【项目名称】Nano Banana Pro – AI 图像生成平台

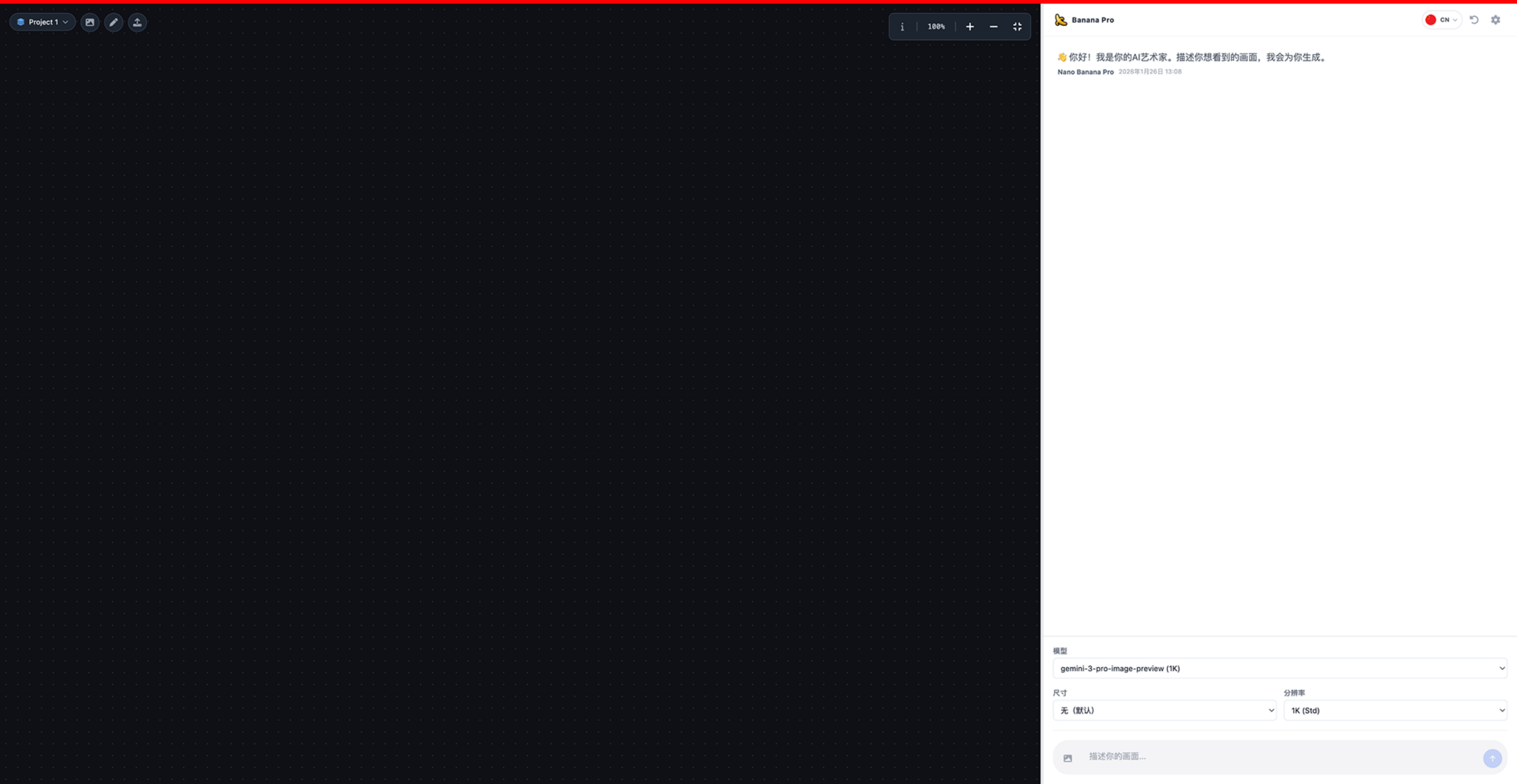
【项目简介】
基于 FastAPI + GPT-5 的智能图像生成 Web 应用，通过自然语言描述生成高质量 AI 图像。
集成 GPT-5 智能对话、多模型图像生成、无限画布系统，提供完整的创意到成品工作流。

【技术栈】
Python 3.9 | FastAPI | GPT-5 API | Gemini API | JavaScript ES6+ | TailwindCSS | Canvas API

【核心功能】
- 智能提示词优化：GPT-5 自动生成 3 个优化方案，点击即可生成图像
- 多模型集成：支持 Gemini（1K/2K/4K）、GPT-5、DALL-E 3，统一 API 接口
- 无限画布系统：支持缩放拖拽、绘图工具、自定义背景
- 图生图功能：上传参考图，AI 理解风格并生成相似图像
- 上下文理解：GPT-5 记忆对话历史，支持连续优化指令

【技术亮点】
- 异步 API 调用：使用 httpx 异步客户端，智能超时控制（img2img 180s，普通 60s）
- 智能端口检测：自动扫描 8001-8010 端口，避免端口冲突
- GPT-5 上下文处理：智能分析对话历史，提取关键信息优化提示词
- 完善错误处理：全局异常处理器，确保所有错误返回 JSON 格式
- 前端状态管理：LocalStorage 实现项目状态持久化

AI 图像生成平台-首页i截图



项目概述

智能 GPT 对话应用是一款基于 OpenAI API 的高性能对话系统，通过创新的分层摘要算法和智能上下文管理，解决了长对话场景下的成本爆炸和上下文限制问题。项目采用 React + TypeScript 技术栈，实现了完整的会话管理、记忆系统、历史检索和图片识别等功能。

核心技术创新

项目的核心创新在于设计并实现了业界首创的 5 层金字塔摘要系统。该系统通过智能分组算法自动识别对话单元（一问一答、话题转换等），将历史对话分层压缩为 Level 1–5 五个层级，配合保留最近 5 条原始消息的策略，成功将上下文大小固定在约 5,615 tokens，实现了 96%+ 的成本优化。在 500 条消息的场景下，成本从 ¥9,076 降至 ¥356，同时支持 1000+ 轮无限长对话。

此外，项目实现了三层记忆管理系统，包括用户手动添加的永久记忆、AI 自动生成的对话摘要和最近的原始消息，三层记忆智能融合到上下文中。结合 RAG（检索增强生成）技术，系统能够自动检测历史查询意图，支持模糊搜索、时间理解和相关性排序，实现了智能的历史对话检索。

技术架构与实现

前端采用 React 18 + TypeScript + Vite 构建，使用 Zustand 进行状态管理，Tailwind CSS + Framer Motion 实现精美的 UI 交互。后端使用 Node.js + Express 提供本地文件存储服务，数据采用 IndexedDB + 本地文件双重存储机制，确保数据安全可控。

在算法层面，实现了智能分组算法用于识别对话边界，自研 Token 估算算法用于精确计算上下文大小，以及动态预算分配算法用于优化历史检索结果。项目还设计了模型智能切换机制，根据任务类型自动选择最优模型：日常对话使用 gpt-5.2 保证质量，图片识别使用 gpt-4o 平衡效果和成本，摘要生成使用 gpt-4o-mini 降低成本。

功能特性

系统支持无限会话管理，包括会话置顶、私密会话（密码保护）、拖拽排序等功能。实现了完整的消息操作能力，支持编辑、删除、重新生成和导出（JSON/Markdown 格式）。图片功能支持拖拽、粘贴、点击上传，自动进行 OCR 识别并提取文字内容。所有 AI 响应采用流式输出，支持实时显示和随时中断。

记忆系统能够自动识别用户的"记住"指令，通过特殊标记 `[MEMORY_ADD:内容]` 自动提取并保存重要信息，支持通用记忆和会话专属记忆的分组管理。历史搜索功能支持自然语言时间表达（如"上周"、"最近"），自动触发检索，并按相关性和时间权重排序结果。

项目成果

项目实现了显著的成本优化效果：100 条消息节省 96.7%，500 条消息节省 96.1%，1000 条消息节省 98.0%。上下文大小固定在约 5,615 tokens，不随对话增长而增长，从根本上解决了长对话的成本问题。摘要生成采用异步处理，1–2 秒完成，不阻塞用户操作。

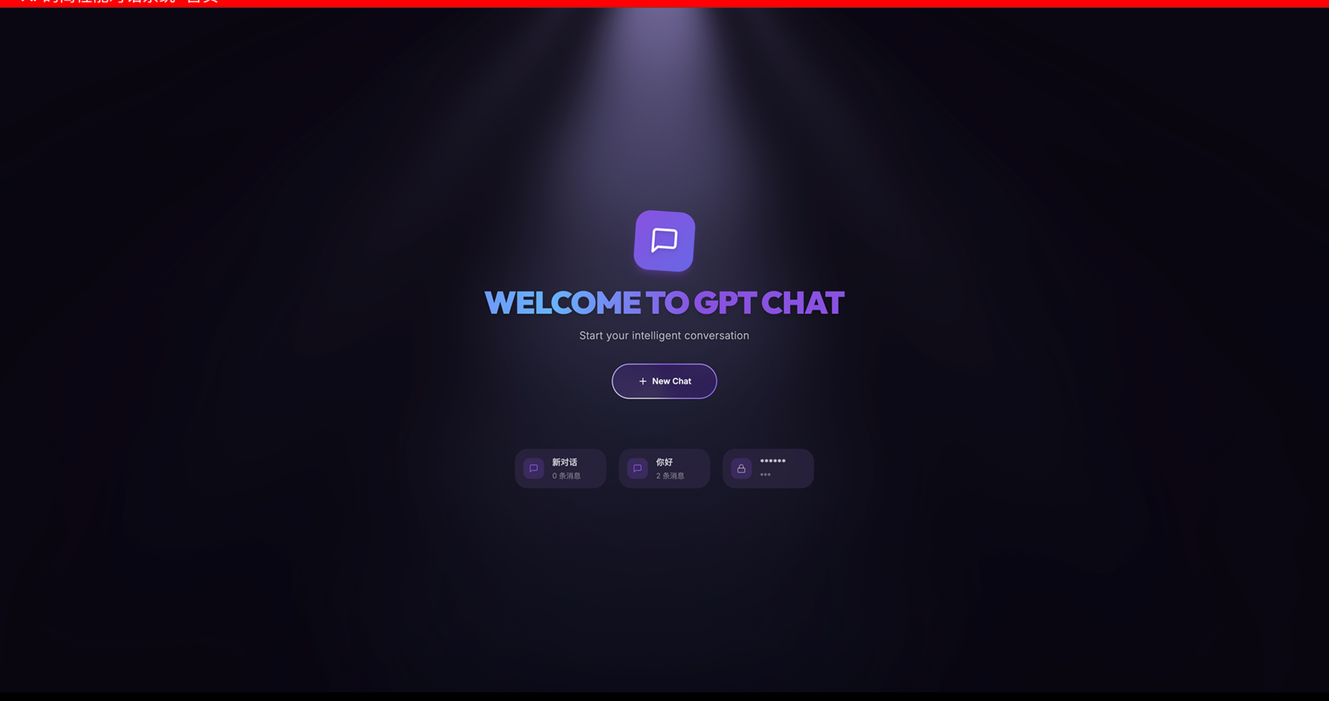
代码层面，核心功能模块超过 2,000 行，测试覆盖率达到 80%+，技术文档超过 7,500 行。项目采用模块化设计，功能边界清晰，易于维护和扩展。实际使用中，系统稳定运行，无崩溃和数据丢失，用户体验流畅。

技术价值

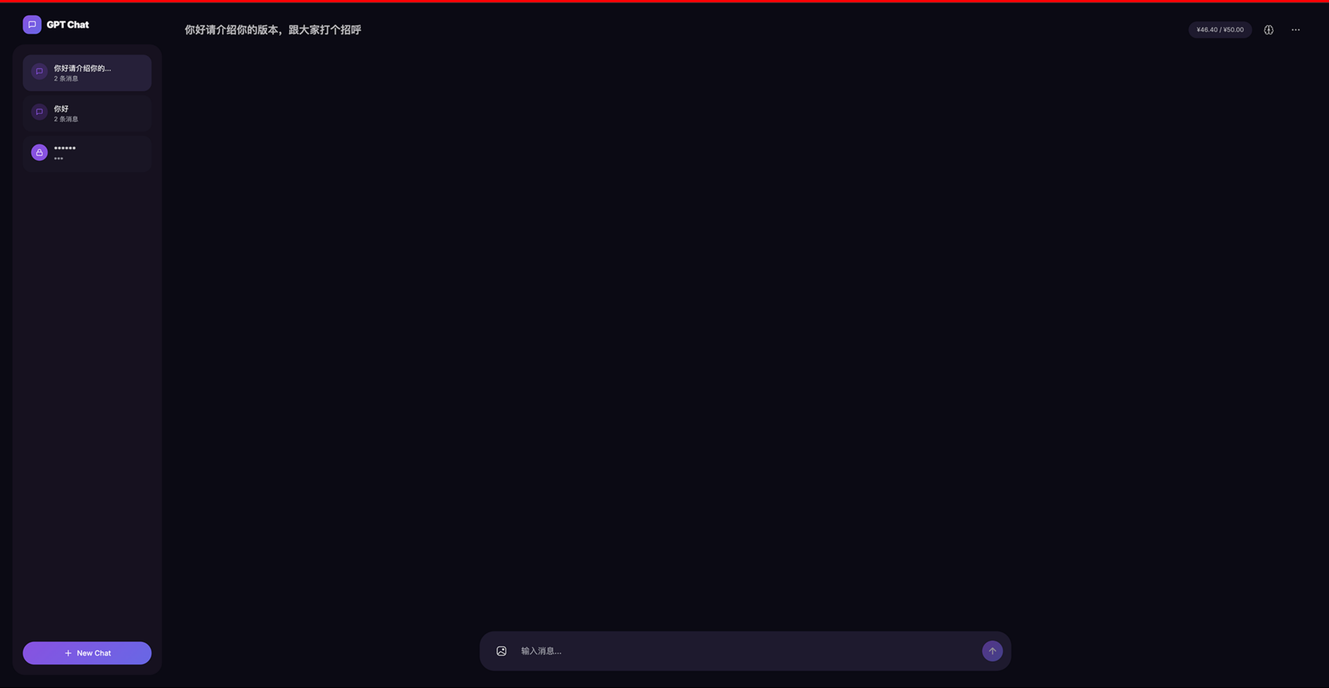
本项目的技术价值体现在多个方面：首先，5 层金字塔摘要算法具有创新性和实用性，可作为长对话场景的通用解决方案；其次，三层记忆系统和 RAG 检索的结合，为 AI 对话应用提供了完整的上下文管理方案；第三，模型智能切换和成本优化策略，为商业化应用提供了可行的成本控制方案。

项目展示了从算法设计、架构实现到工程落地的完整能力，涵盖了前端开发、后端服务、算法优化、性能调优等多个技术领域，具有较高的技术深度和广度。

AI 的高性能对话系统–首页



AI 的高性能对话系统–内页



UI自动化转前端代码工具（这是一个插件工具，没有界面，所以你看下怎么表达。这个工具比较重要）

项目概述
基于 AI 驱动的设计稿自动化转换工具，实现 Figma 设计稿到前端代码的高精度还原。通过自研的设计系统提取算法和多阶段验证机制，将设计师的视觉稿转换为生产级前端代码，还原度达到 90%以上。

核心技术架构

1. 智能设计系统提取
- 自动识别并提取 Figma 设计系统（颜色变量、字体库、组件库、样式规范）
- 智能分析设计稿结构，自动生成前端设计系统代码
- 支持设计系统变更检测与增量更新

2. 多阶段渐进式转换流程
- **阶段一**: 固定宽度精确还原（1:1 像素级匹配）
- **阶段二**: 响应式布局自适应（多断点适配）
- **阶段三**: 移动端独立开发（H5 端）
- **阶段四**: 跨端整合与优化

3. 精确定位与样式还原
- 基于 Figma API 获取精确的元素坐标、尺寸、样式数据
- 支持复杂视觉效果：渐变、阴影、模糊、混合模式、圆角等
- 自动处理切图导出、命名规范、目录映射

4. 交叉验证机制
- **代码数据验证**: 逐项对比 Figma 数据与生成代码的数值匹配度
- **视觉走查验证**: 截图对比，检查布局、间距、对齐、遗漏元素
- **自动修正循环**: 发现问题自动修正，循环验证直到完全一致

技术栈
- **前端框架**: React + TypeScript + Vite
- **样式方案**: Tailwind CSS + CSS Variables
- **设计工具**: Figma Desktop API / MCP 协议
- **版本管理**: Git + 语义化版本控制

项目亮点

高精度还原
- 设计稿还原度达到 **90%**
- 像素级精确定位（x/y/width/height 完全匹配）
- 支持 Figma 所有主流视觉效果

自动化程度高
- 自动识别并导出切图资源（SVG/PNG/JPG）
- 自动生成设计系统代码（变量、组件、样式库）
- 自动处理命名规范、目录结构、文件组织

工程化规范
- 严格的代码规范与注释标准
- 完整的进度文档与变更记录
- 支持多人协作与版本回溯

灵活的适配方案
- 支持 Web 端响应式布局（1024px ~ 设计稿宽度）
- 支持 H5 端独立开发与适配（rem/vw 方案）
- 支持跨端断点切换与整合

应用场景
- 快速将设计稿转换为可交付的前端代码
- 设计系统的自动化提取与维护
- 多端（Web/H5）界面的统一开发流程
- 设计与开发的协同效率提升

项目成果
- 实现了完整的 Figma 到前端代码的自动化转换流程
- 建立了可复用的设计系统提取与管理机制
- 验证了 AI 辅助前端开发的可行性与高效性
- 为团队节省了大量重复性的切图、布局、样式编写工作