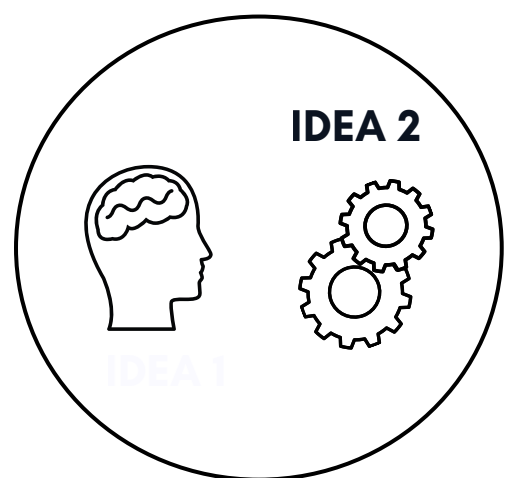
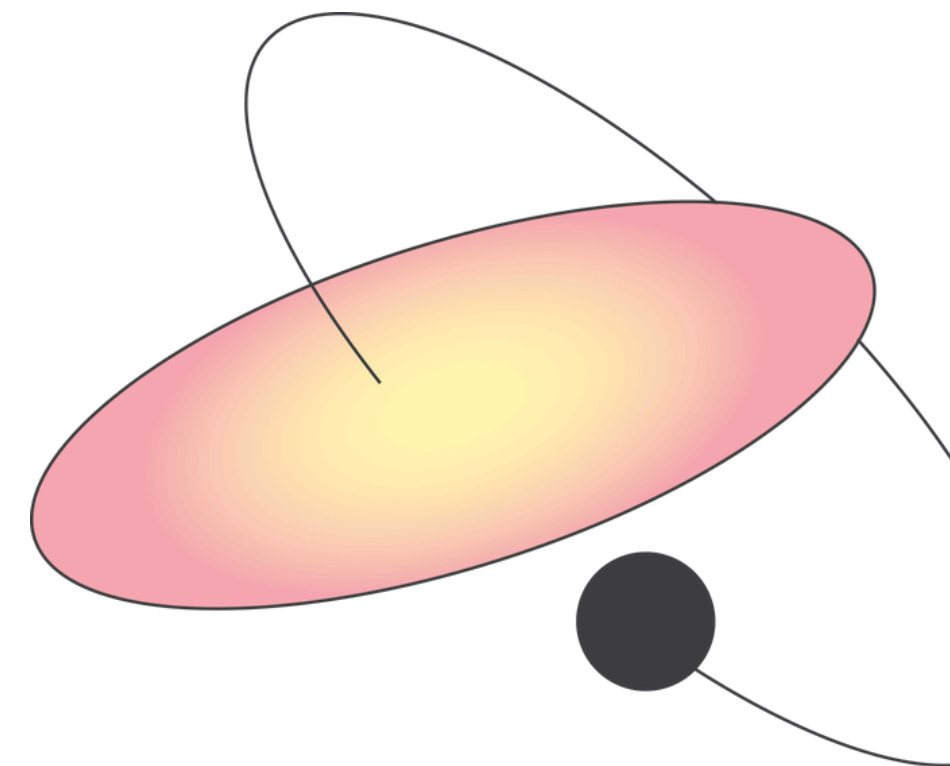


原型-系統設計與部署

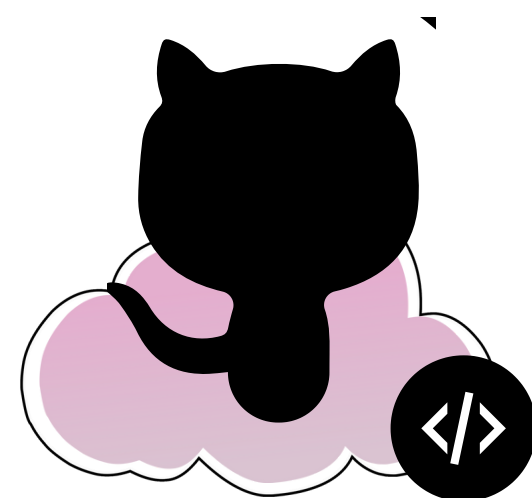
讓我們的網站從電腦 → 雲端 → 可公開使用



終端設計



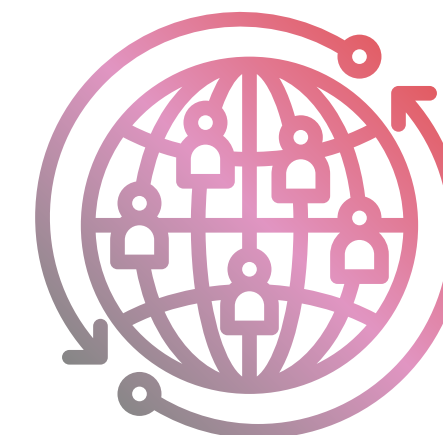
index.html
api/....file
.gitignore
.example



GITHUB

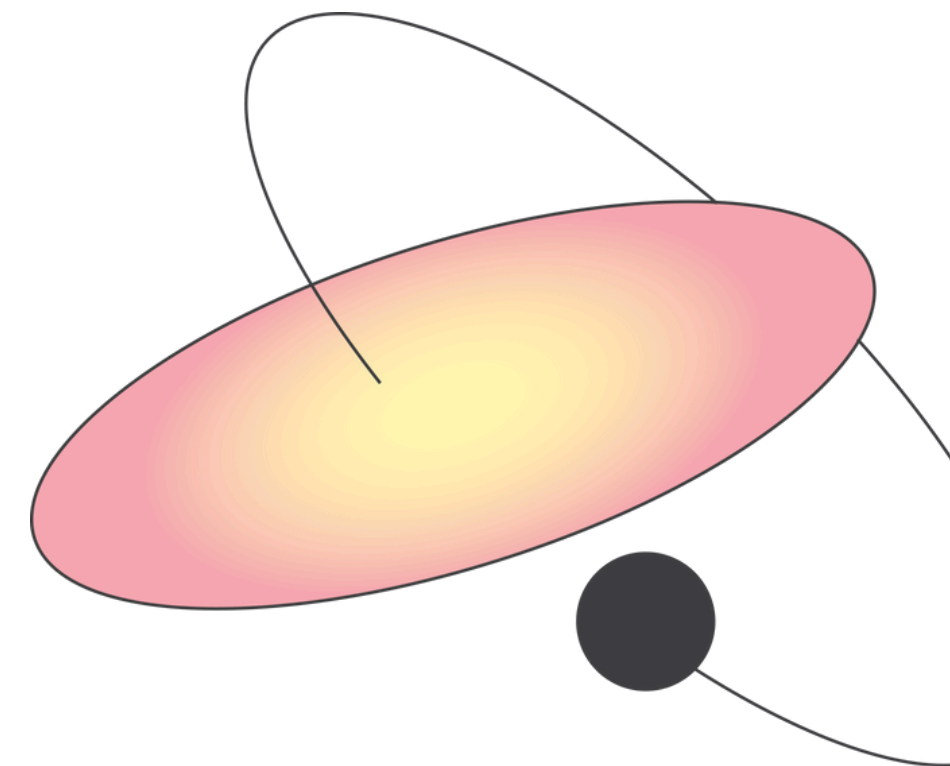


VERCEL



部署
NETWORK

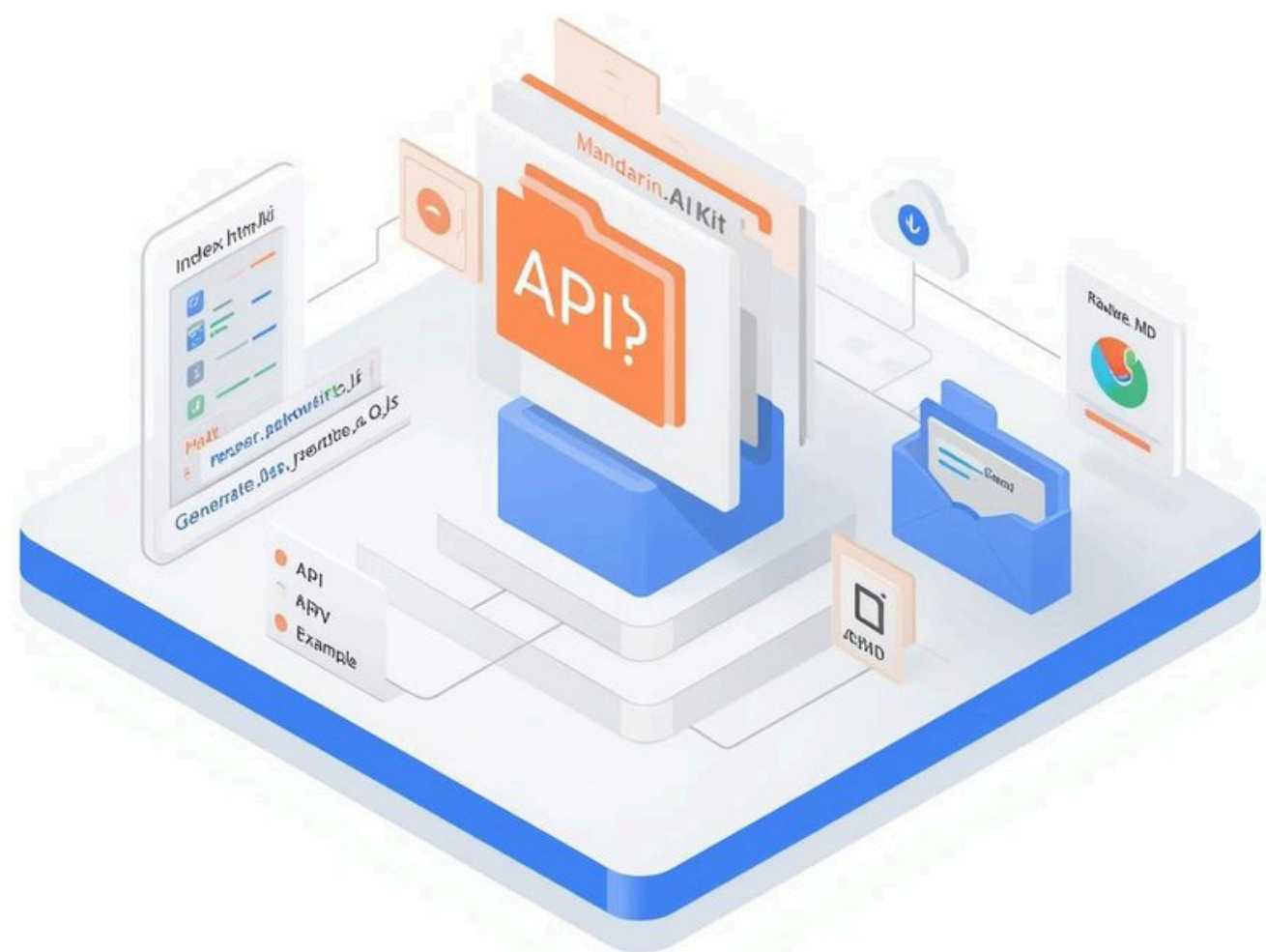
原型-系統執行順序



- 第 步 ● 前端 | 整合目前的 index.html
- 第 步 ● 後端 | 加入 api/generate.js 與 vercel.json
- 第 步 ● 上傳到 GitHub 新專案 → 命名為 mandarin-ai-kit
- 第 步 ● 在 Vercel 建立專案，設定 OPENAI 金鑰
- 第 步 ● 第一次 Deploy → 可運作的 AI 教材生成網站
- 第 步 ● 之後再逐步加入 rewrite.js / qa.js / assess.js 等模組

GitHub + Vercel 雲端部署

最快可上線、有 GPT 互動index.html，
加上 Serverless API 與部署設定



MANDARIN-AI-KIT/

INDEX.HTML

前端互動頁

VERCEL.JSON

告訴 Vercel：
index.html 是靜態
檔、/api 是 Node 函式

API/

SERVERLESS FUNCTIONS (後端)

GENERATE.JS # 生
成教材 (主題/程度/
語法 → 回傳 JSON)

REWRITE.JS #
文本改寫 (難度
調整/形式轉換)

QA.JS # 問答
延伸 (理解題/
討論題)

.GITIGNORE

忽略 node_modules/.env 等

ENV.EXAMPLE

範例環境變數 (OPENAI_API_KEY=
填入API_KEY)

README.MD

專案說明與部署步驟

GitHub + Vercel 雲端部署

最快可上線、有 GPT 互動index.html，
加上 Serverless API 與部署設定

層次	位置	功能	說明
 原始碼層	GitHub	程式碼、HTML、API 函式	上傳的所有檔案都在這裡維護
 執行層	Vercel	自動抓取 GitHub 程式並部署	把前端（靜態頁面）與後端（API 函式）變成可上線的雲端網站
 智能層	OpenAI API	負責生成教材	Vercel 的後端（/api/generate.js）會連線到這裡