# 项目总览与执行顺序（移动端）

形态：Android 与 iOS 原生应用，建议统一采用 Flutter 构建跨端 UI（如需高原生度场景可在关键模块使用 Platform Channels 或原生工程）。后端仍提供 API/任务队列/模型接入。

## 目标

* PDF/教材解析为结构化知识块；
* 基于年级与兴趣进行个性化重写（阅读等级控制、兴趣替换）；
* 生成多种学习素材：沉浸式文本、章节测验、思维导图；
* 在 Flutter App 中采集画像、展示内容、离线缓存与交互测验。

## 架构概览

* 客户端（Flutter）：
* 模块：画像表单、内容浏览（沉浸式文本/导图）、测验作答；
* 数据：Dio/http 调用后端；Riverpod/Bloc 状态管理；sqflite/hive 缓存；
* 原生桥接：文件选择、PDF 预览、分享、通知。
* 服务端：解析、个性化、生成、检索；异步队列与回调/轮询。

## 执行顺序（里程碑）

1. 环境与项目初始化（Flutter + 后端）
2. 数据摄取与 PDF 解析（后端能力）
3. 个性化与阅读等级调整（提示词+评测）
4. 学习素材生成（沉浸式/测验/导图）
5. API 设计与集成（移动端需要的契约）
6. 移动端应用与测试（UI/状态/缓存/权限）
7. 部署与上架（签名、打包、商店物料）

## 测试策略

* 客户端：Widget/Golden 测试、集成测试（flutter test/integration\_test）、E2E（flutter\_driver 替代方案：integration\_test + patrol）。
* 服务端：单元/集成/契约测试；
* 联调：利用 Mock Server 与 Postman Collection；灰度发布 + 崩溃/性能监控（Firebase Crashlytics/Performance）。

## 模型使用策略（已确定）

* 仅使用托管大模型 API，不做本地训练/部署；
* 支持多家 Provider 可切换（OpenAI/Anthropic/Gemini/国内可替代），由环境变量控制；
* 通过提示词工程 + 结构化输出 + 自动化评测保证质量；
* 当且仅当累计足够高质量样本并明确有降本/延迟收益时，才评估微调。