

学员图文讲义

AI 日程管理应用原型实操

从需求梳理到交互迭代

原课程讲师：陈文博

源视频时长：1 小时 36 分 37 秒

关键课程截图：14 张

整理日期：2026 年 9 月 9 日

依据腾讯会议录屏整理。适合课后复习与跟做。

目录

C O N T E N T S

课程导读

01 建立项目并说明真实场景

02 把业务规则问清楚

03 将讨论沉淀为需求文档

04 先比较单页风格再展开全部页面

05 生成完整原型并走通关键流程

06 通过页面批注调整布局

07 用拖拽快捷键和抽屉降低操作成本

08 从原型走向正式应用

09 课后练习与交付清单

视频时间轴索引

课程导读

这节课以个人日程管理应用「日序」为案例，演示如何用 AI 把一个模糊想法推进到可以操作、可以讨论的产品原型。学习重点是：怎样把需求问清楚，怎样选择界面风格，以及怎样通过实际操作继续修正需求。

课程日期为 2026 年 9 月 9 日，讲师为陈文博，录屏总长 1 小时 36 分 37 秒。本讲义为结构化整理版，已合并口头重复与等待过程；提示词均按课堂意思整理，便于复用。图下注明的是视频播放器时间，不是画面右下角的会议时钟。

学完后你应该能做什么

- 用具体场景向 AI 说明一个产品想法，并通过多轮问答收敛第一版范围。
- 把任务关系、日期安排和完成状态写成清楚的业务规则。
- 先比较一个页面的多种 UI 风格，再生成完整交互原型。
- 通过页面批注、拖拽试用和快捷键操作，提出可执行的修改要求。
- 将原型中的新决定回写需求文档，为后续正式开发准备输入。

本课做到哪一步

课堂完成了需求讨论、需求文档生成、单页风格探索和多轮网页原型修改。后半段介绍了后端即服务的思路，并展示另一个已有应用的云端后台。本案例尚处于原型阶段，不能把它当作已经接入账号、数据库和跨设备同步的正式产品。

课堂使用的主工具是 Codex；设计比较还涉及 Open Design、Stitch 和 Claude Design。Logo 作为支线使用了 ip-as-logo 技能。界面、模型选项及服务配置以录课当天画面为准，学习时重点掌握方法和判断过程。

全课工作路线

阶段	核心动作	应得到的结果
想法澄清	讲真实场景，让 AI 追问	用户流程和第一版边界
需求成文	梳理功能与例外情况	产品需求文档 PRD
风格探索	同一页面生成多个版本	选定一套 UI 风格
原型展开	补全页面及状态	可点击操作的网页原型

阶段	核心动作	应得到的结果
体验迭代	试用、批注、修改	更贴近真实使用的交互
开发准备	更新 PRD，讨论技术方案	后续实现的明确输入

01 建立项目并说明真实场景

回看范围：00:03:18—00:12:15。

将项目放在固定工作目录

课堂先建立专门的项目目录「日程管理 Demo」，并将其放在统一的 agent-workspace 下，再在 Codex 中选择该目录创建项目。这样需求文档、设计稿和原型文件集中存放，后续查找、交接和版本管理更方便。

复做时，先为自己的项目建立一个独立文件夹，再让 AI 在该文件夹中工作。不要只记住聊天窗口名称而忽略产出文件的位置。

第一轮先讲背景和使用过程

最初的想法来自日常记录任务的习惯：想到事情时先记下来，之后安排到某一天，到当天再决定如何执行。原有流程涉及 Obsidian，但课堂为了快速做出可体验的版本，先将范围收敛为网页应用。

下面的提示词可作为开场。它保留了课堂的沟通方式，并把“先讨论”的阶段要求写清楚。

需求探索提示词：你现在是一个产品专家，请引导我一起梳理产品创意。我想做一个个人日程管理网页应用：平时随手记录任务，先安排到某一天，到当天再决定具体执行时间。我希望收集任务足够方便。请先和我讨论使用场景与功能边界，暂时不要生成代码。

我们应该在日程管理Demo中做些什么？

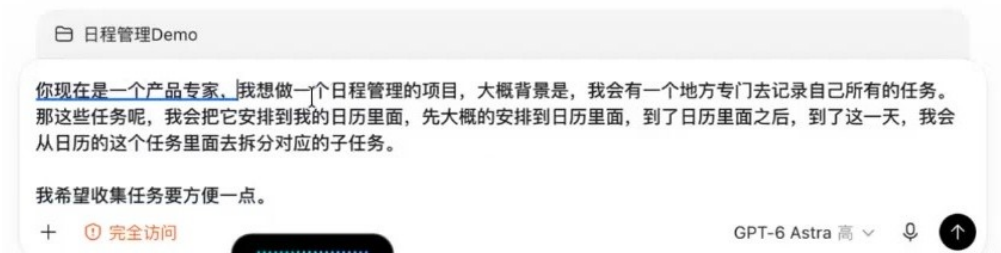


图 01 | 以产品专家角色说明背景并开始需求讨论 · 00:07:45

观察图中的输入方式：先说明背景和期望，再让 AI 帮助梳理。课堂后续多轮追问中，讲师不断修正最初“拆分任务”的含义，而不是默认 AI 第一轮理解就正确。

把收集动作减到最少

课堂明确了三个边界：手机和电脑都可能使用；第一版只处理文字；语音转文字交给现有输入法。应用只需保存最终文字，不必额外开发语音识别、链接解析和截图识别。

收集时只回答“我要记什么”。日期、时间、分类等信息不应成为记录一件事的前置条件。这个取舍直接服务于“随手记下来”的目标。

边界澄清提示词：第一版先只收集文字，输入一句话就进入 Inbox。语音转文字由我自己的输入法负责，不在这个应用里开发。收集时不强制填写日期或具体时间。

02 把业务规则问清楚

回看范围：00:12:26—00:23:31。这一段是本课需求分析的重点。

先定日期 再定时间

“安排到周三”与“安排在周三 14:00”是两个不同层次的决定。用户在收集任务时，通常还不知道执行细节；因此允许任务已经有日期，但没有具体时间。到执行当天，再补充时间安排。

这一点会影响页面分组：同一天应能同时出现未定时间、已定时间和已完成的任务。不能因为时间没填，就把任务当作还没安排。

关联任务保持独立

课堂最初提到“拆任务”，AI 一度将其理解为父子任务结构。讨论后澄清：执行过程中想到的新工作，先创建为独立任务，进入 Inbox，再和当前任务建立关联。

例如正在准备培训方案时，想到还要与客户确认参训人数，就另建一条“和客户确认参训人数”。它与“准备培训方案”有关系，但可以单独安排日期和完成。

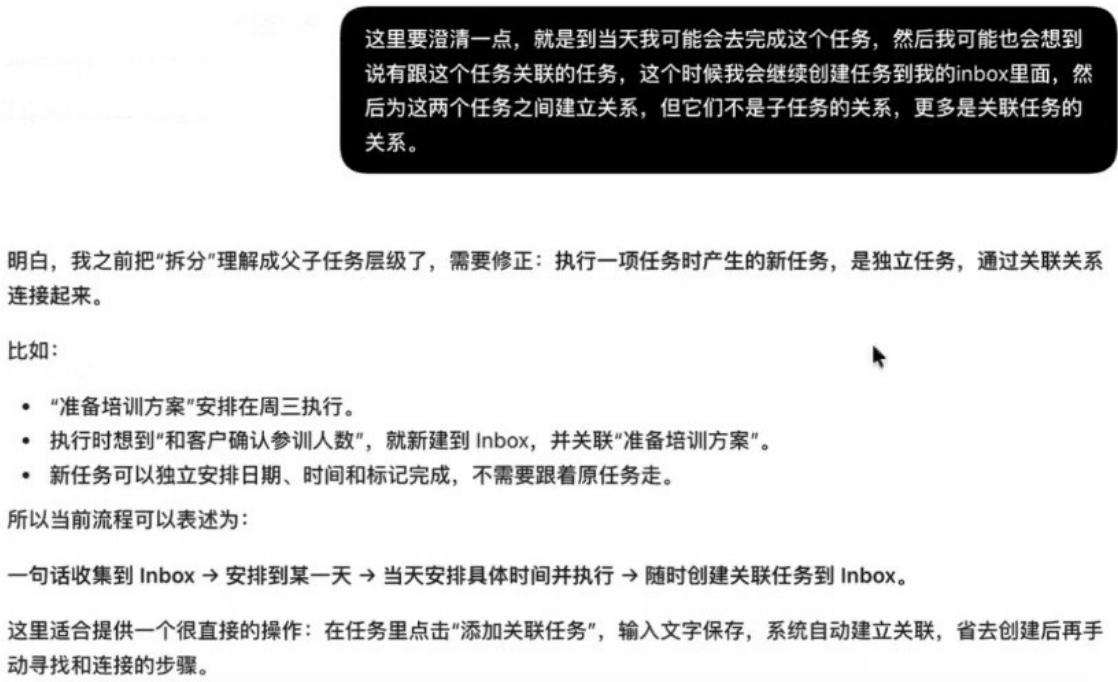


图 02 | 明确新任务进入 Inbox 并与原任务建立独立关联 · 00:17:20

这里应检查的不是有没有“关联”按钮，而是新任务是否保有独立的安排与状态。若界面把它强制放进子任务树，就已经偏离课堂确认的产品模型。

区分完全没做与部分完成

日终遇到未完成任务时，不能统一标记完成，也不能把所有任务机械地挪到明天。课堂区分了两类情况。

实际情况	处理方式	是否生成新任务
完全没做 暂不确定何时做	原任务退回 Inbox	否
完全没做 已确定其他日期	原任务改期	否

实际情况	处理方式	是否生成新任务
只完成一部分	留下已完成内容，把剩余内容建为关联任务	是

“完成培训方案” 只做完需求梳理时，今天可以留下“整理培训需求”并标记完成；另外新建“编写培训方案”，关联今天的记录，放进 Inbox 或安排到后续日期。

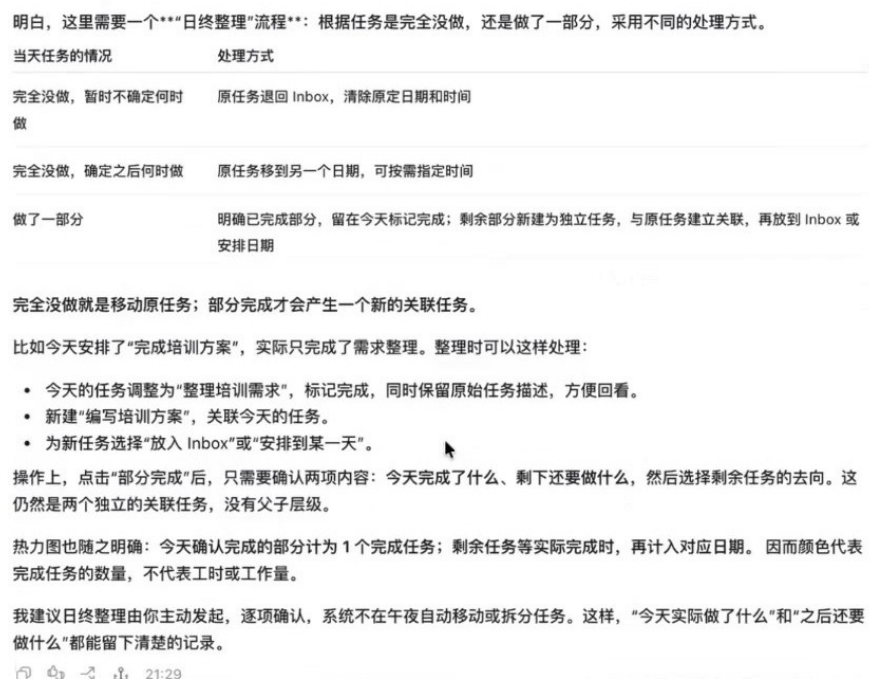


图 03 | 完全没做移动原任务 部分完成生成关联任务 · 00:23:30

部分完成的交互至少要让用户确认两项内容：今天完成了什么、剩下还要做什么。任务的范围发生变化，名称也应能准确表达新范围。原型方案还保留原始计划，方便回顾。

热力图表达完成数量

课堂希望参考代码提交热力图：某天实际完成的任务越多，颜色越深。安排了很多任务但没有完成，不应因此变深。原型采用实际完成日期统计，例如周三计划、周四完成的任务记到周四。

热力图表达的是完成任务数量，不是工作时长或工作量。部分完成整理后，今天确认完成的内容计一次；新产生的剩余任务在以后真正完成时再计入。

03 将讨论沉淀为需求文档

回看范围：00:23:31—00:28:14。

需求文档 PRD 的作用，是把分散在聊天里的决定集中保存，作为原型设计的输入。课堂使用 Markdown 文件，不强加复杂模板；如果团队已有统一规范，可以通过对应技能生成标准格式。

需求文档提示词：基于我们聊过的需求，帮我整理一份产品需求文档。接下来我会基于这个文档生成产品原型。请区分已经确认的需求与尚待确认的交互建议。



图 04 | 将讨论整理为 PRD 并区分确认需求与原型建议 · 00:27:30

生成后应该检查什么

课堂强调，正常情况下应认真阅读生成的文档，确认 AI 是否理解正确。演示中为推进流程略过了逐条审阅；同学复做时应补上这一步。

重点核对：文字收集是否足够简单；是否允许只定日期；关联任务有没有被改成子任务；完全没做与部分完成是否分别处理；完成统计是否按实际完成日计算。

再检查“第一版不做什么”。本课暂不接 Obsidian 同步，不开发应用内语音输入，不建立复杂子任务树。未讨论过的功能不要因为 AI 顺手写入，就变成已确认需求。

让文档能指导画原型

一份可执行的 PRD 不应只有功能名称，还应说明入口、操作、结果和关键状态。例如“部分完成”不能只写一个按钮，而要描述填写两段内容、选择剩余去向、确认后产生两条关联记录的过程。

可以用一条具体任务走查全文：从“准备培训方案”进入 Inbox 开始，安排日期、设置时间、产生关联工作，再到完成或日终整理。如果某个环节解释不清，先补文档再继续。

04 先比较单页风格再展开全部页面

回看范围：00:27:45—00:42:06，补充比较见 00:46:53—01:00:24。

用小样确定设计方向

课堂没有直接要求 AI 一次做完所有页面，而是先选一个核心页面，生成五种不同风格。这样能在较小范围内比较排版、字体、颜色和层次，确认后再扩展。

风格探索提示词：基于产品需求文档，挑一个核心页面，帮我生成 5 种不同风格的 UI 设计 Demo。先只做一个页面，我确定风格之后再生成完整原型。不同方案应能直接预览和比较。

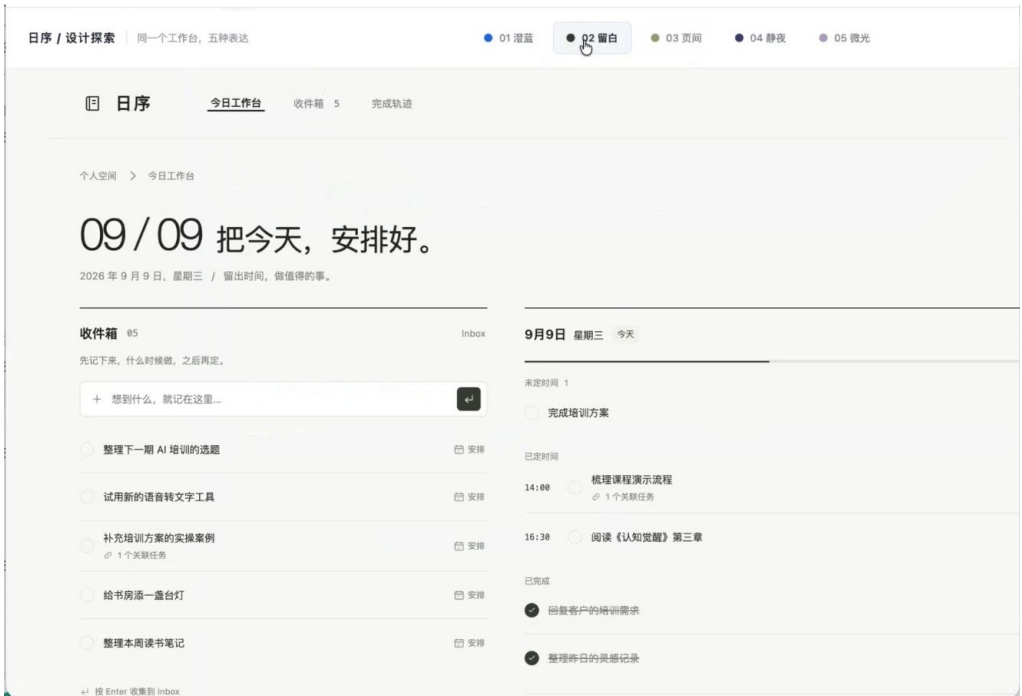


图 05 | 同一工作台生成五种方向并选中留白方案 · 00:39:25

Codex 的五个方向分别为澄蓝、留白、页间、静夜和微光。课堂讨论中对留白和静夜有好感，随后选择「02 留白」继续生成完整原型。

这一轮看的是整体视觉方向，不要把演示文字或临时数据误当作最终内容。比较时可以先问：字是否舒服、层次是否清楚、操作入口是否容易找到、是否适合长时间使用。

同一份需求可以交给不同设计工具

课堂还尝试了 Open Design、Stitch 和 Claude Design，用多份结果辅助判断。这个做法的价值在于增加具体参照，不是预先认定某个工具一定最好。



图 06 / Stitch 中选择设计系统作为风格输入 · 00:30:55

课堂在 Stitch 中展示了选择设计系统的入口，并讨论沿用熟悉产品的风格或使用自己的设计预设。已有设计规范的团队可直接提供规范；新产品则可以先做风格探索。

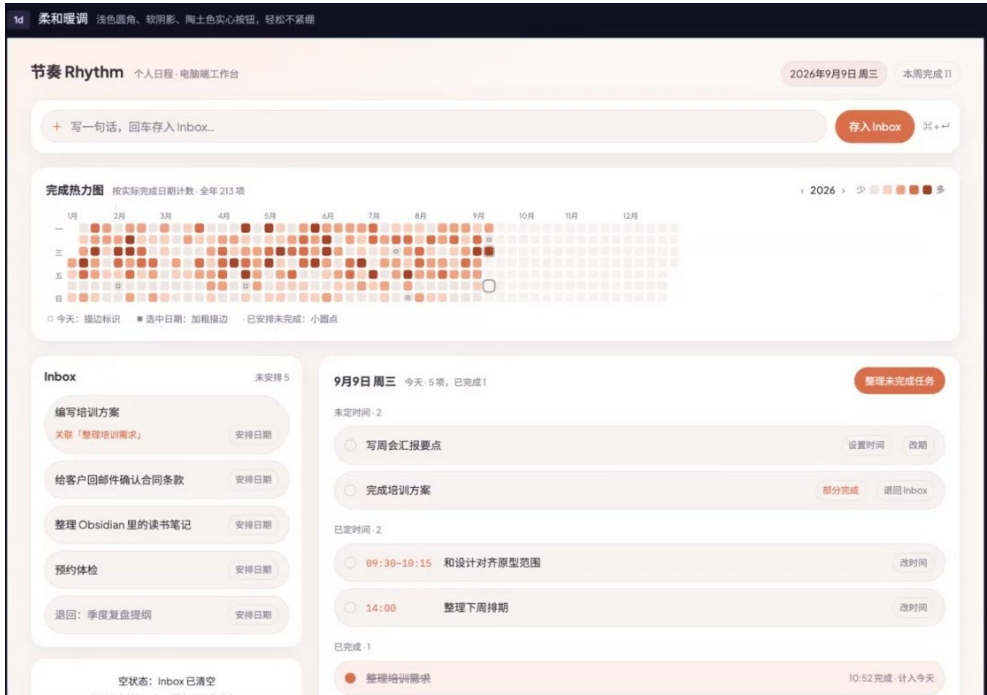


图 07 | Claude Design 展示柔和暖调的另一套方案 · 01:00:10

Claude Design 中又出现了不同方向，课堂选了柔和暖调继续比较。这是另一组设计探索，主线原型仍在 Codex 的留白方案中迭代。复做时不必把所有工具都跑一遍，掌握“提供同一需求、观察结果、作出选择”的方法即可。

Logo 是可独立推进的支线

等待原型时，课堂用 ip-as-logo 技能，根据产品需求生成 Logo 候选，再从多张图中挑选。它可以帮助探索产品形象，但不能替代核心流程的确认，也不意味着候选图已经接入正式产品。

05 生成完整原型并走通关键流程

回看范围：00:41:36—00:58:11。

完整原型提示词：基于第 2 种留白风格，创建所有页面的产品原型，放到 prototype 目录中。以已经确认的产品需求文档为依据，覆盖主要页面和关键状态。

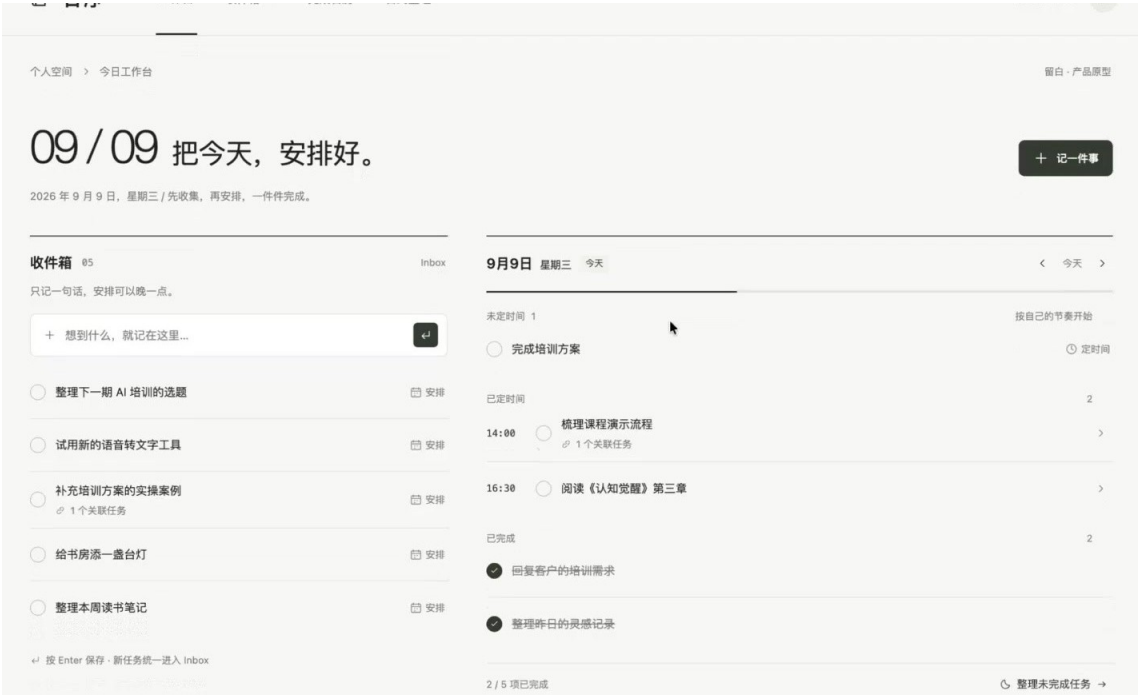


图 08 | 第一版工作台并列展示收件箱与当天任务 • 00:53:20

第一版工作台把收件箱和当天任务并列展示，当天任务分为未定时间、已定时间和已完成。页面下方还有完成轨迹。这个布局让用户能开始操作，也暴露出后续要修改的问题。

按真实任务走一遍

先新增一条“准备培训方案”，检查它是否进入 Inbox。再只指定日期，确认不填时间也能安排成功。进入该日期后补充时间，观察任务分组是否随之变化。

随后在任务中添加关联工作，检查新任务是否进入 Inbox，以及能否从关联入口找到它。最后分别尝试标记完成、退回收件箱、改期和部分完成。

课堂实际尝试了部分完成后的结果和关联入口，也发现任务指定到其他日期后不容易找回。这是原型评审的重要反馈：按钮能点，并不代表整个体验已经顺畅。



图 09 / 原型总览呈现工作台 收件箱 日历和日期任务 · 01:00:30

原型总览可以帮助检查页面是否齐全，但最终仍要进入页面实际操作。需要覆盖的不只有正常列表，还包括空状态、取消操作、整理结果和关联任务的查看。

怎样判断第一版是否够用

- 一句话新增任务后，能在 Inbox 找到它。
- 指定某个日期后，能到该日期找到同一任务。
- 未定时间与已定时间的分组符合实际状态。
- 关联任务独立存在，日期和完成状态不会被误联动。
- 部分完成后，已完成内容与剩余工作都有清楚的去向。
- 取消整理不会留下多余任务。

06 通过页面批注调整布局

回看范围：01:01:24—01:12:42。

把修改意见定位到具体元素

课堂使用内置浏览器的批注功能，选中页面元素后添加评论，可以记录多条再统一发送。比起只说“页面再好看一点”，指出位置、问题和目标更容易得到符合预期的修改。

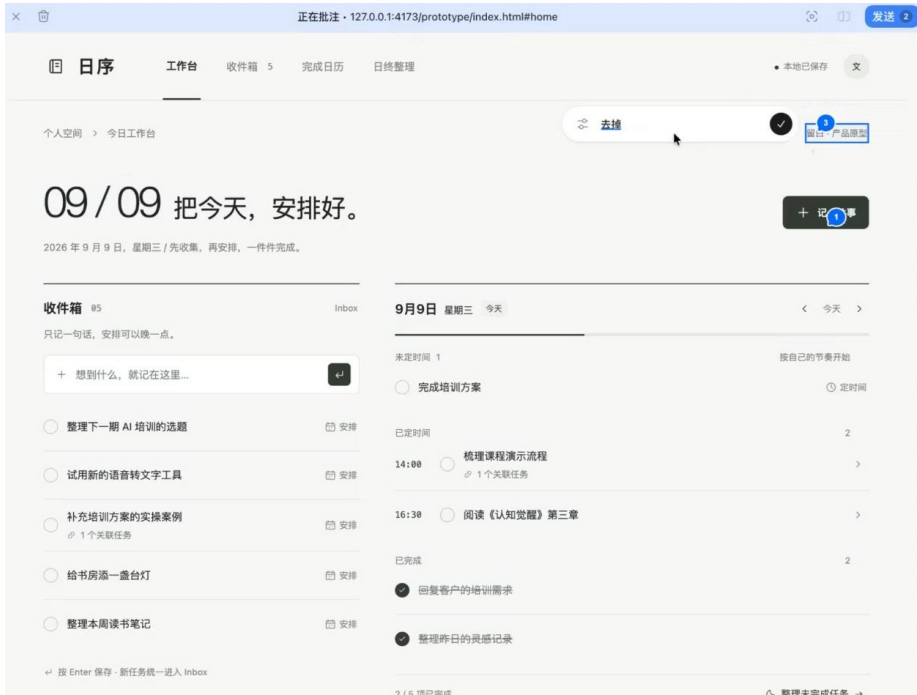


图 10 | 通过批注定位冗余标识并集中发送修改意见 · 01:02:55

图中的多条批注属于同一轮修改。课堂先发现顶部“记一件事”与收件箱输入框重复，再提出把完成轨迹移到上方，并去掉不必要的原型标识。

批注整理示例：顶部的“记一件事”和收件箱输入重复，请删除顶部入口。把完成轨迹移到任务列表上方，并支持周、月、季度和年视图。删除页面中不影响使用的原型标识。

从全年热力图走向可安排的日历

全年热力图适合回顾，但日期格子较小，不方便精确安排任务。课堂提出周、月、季度等视图，希望在查看某一天时，下方能找到对应任务。

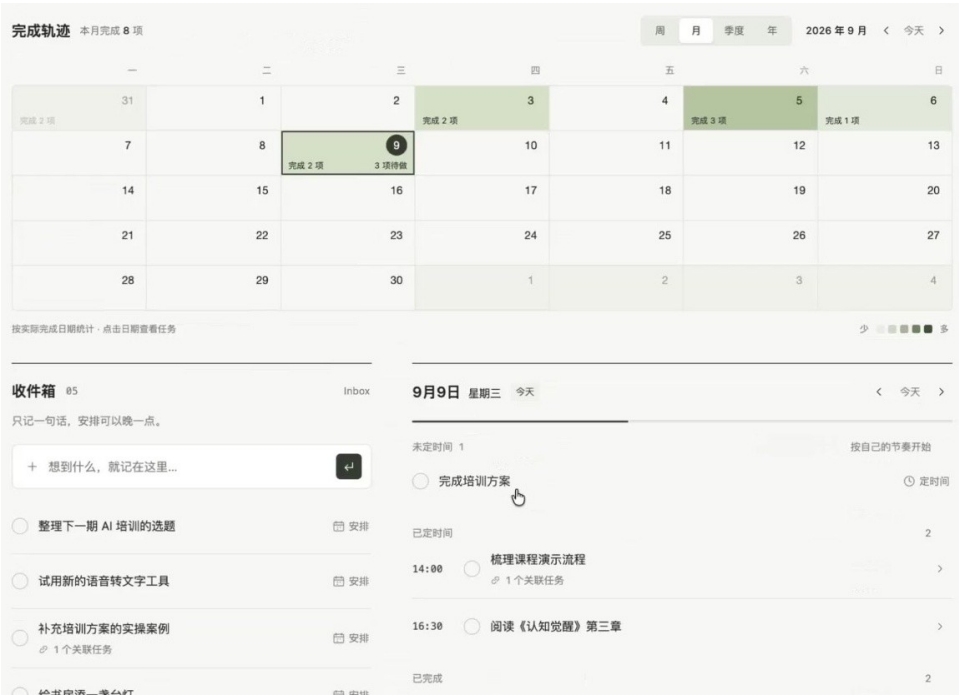


图 11 | 周期日历上移并与下方任务区结合 · 01:07:30

把日历移到工作台之后，原来的独立“完成日历”菜单出现重复。课堂决定删除这个菜单及对应代码，让工作台承担日历查看功能；独立收件箱因为可能需要承载较多任务，暂时保留。

这里的判断依据是使用场景。不是所有重复出现的区域都要删除，而是先确认是否还有独立用途。收件箱可能有二三十条任务，单独页面仍可能方便集中浏览。

看见原型之后继续修正需求

最初以为需要分成多个页面，看到实际界面后又认为可以合并；最初只有完成轨迹，试用后又需要日期导航。这些变化说明原型也是澄清需求的工具。

每轮修改都应能回答：解决了什么不顺手的问题；用户现在少做了哪一步；原本需要的信息是否仍然找得到。

07 用拖拽快捷键和抽屉降低操作成本

回看范围：01:07:32—01:19:53、01:27:53—01:32:45。

拖拽安排任务

第一版需要点击“安排”、选择日期再确认。课堂改为更直接的方式：把收件箱任务拖到右侧任务区，或拖到上方某个日期格。

拖拽提示词：收件箱任务应支持拖拽到右侧任务区，安排到当前选中的日期；也支持拖到上方周、月、季度日历的日期格，安排到对应日期。请让目标日期清楚可见。

年度格子过小，课堂不倾向于将年视图作为拖拽排期入口。完成拖拽安排后，又补充反向操作：将某天未完成任务拖回 Inbox，解除原来的日期与时间安排。

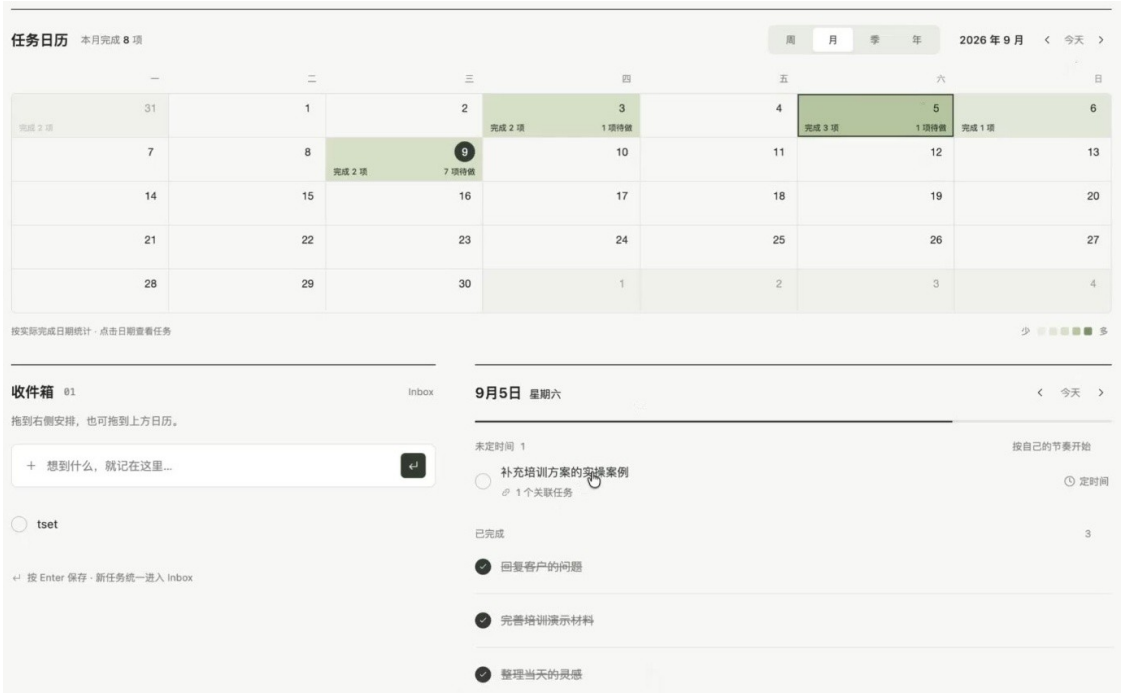


图 12 | 选中 9 月 5 日后右侧呈现该日任务 · 01:28:40

拖拽发生的是同一任务的移动，不应复制出第二条任务。关联关系也应该保留。课堂随后要求去掉逐行“安排”按钮，减少重复入口；是否适合其他设备与场景，仍应在自己的原型中检查。

任意页面快速收集

课堂提出在应用各页面使用 Command + K 唤起新增任务浮窗，让用户想到一件事时不用先跳回收件箱。

快捷键提示词：我希望在应用所有页面都能按 Command + K 唤起新增任务浮窗，保存后进入收件箱。打开浮窗时不要打断原页面正在编辑的内容。

注意“全局”在这里指网页应用内各页面，不是操作系统任意应用。课堂也遇到宿主 Codex 的快捷键先被触发的情况，因此演示时需要关注焦点究竟在宿主还是网页。

任务详情改为侧边抽屉

独立详情页会让用户离开当前工作台。课堂希望从右侧滑出任务详情，保留编辑、完成和关联任务操作，关闭后仍回到原来的页面位置。

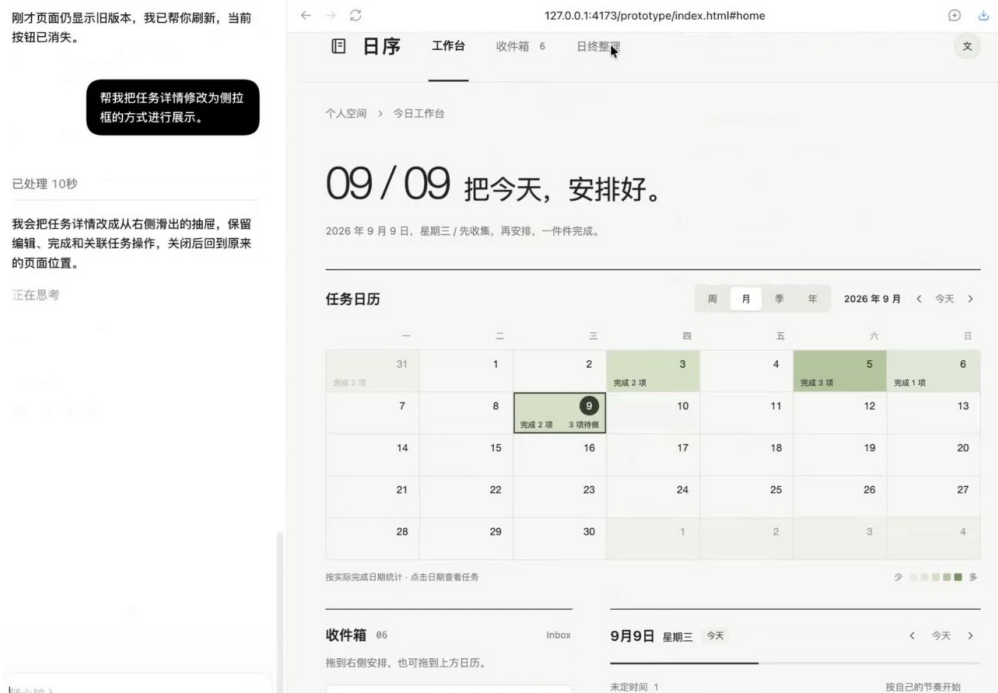


图 13 | 提出将任务详情改为右侧抽屉 此帧仍在修改中 · 01:31:25

详情修改提示词：把任务详情改为右侧滑出的抽屉，保留内容编辑、完成和关联任务操作。关闭后留在原页面，方便我继续安排其他任务。

这项修改体现的原则是保持上下文：用户查看细节时，仍知道自己正在处理哪个日期、哪组任务。课堂还讨论了“任务拖到任务上自动建立关联”，但它属于当时提出的想法，不当当作已经完成的交互。

按阶段安排验证强度

课堂明确区分原型探索与正式代码修改：原型的小改动可由使用者立即查看，不需要每次都启动完整 Playwright 检查；生产开发则仍需要相应验证。减少的是反复等待，不是取消对真实结果的检查。

遇到修改后页面看起来没变化时，先确认是否打开了正确的原型页面并刷新，再判断修改是否生效。课堂尾段就出现过旧页面仍显示“安排”按钮、刷新后才消失的情况。

08 从原型走向正式应用

回看范围：00:32:45—00:33:31、01:20:35—01:27:52、01:31:50—01:35:43。

用最终原型更新 PRD

课堂提出，经过多轮修改的原型已经承载了更明确的需求，应反过来更新 PRD。否则后续开发仍按旧文档执行，容易把已经删除的菜单或已经改变的交互重新做回来。

回写需求提示词：请根据当前确认的原型更新产品需求文档，说明页面结构、任务流转和交互规则的变化。区分已经落到原型中的内容、仍未确定的问题以及后续开发计划。

后续开发的推进顺序

课堂介绍的后续路线是：确定技术方案，搭建最小可运行骨架，确认前后端与数据库连通，然后按用户故事逐个实现功能。

这里的“骨架”是一个最小验证链路。例如页面能够调用后端，读取数据库中的一条记录并显示出来。讲师提到 React、Spring Boot、MySQL 作为既往教学中的示例，并未把这套组合确定为本项目必须使用的技术栈。

理解后端即服务

为了说明个人轻量应用怎样保存数据和处理登录，课堂展示了另一个已有 ToDo 应用使用的 CloudBase 后台。讲师用三个部分解释它的作用：数据库保存数据；云函数承载逻辑并供前端调用；身份认证处理登录相关能力。



图 14 | 已有应用后台展示身份认证的登录方式配置 • 01:26:55

这张图用于理解后端能力的分工，不是本项目已经完成接入的证明，也不是让同学照抄全部开关。课堂示例使用用户名密码登录；具体项目应在技术方案阶段决定自己的认证与数据访问规则。

后续课程计划还提到 CloudBase 或 Supabase 等服务。套餐价格、额度和当前功能不在本讲义中作结论，本课重点是理解可以将一部分基础设施交给托管服务。

版本管理与未来集成

课堂末尾提出用 Git 管理原型，并通过 GitHub 交接给同学。版本管理帮助记录每轮修改，便于继续协作；提出交接要求与完成上线是不同阶段。

另一个未来方向是提供 CLI 或 MCP 接口，再通过定时任务将 Obsidian 日记中的任务写入收件箱。这一部分属于拓展设想，课堂没有完成同步开发。先把网页里的核心流程打磨顺畅，再考虑集成。

09 课后练习与交付清单

课堂明确留下的作业是：继续调整原型，直到它更符合自己的使用习惯。下面把这项作业整理为可提交、可复查的步骤；提交材料清单为学习补充建议。

练习一 梳理你自己的真实流程

写出一条最近真的需要完成的任务，描述你什么时候想到它、现在记在哪里、什么时候决定执行日期，以及未完成时怎样处理。用第 01 章的提示词与 AI 讨论，再写清第一版不做什么。

完成标准：你能用具体例子解释“收集—安排—执行—整理”的全过程，而不只是列出一串功能名。

练习二 做一次有理由的风格选择

使用同一份需求和同一个页面生成多个设计方向，选出一个，并写下选择理由。至少检查字体可读性、信息层次和主要操作位置。不要只写“比较好看”。

完成标准：产出一个可查看的风格方案，并能解释它为什么适合你的工作场景。

练习三 连续使用并修改原型

在原型中新增几条自己的示例任务，把其中一条安排到其他日期，再找到它；把另一条按部分完成处理；试用收件箱、日历和详情之间的切换。记录至少三处不顺手的地方，用定位明确的批注进行修改。

完成标准：每条修改意见都包含“在哪个位置”“遇到了什么问题”“希望怎样变化”，修改后再检查实际行为。

交付前检查

- 能仅输入一句话新增任务，不强制填写日期或时间。
- 只安排日期、不设置时间时，任务仍可在目标日期找到。
- 改期、拖拽和退回 Inbox 不生成重复任务。
- 关联任务保持独立，并能找到关联对象。
- 部分完成保留已完成内容，剩余内容形成关联任务；取消不改变原状态。
- 热力图深浅对应实际完成数量，不把仅计划的任务算作完成。
- 切换日期和关闭详情后，页面位置及任务上下文清楚。
- 需求文档与最终原型一致，后端、同步等未完成项单独列出。

建议提交的材料

材料	内容
产品需求文档	已确认流程、规则、边界和未决问题
可操作原型	同学或老师能打开并试用的版本
修改记录	至少三条问题及对应调整结果
后续计划	下一步要实现的功能与技术问题

本课最值得复用的习惯是：先让需求具体起来，再用原型检验理解，最后把新的决定写回文档。这样下一轮开发面对的就是更清楚的产品问题。

视频时间轴索引

下列时间用于定位相应讲解的起点；截图时间单独标在图注中。可先阅读相关章节，再跳回视频查看操作过程。

播放器时间	回看内容
00:00:00	已有 ToDo 应用示例和新项目起因
00:03:18	收敛为网页应用并建立项目目录

播放器时间	回看内容
00:05:31	先与 AI 讨论产品想法
00:09:15	收集入口 手机电脑与文字输入边界
00:12:26	先安排日期 到当天再安排时间
00:14:34	讨论任务层级并澄清独立关联任务
00:17:22	从起点到终点检查完整体验流程
00:18:16	按完成数量设计热力图
00:19:26	完全没做与部分完成的日终处理
00:23:31	生成产品需求文档
00:24:39	比较可用于原型设计的工具
00:27:45	单页五种 UI 风格探索
00:30:41	Stitch 设计系统入口
00:32:45	后续技术方案和最小骨架的思路
00:38:32	比较 Codex 风格并选择留白
00:41:36	生成全部页面原型
00:45:38	通过技能生成 Logo 候选
00:49:31	Claude Design 风格探索
00:52:43	试用工作台与部分完成流程
01:01:24	页面批注及布局修改
01:07:32	拖拽排期与删除重复日历入口
01:13:19	原型阶段的验证方式
01:17:07	Command + K 快速新增任务
01:20:35	后端即服务的思路
01:26:00	CloudBase 数据库 云函数和身份认证
01:27:53	快捷键和拖拽演示及反向拖回需求

播放器时间	回看内容
01:30:44	任务详情改为侧边抽屉
01:31:50	根据原型反向更新 PRD
01:32:54	Git 管理和原型交接要求
01:33:25	课后作业与下一阶段安排
01:34:32	CLI MCP 与 Obsidian 同步设想

课程素材：2026 年 9 月 9 日腾讯会议录屏 meeting_01.mp4。术语与项目结构辅以同项目的产品需求文档、设计说明和原型说明核对；学习练习及检查清单为本讲义的整理补充。