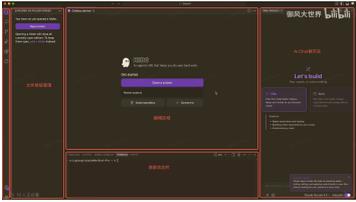
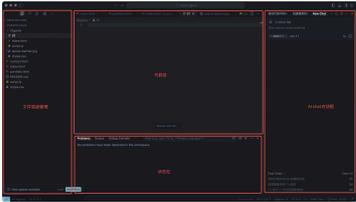


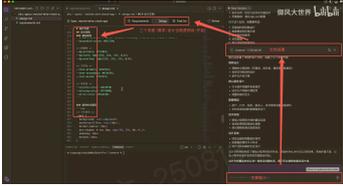
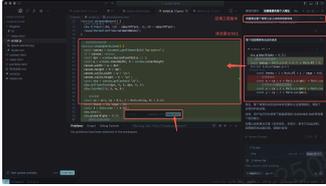
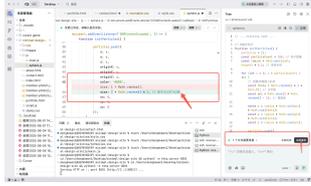
AI coding竞品分析

主要是针对kiro、cursor、trae、v0、coze进行竞品调研



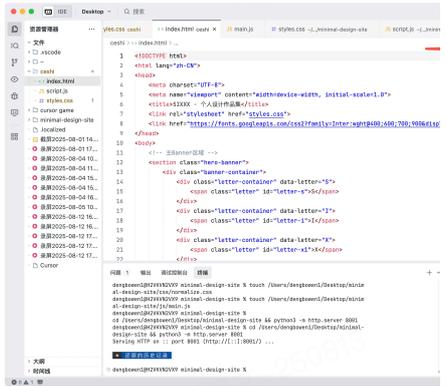
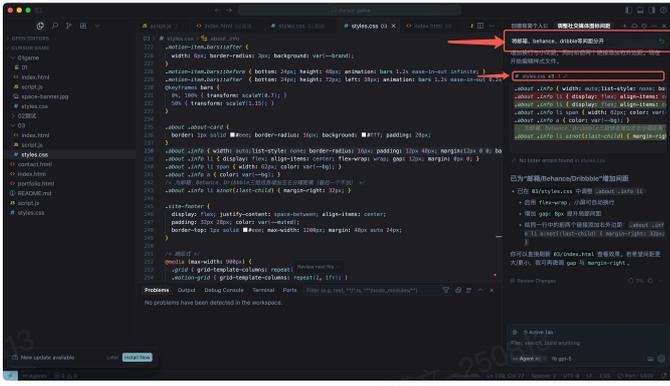
整体思路：产品整体定位——核心功能链路——核心功能点、用户体验（设计师、研发）、未来设计的协作模式挑战

产品名称	kiro	vs	Cursor	vs	Trae
产品定位	Kiro 是一款代理式 IDE，它通过规范驱动的开发方式，帮助您从原型开发走向生产。从简单到复杂的任务，Kiro 都能与您协同工作，将提示转化为详细的规范，进而转化为可运行的代码、文档和测试编辑助手，基于vscode		一款基于人工智能的代码编辑器，创建了一个集成开发环境（IDE）。旨在帮助开发人员编写代码，并与AI进行实时互动，提供代码建议、错误检测和自动补全功能。 基于vscode		Trae: IDE 与 AI 深度集成，提供智能问答、代码自动补全以及基于 Agent 的 AI 自动编程能力。使用 Trae 开发项目时，可以与 AI 灵活协作，提升开发效率。
主页页面布局					
功能链路	vibe模式-输入自然语言，AI助手直接开干（小任务，快速修改代码） spc模式-引入一个虚拟的开发团队（需求-设计-开发计划），分任务模块独立进行开发，（复杂项目，能够进行深度思考和规划，		代码导入，进行AI对话修改 想法构思→ Prompt文本→ AI代码生成→ 预览查看效果→代码编辑修改→ 发布上线		想法构思→ Prompt文本→ AI代码生成→ 预览查看效果→ 代码编辑修改→ 发布上线
展示形式	项目（网站/APP）/代码		项目（网站/APP）/代码		项目（网站/APP）/代码
模型	claude sonnet 4.0/3.7		claude sonnet 4.0/3.5 gpt-4.1 gemini pro-2.5		claude sonnet 4.0/3.7/3.5 gemini-2.5-Pro GPT-4.1/4o deepSeek-V3-0324/v3/r1
修改方式	文件引用# <ul style="list-style-type: none"> AI chat对话（基于上下文，可能会乱修改） 手动code代码修改 		文件引用@ <ul style="list-style-type: none"> AI chat对话（基于上下文，可能会乱修改） 手动code代码修改 		文件引用# <ul style="list-style-type: none"> AI chat对话（基于上下文，可能会乱修改） 手动code代码修改

			
核心功能	规范驱动开发 (spec- Driven Development)、自动化智能钩子 (Agent Hooks) (具体的规范)	AI自动补全代码、光标位置预测、函数重构 (语言文本-目标代码)	AI 自动补全、解释、生成代码、测试
核心用户群体	开发者/编程学习者/独立开发者	软件开发行业/开发者/爱好者	开发者 (独立)、全栈工程师
For 设计师	<ul style="list-style-type: none"> Vibe 模式下, 设计师只需用自然语言描述需求, 可完成页面生成 → 快速验证设计思路 在spc模式下, 设计师 = 半开发者: 能够参与到整个开发流程当中, 甚至可以自己完成原型到静态页面的全流程交付, 大幅提高个人产出效率 	<p>新手门槛低, 能够轻松构建网站页面/App</p> <ul style="list-style-type: none"> 0基础: 效果有点类似抽卡, 报错时, 修改成本高, 大部分时间在debug 一定基础: 高效的AI助手 (不限于复杂项目, AI会乱来), 能够帮助设计师实现一些比较好的效果 (创意审美) 	<p>整体上手程度和cursor几乎会差别, 新手交互流程会比cursor更顺畅 (引导流程、提示流程), 回溯本上cursor体验感更好</p> <ul style="list-style-type: none"> 生成的页面重复性会比较高 (同一个提示词和cursor对比, 整体框架类似, cursor丰富性更高) trae动效交互实现会更懂人话, 样式有点复古
挑战	0基础设计师搭配模板或结构引导使用, 对动画、微交互、响应式设计等高级体验仍需设计师手动补全或审美把控		
For 研发	<ul style="list-style-type: none"> 以结构化方式高效协作、稳定调用 AI 执行模块化开发任务 降低重复劳动, 提升页面一致性和开发效率 相较于传统 IDE, 它更适合规范驱动和组件级控制的前端/全栈开发。 	<ul style="list-style-type: none"> 能够缩短代码编写时间 不明所以 (对不熟悉的语言环境) /无成就感 (纯vibe coding) AI输出的内容越多, 带来的风险就越大 	/
网站链接	https://kiro.dev/	https://cursor.com/cn	https://www.trae.ai/
一致提示词	框架上: 同一个提示prompt, 动效修改意见保持一致 小细节: 颜色样式手动代码修改		
demo实现	/	<p>网站首页 (网页搭建)</p> <p>录屏2025-08-01 17.41.15.mov</p>	<p>网站</p> <p>录屏2025-08-05 10.52.20.nv</p>

AI coding 平台测试

平台	cursor	trae
控制变量	prompt提示词, 大模型选择 (gpt-5), 动态选择的修改也保持一致	
prompt	保持一致生成初始效果, 后续每次修改prompt提示词也保持一致	
交互问题	chat聊天框, 悬浮固定为Prompt文本内容, 能够直观感受到, 目前处在哪个修改问题下	chat聊天框, 悬浮固定为当前浏览的修改内容, 用户在向上浏览时-



本次测试实验主要集中在两款软件在**主banner**和**走马灯展示**这两块效果的呈现上, 其余利用AI理解能力进行生成, 后续手动

初始效果	录屏2025-08-12 16.15.05.mov	录屏2025-08-12 17.56.58.mov
视觉契合	<p>❤️❤️❤️❤️❤️</p> <p>整体视觉风格比较契合prompt文本预设, 色彩和谐, 没有约束的地方会以匹配的色系与主色系进行配合, 不滥用色彩, 且走马灯渐变色系不会扎眼睛</p>	<p>❤️❤️❤️❤️❤️</p> <p>整体视觉风格稍契合prompt文本, 给予颜色本含义, 未控制的区域未使用主色调, 颜色“脏”, 且走马灯太亮了</p>
交互动效	<p>❤️❤️❤️❤️❤️</p> <ul style="list-style-type: none"> 基础动效预设能够完全理解prompt文本, 字母的不同部分根据背景颜色动态变化未实现 <p>复杂交互动效会存在一定偏差, 需要手动进行修改调整 运行流畅, 不卡顿</p>	<p>❤️❤️❤️❤️❤️</p> <ul style="list-style-type: none"> 基础动效预设能够完全理解prompt文本, 字母的不同部分根据背景颜色动态变化未实现 <p>复杂动效基础样式能够呈现, 但未能完全实现 运行不流畅, 非常卡顿</p>
错误效果	/无	<p>! 走马灯动效实现错误!</p> <p>! 主banner动效层级代码错误!</p>

		<p>平面插画</p> 
效果呈现	<p>❤️❤️❤️❤️❤️</p> <p>布局上会存在gap和padding过大过小的状况，后续可手动代码修改 整体布局合理，颜色和谐，能够达到产品预期</p>	<p>❤️❤️❤️❤️❤️</p> <p>部分效果呈现出明显错误，同时还会出现 整体布局一致性很好，暂不能达到产品预期</p>
修改效果	<p>一次</p> <p>录屏2025-08-12 17.42.36.mov</p>	<p>修改第4次，待上传</p>
前端开发	<p>代码质量高，开发成本低，设计师：网站开发，上传至网站服务器</p>	

Prompt文本：

在XX文件夹中创建一个极简但富有视觉冲击力的个人设计网站，整体风格简洁大气，留白充足。

1. 主 Banner：

- 显示“SIXXX”五个大号粗体字母。居中在主banner
- 每个字母带有呼吸动效（平滑缩放与透明度变化）。
- 鼠标可以随意拖拽这些字母，带有平滑动画和惯性效果。当鼠标把字母拖出可视区域，则字母自动弹回原始位置
- 背景一半#2372FB，一半白色。在白色区域的“SIXXX”字母颜色为#2372FB，在#2372FB颜色区域的字母，颜色为白色，鼠标在拖动的时候逻辑按照这个来
- 主banner区域大小需要固定居中

整个页面同时还有作品展示模块，按照走马灯的形式进行展示，以及一些平面插画和动效展示模块，以及最后的个人信息展示模块，剩下的按照整体风格进行补充

实现效果不好，代码质量高也开发无意义