完全离线 App 安全加固流程（含电脑环境与软件）

第一步：获取网站完整源码

- **环境**：Windows / macOS / Linux，普通开发环境即可。
- **软件**：网站编辑器或 IDE（VSCode、WebStorm）、文件传输工具（FTP/SCP 客户端，若有服务器）。（编程工具可按照自己习惯的去选择即可）
- **操作**：将整个网站前端源码（HTML、CSS、JS、JSON、等）导出，作为本地打包基础。

第二步：HTML 层处理

- **环境**：任何文本编辑器或 IDE。
- **软件**：VSCode、Sublime Text、Notepad++。
- **操作**：
  - 删除所有与网络相关的完整功能模块（API）。
  - 保留纯本地展示与本地数据渲染结构。

第三步：JavaScript 层断网处理

- **环境**：Node.js 环境（可选用于代码扫描），IDE。
- **软件**：
  - VSCode / WebStorm
  - Node.js + npm（可用 ESLint 或正则扫描 JS 代码）
- **操作**：
  1. 删除 fetch、axios、XMLHttpRequest、WebSocket 等网络函数。
  2. 删除所有 http://、https:// 字符串。
  3. 在入口 JS 顶部添加强制断网保护代码（覆盖相关 API）。
  4. 确保不存在任何 API 调用逻辑。

第四步：资源层本地化

- **环境**：普通开发电脑即可。
- **软件**：VSCode、图片/字体管理软件（Photoshop / Illustrator / FontForge 可选）、终端工具。
- **操作**：
    1. 所有 JS、CSS、字体、图片必须本地存储。
    2. CSS 中不得包含远程 url。
    3. JSON 数据全部放入本地 data 目录。
    4. 禁止使用 CDN、统计脚本、外部字体。
    5. 开发外壳部分（安全外衣：如外形为计算器，触发特定按钮后才是真正的内容界面）

第五步：Android 层物理断网

- **环境**：Windows / macOS（Android Studio 环境）。
- **软件**：
    - Android Studio
    - Java JDK
    - Cordova CLI（如果使用 Cordova 封装）
- **操作**：
    1. 删除 AndroidManifest.xml 中的 INTERNET 权限。
    2. 禁止 WebView 加载外部 URL。
    3. 关闭调试模式。
    4. 开启代码混淆（Proguard）。

第六步：数据内置方案

- **环境**：Node.js 或任意可处理 JSON / SQLite 的环境。
- **软件**：
    - SQLite Studio / DB Browser for SQLite
    - VSCode / Python（可处理 JSON 转 SQLite）
- **操作**：
    - 方案 A：使用本地 JSON 文件存储数据。
    - 方案 B：使用 SQLite 数据库存储数据（推荐），可加密。

第七步：强制测试流程

- **环境**：Android 设备或模拟器。
- **软件**：
  - Android Studio Emulator 或真机
  - USB 调试 / APK 安装工具
- **操作**：
  1. 打包 APK。
  2. 开启飞行模式。
  3. 彻底断开 WiFi。
  4. 重启 App，确认完全离线运行。

第八步：安全增强建议

- **环境**：同 Android Studio / Node.js 环境。
- **软件**：
  - Java Proguard
  - JS 混淆工具（UglifyJS / Terser / Webpack 插件）
  - SQLite 加密工具（SQLCipher / SQLite Studio）
- **操作**：
  1. JS 混淆。
  2. SQLite 数据加密。
  3. 关闭调试端口。
  4. 使用 Proguard 混淆原生代码。