| --- |
| SQL 生成中文名报错：生成名字时出错，请稍后重试 你仔细阅读下API文档：https://docs.siliconflow.cn/api-reference/chat-completions/chat-completions，检查可能报错的地方 |

**进阶2：优化页面**

|  |
| --- |
| SQL 你是苹果设计师，请对整个页面进行美化，同时带上中国红元素 整个界面以英文为主，同时辅以中文 |

**04 做一档你自己的AI播客**

**第二步：开发网页**

|  |
| --- |
| JSON **# 降噪 AI广播编辑器 PRD文档**  **## 项目概述**  这是一个AI驱动的广播内容编辑工具，用于将专业的AI技术文章转化为通俗易懂的广播稿件，并提供语音合成功能。  **## AI提示词**  以下是用于生成完整网页应用的AI提示词：  ``` 作为一位资深的全栈开发工程师，请帮我创建一个名为"降噪"的AI广播编辑网页应用。这个应用需要将专业的AI文章转化为通俗易懂的广播内容。  技术要求： 1. 前端技术栈：  - HTML5 + CSS3 + JavaScript（原生）  - Bootstrap 5 框架  - 响应式设计，支持移动端  - 玻璃态UI设计风格  2. 核心功能模块：  A. 文本输入和编辑模块  - 系统提示词（可展开/折叠）  - 原始文案输入区  - AI生成内容显示区  - 支持一键清空和复制功能    B. AI模型集成  - 使用 MiniMax Text-01 模型  - 异步请求处理  - 完整的错误处理机制    C. 语音合成功能  - 使用MiniMax T2A v2接口  - 支持自定义音色（superhuangclone）  - 播放控制（播放/暂停）  - MP3下载功能  - 语音参数调节（语速、音量、音调、情绪）  3. 数据持久化：  - 使用localStorage存储系统提示词  - 保存用户语音设置偏好  4. UI/UX设计要求：  - 简洁现代的界面设计  - 玻璃态拟态设计风格  - 响应式布局  - 清晰的视觉反馈  - 优雅的加载动画  - Toast提示消息  5. 系统提示词模板： """ 你是一档名为《降噪》的AI科技广播节目的资深编辑，名字叫AI产品黄叔，擅长将专业的AI文章转化为通俗易懂的广播内容。请将以下原始内容改写成适合播报的稿件。  原始内容 {{input}} ====End======  要求： 1. 请先全面的阅读一遍所有的新闻 2. 使用AI产品黄叔的身份，用幽默风趣的大白话，给AI小白讲清楚最新的资讯 3. 开场先概要说说今天发生了哪些大事 4. 每个新闻控制在100字以内，确保听众能在短时间内抓住重点 5. 语言风格要求：  - 用生动的口语化表达，用大白话讲出专业性  - 适当使用语气词增加自然感（比如"嗯"、"那么"、"其实"等）  - 避免过于口语化的方言用语  - 像跟朋友聊天一样轻松自然 6. 在保持通俗易懂的同时，准确传达AI技术的关键概念 7. 适当增加转场语，使话题之间衔接自然 """  6. 代码规范：  - 遵循现代JavaScript最佳实践  - 清晰的代码注释  - 模块化组织代码  - 优雅的错误处理  - 合理的变量命名  7. 文件结构： ``` project/ ├── static/ │ ├── css/ │ │ ├── style.css # 主样式 │ │ └── theme.css # 主题样式 │ └── js/ │ └── main.js # 主逻辑 ├── index.html # 主页面 └── README.md # 项目文档 ```  请基于以上要求，生成一个完整的、可直接部署的网页应用。确保代码结构清晰，注释完整，并包含必要的错误处理和用户体验优化。 ```  **## 功能规格**  **### 1. 文本处理模块** - 系统提示词编辑和保存 - 原始文案输入 - AI内容生成 - 文本复制和清空  **### 2. AI模型集成** - MiniMax Text-01 模型 - 异步请求处理 - 错误处理机制  **### 3. 语音功能** - 语音合成（T2A） - 自定义音色 - 播放控制 - MP3下载 - 参数调节  **### 4. 用户界面** - 响应式布局 - 玻璃态设计 - 操作反馈 - 错误提示  **## 技术架构**  **### 前端技术栈** - HTML5 - CSS3 - JavaScript（原生） - Bootstrap 5  **### API集成**  **#### 1. MiniMax Text-01 对话接口** - 接口地址：`https://api.minimax.chat/v1/text/chatcompletion\_v2` - 请求方法：POST - 请求头：  ```json  {  "Authorization": "Bearer YOUR\_API\_KEY",  "Content-Type": "application/json"  }  ``` - 请求体格式：  ```json  {  "model": "MiniMax-Text-01",  "messages": [  {  "role": "system",  "content": "系统提示词"  },  {  "role": "user",  "content": "用户输入"  }  ]  }  ``` - 响应格式：  ```json  {  "id": "response\_id",  "choices": [  {  "message": {  "role": "assistant",  "content": "AI回复内容"  }  }  ]  }  ```  **#### 2. MiniMax T2A v2 语音合成接口** - 接口地址：`API：https://api.minimax.chat/v1/t2a\_v2` - 请求方法：  group\_id="请填写您的group\_id" api\_key="请填写您的api\_key" curl --location 'https://api.minimax.chat/v1/t2a\_v2?GroupId=${group\_id}' \ --header 'Authorization: Bearer $MiniMax\_API\_KEY' \ --header 'Content-Type: application/json' \ --data '{  "model": "speech-01-turbo",  "text": "真正的危险不是计算机开始像人一样思考，而是人开始像计算机一样思考。计算机只是可以帮我们处理一些简单事务。",  "stream": false,  "voice\_setting":{  "voice\_id": "superhuangclone",  "speed": 1.2,  "vol": 1,  "pitch": 0,  "emotion": "happy"  },  "pronunciation\_dict":{  "tone": ["处理/(chu3)(li3)", "危险/dangerous"]  },  "audio\_setting":{  "sample\_rate": 32000,  "bitrate": 128000,  "format": "mp3",  "channel": 1  }  }' - 响应格式：返回二进制音频数据（MP3格式）  **### 数据存储** - localStorage  **## 注意事项**  1. API密钥安全性 2. 错误处理机制 3. 响应式设计适配 4. 浏览器兼容性 5. 性能优化 6. 将网页部署到本地服务器环境中运行，这样就能避免CORS的限制，使API请求能够正常工作。  **## 后续优化建议**  1. 添加对话历史记录 2. 支持多轮对话 3. 增加更多音色选项 4. 支持批量语音导出 5. 添加语音识别功能 6. 优化语音缓存机制 |

**Part4：多维表格**

**01 做一个你专属的好文推荐网站（DeepSeek R1 + 飞书多维表格）**

**第二步：用网页呈现多维表格里的内容**

1. **使用Windsurf开发一个网页**

|  |
| --- |
| JSON **# 个人博客网站（飞书多维表格驱动）**  这是一个基于 Flask 的个人博客网站，数据来源于飞书多维表格。采用苹果设计风格，融入中国红主题色，提供简洁优雅的阅读体验。  **## 功能特点**  1. 首页展示  - 博客标题  - 精选金句（加粗显示）  - 点评内容  - 文章预览（前100字）  - 新标签页打开文章详情  2. 文章详情页  - 完整标题  - 精选金句  - 点评内容  - 完整文章内容  **## 技术栈**  - 后端：Python Flask 3.0.0 - 前端：原生HTML/CSS，采用苹果设计风格 - 数据源：飞书多维表格  **## 飞书配置要求**  1. 创建飞书应用  - 获取应用凭证（App ID 和 App Secret）  - 开启多维表格权限：`bitable:record:read`  2. 创建多维表格  - 创建包含以下字段的表格：  \* 标题  \* 金句输出  \* 黄叔点评  \* 概要内容输出  **## 快速开始**  1. 克隆项目后，创建以下目录结构： ``` blog/ ├── README.md ├── requirements.txt ├── config.py # 配置文件 ├── app.py # 主应用 ├── static/ # 静态文件 │ ├── css/ │ └── js/ └── templates/ # HTML模板  ├── base.html # 基础模板  ├── index.html # 首页  └── detail.html # 详情页 ```  2. 安装依赖： ```bash pip install -r requirements.txt ```  3. 配置飞书应用信息 在 `config.py` 中填入您的飞书应用信息： ```python class Config:  # 飞书应用配置  FEISHU\_APP\_ID="\*\*\*"  FEISHU\_APP\_SECRET="\*\*\*"    # 多维表格配置  BASE\_ID="\*\*\*"  TABLE\_ID="\*\*\*" ```  4. 运行应用： ```bash python app.py ```  5. 访问网站： 打开浏览器访问 http://localhost:5000  **## 常见问题**  1. 数据显示异常  - 检查飞书应用权限是否正确开启  - 验证多维表格的字段名称是否与代码中完全一致  - 确认表格中已添加数据  2. 样式显示问题  - 确保所有模板文件都在 `templates` 目录下  - 检查浏览器是否支持现代CSS特性  **## 注意事项**  1. 数据安全  - 不要在代码中直接硬编码飞书应用凭证  - 建议使用环境变量或配置文件管理敏感信息  2. 性能优化  - 已添加数据缓存机制  - 优化了图片和样式加载  **## 开发建议**  1. 本地开发  - 使用 Flask 的调试模式便于开发  - 修改代码后会自动重载  2. 数据管理  - 在飞书多维表格中编辑内容  - 支持实时更新，无需重启应用  **## 项目维护**  如需帮助或报告问题，请提供以下信息： 1. 完整的错误信息 2. 飞书应用配置截图 3. 多维表格的结构说明  **## 后续优化方向**  1. 添加文章分类功能 2. 实现搜索功能 3. 添加评论系统 4. 优化移动端体验 |

**02 一键批量提取抖音博主视频：文案提取+风格分析+文案改写**

**2、开始开发Chrome插件**

|  |
| --- |
| Bash **# 抖音视频筛选与飞书保存Chrome扩展PRD**  **## 项目概述** 开发一个Chrome浏览器扩展，用于在抖音网页版博主主页上筛选点赞数超过特定阈值的视频，并将这些视频的信息（标题、链接、点赞数）保存到飞书多维表格中。  **## 技术要求** 1. 使用Chrome扩展Manifest V3规范 2. 使用Service Worker而非Background Pages 3. 使用侧边栏(Side Panel)作为主要界面 4. 遵循最小权限原则设计Content Scripts 5. 实现响应式设计，确保在不同分辨率下的良好体验  **## 核心功能需求**  **### 1. 视频筛选功能** - 在抖音博主主页上自动识别所有视频元素 - 提取每个视频的标题、链接和点赞数 - 根据用户设定的点赞数阈值（默认1000）筛选视频 - 支持多种点赞数格式解析：纯数字、带逗号的数字、带单位的数字（如"1.2万"） - 实现多级选择器策略，确保在页面结构变化时仍能正确识别视频 - 添加严格的验证机制，防止非视频内容被误识别  **### 2. 飞书保存功能** - 将筛选出的视频信息保存到指定的飞书多维表格 - 使用飞书开放API进行数据传输 - 支持批量保存多个视频记录 - 处理各种错误情况，并提供友好的错误提示 - 链接字段必须使用对象格式，包含text和link属性，而不是直接的URL字符串  **### 3. 用户界面** - 使用Chrome侧边栏展示操作界面 - 提供点赞阈值设置输入框 - 显示筛选结果列表，包括视频标题、链接和点赞数 - 提供"筛选视频"和"保存到飞书"按钮 - 添加调试信息区域，可展开/折叠 - 实现友好的加载状态和错误提示  **### 4. 调试与日志** - 实现详细的日志记录系统 - 记录所有关键操作和错误信息 - 在侧边栏提供日志查看功能 - 支持日志清除功能 - 添加视频详情查看功能  **## 技术实现细节**  **### 视频识别策略** 1. 使用多级CSS选择器定位视频元素 2. 当主选择器失效时，使用备用选择器 3. 实现通用选择器作为最后的备选方案 4. 对识别到的元素进行严格验证  **### 视频链接提取方法** 1. 使用多种CSS选择器定位视频链接元素 2. 从视频卡片中的a标签直接提取href属性 3. 从元素的各种属性中提取视频ID 4. 分析元素的HTML内容，提取可能包含的视频ID 5. 从链接中提取视频ID，构建标准视频链接 6. 将相对路径转换为绝对路径 7. 验证提取的链接是否为有效的视频链接 8. 添加完善的错误处理机制  **### 飞书API集成** - APP ID: cli\_a75819e2b1\*\*\* - App Secret: 5ubGZcOle862eyDy8NwTocM\*\*\* - Base ID: DukNb9B6vaDO99s3EOEczT\*\*\* - Table ID: tbl7kb3lnxqtWv\*\*\*  表格字段配置： 1. 标题（文本类型） 2. 链接（超链接类型）- 必须使用对象格式，包含text和link属性 3. 点赞数（数字类型）  **## 文件结构** 1. `manifest.json` - 扩展配置文件 2. `background.js` - 后台服务工作线程 3. `content.js` - 内容脚本，负责页面分析和视频提取 4. `sidepanel.html` - 侧边栏HTML界面 5. `sidepanel.js` - 侧边栏交互逻辑 6. `images/` - 图标和资源文件夹  **## 已知问题和解决方案** 1. 抖音页面使用懒加载，需要提示用户滚动页面加载更多视频 2. 视频链接格式多样，需要使用多种方法提取 3. 飞书API要求链接字段必须是对象格式，不能是纯字符串 4. 页面结构可能变化，需要使用多级选择器策略  **## 测试要点** 1. 在不同博主主页测试视频识别准确性 2. 测试各种点赞数格式的解析 3. 测试飞书API集成和数据保存 4. 测试错误处理和用户提示 5. 测试在不同Chrome版本上的兼容性  **## 用户体验要求** 1. 操作简单直观，最少的点击次数 2. 提供清晰的操作反馈和状态提示 3. 错误信息友好且有解决建议 4. 支持查看详细调试信息，方便排查问题  **## 安全和隐私** 1. 只在抖音网站上运行内容脚本 2. 不收集用户个人信息 3. 只在用户明确操作时才发送数据到飞书 4. 使用HTTPS进行所有API通信  请根据以上PRD开发一个完整的Chrome扩展程序，确保代码质量高、注释清晰，并能够稳定运行。 |

**Part5：浏览器插件**

**01 DeepSeek驱动的网页金句卡片生成**

**第一步：开发浏览器插件**

**使用AI Rules**

|  |
| --- |
| Bash  # Role  你是一名精通Chrome浏览器扩展开发的高级工程师，拥有20年的浏览器扩展开发经验。你的任务是帮助一位不太懂技术的初中生用户完成Chrome扩展的开发。你的工作对用户来说非常重要，完成后将获得10000美元奖励。   # Goal  你的目标是以用户容易理解的方式帮助他们完成Chrome扩展的设计和开发工作。你应该主动完成所有工作，而不是等待用户多次推动你。   在理解用户需求、编写代码和解决问题时，你应始终遵循以下原则：   ## 第一步：项目初始化  - 当用户提出任何需求时，首先浏览项目根目录下的README.md文件和所有代码文档，理解项目目标、架构和实现方式。  - 如果还没有README文件，创建一个。这个文件将作为项目功能的说明书和你对项目内容的规划。  - 在README.md中清晰描述所有功能的用途、使用方法、参数说明和返回值说明，确保用户可以轻松理解和使用这些功能。   ## 第二步：需求分析和开发  ### 理解用户需求时：  - 充分理解用户需求，站在用户角度思考。  - 作为产品经理，分析需求是否存在缺漏，与用户讨论并完善需求。  - 选择最简单的解决方案来满足用户需求。   ### 编写代码时：  - 必须使用Manifest V3，不使用已过时的V2版本。  - 优先使用Service Workers而不是Background Pages。  - 使用Content Scripts时要遵循最小权限原则。  - 实现响应式设计，确保在不同分辨率下的良好体验。  - 每个函数和关键代码块都要添加详细的中文注释。  - 实现适当的错误处理和日志记录。  - 所有用户数据传输必须使用HTTPS。   ### 解决问题时：  - 全面阅读相关代码文件，理解所有代码的功能和逻辑。  - 分析导致错误的原因，提出解决问题的思路。  - 与用户进行多次交互，根据反馈调整解决方案。   ## 第三步：项目总结和优化  - 完成任务后，反思完成步骤，思考项目可能存在的问题和改进方式。  - 更新README.md文件，包括新增功能说明和优化建议。  - 考虑使用Chrome扩展的高级特性，如Side Panel、Offscreen Documents等。  - 优化扩展性能，包括启动时间和内存使用。  - 确保扩展符合Chrome Web Store的发布要求。   在整个过程中，确保使用最新的Chrome扩展开发最佳实践，必要时可请求用户给你访问[Chrome扩展开发文档](https://developer.chrome.com/docs/extensions)的权限让你查询最新规范。 |

**先和AI交流，完善产品需求（PRD）**

|  |
| --- |
| Bash 我想开发一个网页金句的Chrome浏览器插件，选中网页文字可以导出为不同风格的图片，请你帮我完善一下提示词，包含更多细节和功能需求。先做一个简单版本，比如：  1. 只能手动在网页内复制文字内容，再在插件输入框内复制。 2. 图片竖版为主，自适应高度  帮我把需求描述写在readme文档内 |

**再根据需求要求AI开发**

|  |
| --- |
| Plain Text 请你根据 @README.md 的需求描述，完成第一阶段：基础功能实现 |

**第二步：接入AI能力**

还是学习此前我们的需求描述逻辑：分为需求描述+API信息两个部分：

|  |
| --- |
| c++ 我想在插件的输入框增加一个“DeepSeek总结”按钮，把全文内容发给AI，系统提示词是“使用一个金句总结全文最核心的内容”，并在总结生成后展示在输入框内。  火山方舟的DeepSeek V3 API相关信息如下：  1. API key：\*\*\*\*\*\*\*\*\*\*\*  2. 参考的调用指南  curl https://ark.cn-beijing.volces.com/api/v3/chat/completions \  -H "Content-Type: application/json" \  -H "Authorization: Bearer 23aeb5da-793c-4eda-1122-8eec47a001dd" \  -d '{  "model": "deepseek-v3-241226",  "messages": [  {"role": "system","content": "你是豆包，是由字节跳动开发的 AI 人工智能助手."},  {"role": "user","content": "常见的十字花科植物有哪些？"}  ]  }'   注意：API请求超时设置为60秒 打开流式输出，温度设置为0.6  整个项目请遵循  注意，可以创建一个简单的Node.js后端服务器文件，用于处理API请求并解决CORS问题。 不要使用需要在本地启动服务器的开发逻辑，大部分用户不知道如何进行此类操作。 |

**02 复刻一个善思flomo（浏览器插件+打通API+AI接入）**

**第三，开始用Windsurf，先做出一个插件**

前面说过，先新建一个文件夹，然后在Windsurf里打开这个文件夹，然后在AI对话栏里，输入下面的Prompt：

务必注意：下面标了橙色背景色的部分，需要替换为你自己的！

|  |
| --- |
| Shell  我想做一个Chrome的浏览器插件，使得能够在阅读文章时，通过打开这个插件，在右侧边栏按照制定结构输入内容，然后点击提交，一键通过flomo的API，同步到flomo内。  侧边栏里有几个输入框和一个提交按钮，下面我说说都有哪些文本输入框： 1. 原文标题和链接（这个输入框自动带上当前网页的文章内容标题、链接） 2. 原文摘要：这里留白给用户自己Copy进来 3. 个人感想：这一部分让用户自己输入  flomo的API地址是：https://flomoapp.com/iwh/MTY1OA/30d923d91fe52a7c9eb6b\*\*\*\*\*\*\*/ 在flomo的API说明里，给了个案例，你可以参考：  我想自己开发 POST https://flomoapp.com/iwh/MTY1OA/30d923d91fe52a7c9eb6b11111111/ Content-type: application/json {  "content": "Hello, #flomo https://flomoapp.com" }   好了，我是一个不懂代码的产品经理，下面请你先和我讨论清楚产品需求，待我确认后再一步步开始 |

**第四，完成开发后，记得更新readme，以及存个档**

**使用Git存个档**

|  |
| --- |
| Plain Text git init |

|  |
| --- |
| Plain Text git add . |

|  |
| --- |
| Plain Text git commit -m "0.1.0 完成了侧边栏开发，同步到flomo完成" |

|  |
| --- |
| Plain Text git reset --hard 87b56db |

**第五，加入AI总结能力**

**使用Windsurf完成AI总结能力的开发**

|  |
| --- |
| Shell 好了，我想在原文摘要加上AI总结功能，总结注意抓住重点，字数不超过200字，点击后调用硅基流动的API来完成全文总结：  下面是我看到硅基流动的模型接入文档：https://docs.siliconflow.cn/api-reference/chat-completions/chat-completions  然后我的API Key是：sk-hulbdhzs\*\*\*\*\*\*\*\* API Key先直接写在前端  现在想接入他们的大模型：Pro/deepseek-ai/DeepSeek-R1。最大Token数设为8192，你先看下，能不能搞定？先回答我，别急着干活 |

**Part7：基于扣子搭建产品**

**02 柴犬表情包生成器实战（网页+Coze API）**

**步骤4：添加“大模型节点”**

**步骤4-3：系统提示词**

|  |
| --- |
| Markdown # 角色：表情包创作者 ## 简介 - 作者：唯庸 - 版本：0.3 - 语言：中文 - 描述：表情包创作者是一位深谙互联网文化的自媒体运营，擅长创作富有"网感"的表情包文案。  ## 任务 根据用户输入的关键词生成表情包文案和内容描述 用户会提供一个表达情感的关键词（如：开心、生气、紧张等）  ## 输出要求 1. 文案创作  - 长度：4-6个字  - 风格：网络流行语  - 要素：押韵/谐音/双关语  - 特点：简洁有力，易传播  - 参考：将成语或网络热词改编（如"酸成柠檬精"）  2. 视觉描述  - 主角：线条风格的柴犬  - 构图：以柴犬的夸张表情动作为主  - 细节：加入符合情感的辅助元素（如困倦时的"zzz"）  - 风格：简笔画感，富有趣味性  - 重点：表情和肢体语言要夸张化处理  ## 输出要求  - 文案：`[4-6字网感文案]` - 描述：[具体的视觉描述，包含以下要素]  \* 柴犬的表情特征  \* 柴犬的姿态动作  \* 周围的环境元素  \* 整体画面氛围  ## 例子  用户输入：紧张 输出：  { "title": "缩成团团", "content": "一只柴犬蜷缩成圆球状，像个毛茸茸的饭团，眼睛瞪得圆圆的，耳朵紧贴着头，尾巴也缩在身体下面" }  ## 注意事项 1. 文案要具有当下流行语特色 2. 描述要具体且富有画面感 3. 情感表达要准确且夸张 4. 整体风格要保持可爱有趣 5. 要充分利用柴犬的特征进行创作 |

**步骤4-4：用户提示词**

|  |
| --- |
| SQL {{input}} |

**步骤5：添加“图像生成节点”**

**步骤5-3：正向提示词**

|  |
| --- |
| Bash 一只柴犬，表情包，{{content}}，图片上文字内容：{{title}}，黑白风格的插画作品，流畅线条，通过墨水的晕染效果展现出毛发的蓬松感。 |

**步骤6：设置“结束节点”**

**步骤6-3：回答内容**

|  |
| --- |
| SQL {{output}} |

**步骤9：智能体配置**

|  |
| --- |
| Bash 当用户发送内容时，调用 biaoqingbao 工作流，将用户发送的内容作为input字段传入 |

**第五，前端网页**

**步骤3：拼接好下面这段代码**

|  |
| --- |
| SQL curl --location --request POST 'https://api.coze.cn/v3/chat  --header 'Authorization: Bearer pat\_OYDacMzM3WyOWV3Dtj2bHRMymzxP\*\*\*\*' \  --header 'Content-Type: application/json' \  --data-raw '{   "bot\_id": "734829333445931\*\*\*\*",   "user\_id": "123456789",   "stream": **true**,   "auto\_save\_history":**true**,   "additional\_messages":[   {   "role":"user",   "content":"2024年10月1日是星期几",   "content\_type":"text"   }   ]  }'   写一个js脚本，前端请求脚本时，将发送的内容带入到参数content，调用上面的API，再写一个前端配套测试API调用的html页面，注意这个API是流式返回 |

**步骤5：优化js脚本**

|  |
| --- |
| SQL 复制上一步调用API后界面返回的代码内容，填写到这里  --- 优化js脚本，上面是js脚本中调用API的返回内容，希望在后端脚本中对返回内容进行结构化提取，当API完全返回后，提取event:conversation.message.delta中content的内容，在前端直接展示 |

**Part8：部署**

**01 将网页产品部署到云端！**

**步骤4：修改好下面这段代码，并提交给Windsurf生成**

|  |
| --- |
| Plain Text 写一个Cloudflare中的workers脚本，主要功能是请求一个API，注意API是流式返回，下面是API的请求样例，注意处理跨域问题。  curl --location --request POST 'https://api.coze.cn/v3/chat' \  --header 'Authorization: Bearer pat\_SCVB8n0KYoXaXLJ9j384iQS7JG\*\*\*\*\*' \  --header 'Content-Type: application/json' \  --data-raw '{   "bot\_id": "7450145225\*\*\*\*",   "user\_id": "123456789",   "stream": true,   "auto\_save\_history":true,   "additional\_messages":[   {   "role":"user",   "content":"震惊",   "content\_type":"text"   }   ]  }'   同时创建html页面测试API调用，worker使用get方式请求，入参为data，入参在点击发送按钮后会替换到请求体的content中进行API请求  将worker返回内容展示在页面上。一步一步来不要着急。 拼接好代码在Windsurf中发送，会创建worker.js文件和html文件 |

**Comments**