

AI 旅行规划师 (Web)

汇集 LLM 行程生成、预算管理、语音录入与地图展示的全栈示例应用。前端基于 **Next.js 16 + React 19**，后端使用 **Next.js API Routes** 搭配 **Supabase** 提供认证、数据库与实时能力。

功能亮点

- 🧠 **智能行程生成**：调用阿里云百炼（或兼容 OpenAI API 的模型）输出结构化行程、预算及提示信息。
- 📅 **行程管理**：记录每日活动、交通、备注，可随时编辑、删除、重新生成。
- 💰 **预算 & 费用**：支持 LLM 预算拆分、手动新增费用、预算阈值提醒。
- 🗺️ **地图联动**：高德地图展示每日线路，活动卡片与地图点位双向高亮。
- 🗣️ **语音辅助**：内置语音录制组件，可上传识别结果回填表单或自动生成费用草稿。
- 🔄 **实时同步 & 通知**：Supabase Realtime 推送行程/费用变更；预算超支、行程开始等事件触发提醒。
- 📦 **部署友好**：提供多阶段 Dockerfile、`docker-compose` 与 GitHub Actions CI/CD。

快速开始

1. 环境准备

- Node.js **20+**（推荐配合 `corepack` 管理 npm）
- Supabase CLI（可选，用于本地数据库与认证服务）
- Docker / Docker Compose（可选，用于一键启动整套服务）

2. 克隆与安装依赖

```
git clone <your-repo-url> ai-travel-planner
cd ai-travel-planner
npm install
```

3. 配置环境变量

```
cp .env.example .env.local
```

- 根据实际账号修改 `.env.local`。
- 如需 Docker 运行，可参考 `.env.docker`。

4. 初始化 Supabase

```
# 启动本地服务（可选）
supabase start

# 将 schema 推送到本地或远程 Supabase 实例
supabase db push --file supabase/migrations/20241102_init_schema.sql
```

若未使用 Supabase CLI，可登录 Supabase 控制台，在 SQL Editor 执行迁移脚本。

5. 启动开发服务器

```
npm run dev
```

打开 `http://localhost:3000`，使用 Supabase 注册的账号登录即可体验。

环境变量说明

完整示例见 `.env.example`。常用键位说明如下：

变量	说明	备注
<code>SUPABASE_URL</code> / <code>NEXT_PUBLIC_SUPABASE_URL</code>	Supabase 项目 URL	本地默认为 <code>http://127.0.0.1:54321</code>
<code>SUPABASE_SERVICE_ROLE_KEY</code>	服务端密钥	仅后端使用，务必避免泄露
<code>NEXT_PUBLIC_SUPABASE_ANON_KEY</code>	前端匿名密钥	一级权限，仍需安全存储

变量	说明	备注
<code>SUPABASE_JWT_SECRET</code>	Supabase JWT 签名用 secret	与 Supabase 项目保持一致
<code>BAILIAN_API_KEY</code> / <code>BAILIAN_MODEL</code>	大模型访问凭证	兼容 OpenAI 格式
<code>NEXT_PUBLIC_AMAP_KEY</code> / <code>AMAP_REST_KEY</code>	高德 JS SDK 与 Web 服务密钥	前端需配置 Referer 白名单
<code>VOICE_RECOGNIZER_PROVIDER</code>	语音识别提供商	默认 <code>mock</code> ，可切换 <code>iflytek</code> 等
<code>SMTP_HOST</code> 等	邮件通知配置	启用预算提醒或行程提醒时必填
<code>GLOBAL_API_RATE_LIMIT_*</code>	全局限流参数	毫秒窗口 & 最大次数，可在生产调优

更多字段（如 NextAuth、Playwright 绕过凭证）可参考 `.env.example` 注释。

常用脚本

命令	描述
<code>npm run dev</code>	启动开发服务器 (http://localhost:3000)
<code>npm run build</code>	构建生产包
<code>npm run start</code>	以生产模式运行
<code>npm run lint</code>	ESLint + Prettier 校验
<code>npm run test:e2e</code>	Playwright 端到端测试（会自动拉起本地服务）
<code>npm run export:pdf</code>	生成 README 与实现指南的 PDF（需本地安装 <code>md-to-pdf</code> ）
<code>npm run format</code>	使用 Prettier 全量格式化

PDF 导出

执行以下命令会将 `README.md` 与 `docs/AI旅行规划师项目实现指南.md` 转换为 PDF，输出至 `docs/output/` 目录：

```
# 若尚未安装依赖，请先执行
npm install -D md-to-pdf

npm run export:pdf
```

- 输出文件包含 `docs/output/README.pdf` 与 `docs/output/AI旅行规划师项目实现指南.pdf`。
- 可在交付前打包上传，确保评审无需依赖仓库即可阅读。

Docker 部署

```
# 构建镜像
docker build -t ai-travel-planner:latest .

# 或直接使用 compose 启动 (包含 supabase/postgres 依赖)
docker compose up --build
```

- 需要提前将 `.env.docker` 或其他生产环境变量挂载到容器。
- 默认镜像导出 `3000` 端口，可通过 `PORT` 环境变量覆盖。

示例数据与截图

- 示例行程 JSON: `docs/examples/sample-trip.json`，可用于调试 `/api/trips` 或导入前端状态。
- 建议在 README/PDF 中配合应用实际页面截图，展示行程列表、详情、地图与费用面板（可保存在 `docs/screenshots/`）。

质量保障

- CI: 见 `.github/workflows/ci.yml`，在 push/PR 至 `main` 时自动执行 lint、Playwright e2e、构建，主干 push 额外推送 Docker 镜像到 GHCR。
- 测试策略：核心流程覆盖 Playwright；API/服务层可扩展 Jest/RTL 单元测试。

- 速率限制：默认启用，具体阈值由环境变量控制。
-

常见问题

1. 无法获取 Supabase 客户端？

检查 `.env.local` 是否配置 `NEXT_PUBLIC_SUPABASE_URL` 与 `NEXT_PUBLIC_SUPABASE_ANON_KEY`。

2. LLM 调用失败？

查看服务器日志，确认 `BAILIAN_API_KEY`、模型名称及网络连通性。

3. 地图不显示？

确认高德 Key 已添加当前域名至 Referer 白名单，且 JS SDK 正常加载。

4. Playwright 测试卡住？

需提前启动 Supabase（或在 `.env.test` 中配置可用的后端服务），确保 `PLAYWRIGHT_BASE_URL` 与本地地址一致。

如需更多实现细节、业务上下文或迭代记录，请参阅 `docs/AI旅行规划师项目实现指南.md`。欢迎提交 Issue 或 PR，共同完善项目。*** End Patch