# 第一阶段：完善基础提醒功能和多语言框架 (对应文档第一阶段)

## UI 细化 - 首页 (Homepage):

1. **实现卡片布局:** 按照文档要求，将 ListView 中的 ListTile 改为 Card 布局。
2. **突出显示最近事件:** 实现“排在最前面的事件卡片占位最大，显示信息较多”的逻辑。可能需要根据排序后的第一个未完成事件动态调整其样式或布局。
3. **“今天”事件整合:** 实现**仅将到期日为今天 (Today)** 的多个事件整合到一张卡片上显示，按时间排序。未来日期的事件，即使在同一天，也保持独立卡片展示。。
4. **法定/传统节日显示:** 实现用户无提醒时，自动显示当年剩余法定和传统节日的功能（需要节日数据源）。
5. **入口按钮:** 确保首页下方有清晰的语音和文字录入入口（目前是 FAB，可以调整或增加）。

### UI 细化 - 详情页 (DetailPage):

1. **创建详情页:** 目前点击卡片是进入编辑页，需要创建一个独立的详情页 (reminder\_detail\_screen.dart)。
2. **展示基本信息:** 在详情页清晰展示标题、描述、日期时间。
3. **添加编辑/删除按钮:** 在详情页右上角添加编辑和删除按钮，并实现相应的导航或删除逻辑（删除时加入确认提示）。

#### UI 细化 - 设置页 (SettingsPage):

1. **创建设置页:** 创建基础的设置页面 (settings\_screen.dart)。
2. **添加基础选项:** 先实现用户昵称设置、语言切换的入口（逻辑后续实现）。

##### 多语言支持深化:

1. **字符串外部化:** 将 main.dart, add\_reminder\_screen.dart 等界面中所有用户可见的文本（如按钮文字、提示信息、标签）提取到 .arb 文件中 (如 app\_localizations\_en.arb, app\_localizations\_zh.arb)。
2. **实现语言切换逻辑:** 在设置页添加切换语言的选项，并让应用根据选择的语言重新加载对应的本地化字符串。
3. **日期/时间本地化:** 确保 DateFormat 使用正确的 locale (如 DateFormat('yyyy-MM-dd HH:mm', Localizations.localeOf(context).toString())) 来展示日期和时间。
4. **农历支持 (中国区):** 研究并集成支持农历显示的库或方法，并在需要的地方（如首页卡片、详情页）根据用户设置或地区显示农历。

##### 数据存储评估:

1.SharedPreferences 适合存储简单的键值对。随着功能增加（如节日数据、用户设置、可能的联系人关联等），考虑是否需要迁移到更强大的本地数据库，如 sqflite (关系型) 或 isar / hive (NoSQL)。如果未来功能复杂，**尽早迁移**可能更优。

##### 代码健壮性与优化:

1. **错误处理:** 增强 \_loadReminders, \_saveReminders 以及 API 调用部分的错误处理和用户反馈（如使用 SnackBar 或对话框）。
2. **加载状态:** 在耗时操作（如加载、保存、API 调用）时提供更明确的加载指示器。
3. **状态管理:** 目前主要使用 setState。随着页面增多和逻辑变复杂，考虑引入更系统化的状态管理方案（如 Provider, Riverpod, BLoC）来管理应用状态，特别是跨页面的状态共享。

# 第二阶段：集成核心 AI 功能和语音交互 (对应文档第二阶段)

## 完善 AI 自然语言解析:

1. **测试与优化:** 充分测试 extractReminderInfo 对各种自然语言输入的解析效果（包括文档中提到的“明天下午三点”、“下周五晚上七点”、“4月18号”等），根据 DeepSeek 返回结果和 prompt 进行调优。
2. **处理多事件输入:** **这是个难点。** 需要修改 deepseek\_service.dart 中的逻辑和 Prompt，让 AI 能够识别并返回一个包含多个提醒事件的列表 (JSON Array)。然后在 add\_reminder\_screen.dart 中处理这个列表，一次性创建多个 Reminder 对象。
3. **用户反馈:** 如果 AI 解析不确定或缺少信息（如仅有事件无时间），应通过 UI 提示用户补充。

**实现 AI 辅助生成描述:**

1. **完成** processText**:** 在 deepseek\_service.dart 中实现调用 DeepSeek API 生成描述的逻辑，设计合适的 Prompt（可能需要结合提醒的标题和日期）。
2. **集成到 UI:** 在 add\_reminder\_screen.dart 中，当用户点击“AI 辅助生成描述”按钮后，调用 processText 并将返回结果填充到描述输入框。处理好加载和错误状态。

**语音交互流程优化:**

1. **完善语音输入:** 优化 add\_reminder\_screen.dart 中语音输入的启动、停止、结果处理流程，确保用户体验流畅。
2. **对接 AI 解析:** 将语音识别的结果直接传递给 extractReminderInfo 或处理多事件的函数，实现“语音录入 -> AI 解析 -> 生成提醒卡片”的完整流程。

**第三阶段及以后：个性化、全球化、高级功能和商业化**

**实现详情页 AI 功能:**

1. **祝福语生成:** 添加 DeepSeek 调用逻辑，在详情页根据事件类型（生日、节日等）生成祝福语，并实现“换一批”和订阅判断逻辑。
2. **礼物推荐:** 添加 DeepSeek 调用逻辑，根据事件类型和用户信息（如果未来有）推荐礼物。
3. **智能提示:** 根据事件类型、时间、地点（如果未来有）提供相关的提示（如天气、防晒、交通等）。

**个性化功能:** 实现主题切换、自定义背景、自定义通知方式等。

**全球化适配:** 处理时区、集成更多地区节日、遵守隐私法规。

**高级功能:** 根据优先级实现分享（Deep Linking）、联系人关联、数据导入导出、智能分组、家庭共享等文档中提到的高级特性。

**测试与发布:** 持续进行单元测试、集成测试和用户测试，修复 Bug，优化性能，准备发布。

## 总结与建议:

1. **保持迭代:** 认识到这是一个逐步完善的过程，优先保证核心功能的稳定和可用性。
2. **聚焦当前阶段:** 建议先集中精力完成 **第一阶段** 的任务，特别是 UI 细化、多语言框架和核心数据逻辑的稳定。这将为后续复杂功能的集成打下坚实基础。
3. **测试驱动:** 尽早编写测试用例，尤其是针对核心逻辑和 AI 服务的部分。
4. **API Key 安全:** 务必确保 .env 文件不被提交到代码仓库 (.gitignore 中应包含 .env)。

开发中想到的需要调整和修改的：

1、增加节日的来历故事及传统节日的一些风俗习惯知识。