**1.1 项目基本情况**

Stream游戏推荐系统是基于Next.js的全栈项目，旨在解决玩家游戏选择中的信息过载问题。系统通过AI推荐算法、实时数据同步（Supabase）和交互式界面，实现个性化游戏推荐、收藏管理、排行榜等核心功能。项目周期13周，采用瀑布开发模式，团队规模5人。

**1.2 承担的主要工作情况**

**（1）收藏模块开发**

数据库设计：  
在Supabase中创建collections表，采用半结构化存储（u\_id, g\_ids字段），支持用户-游戏多对多关系。

核心功能实现：

游戏详情页收藏/取消收藏的实时状态切换（使用Supabasesubscribe() 监听变更）

收藏列表拖拽排序功能（持久化排序数据到g\_ids字段）

**（2）个人信息模块开发**

安全架构：  
基于Supabase 实现，敏感操作（如密码修改）强制重新认证。

功能实现：

响应式个人中心页

邮箱/用户名修改的防冲突校验（Supabase唯一索引约束）

**（3）部署手册编写**

* **全流程文档**：  
  编写7步标准化部署指南，覆盖从代码下载到生产启动：

1. 访问 https://github.com/SPM-PSP/SPM25-2/releases   
2. 下载 `stream-game-app.zip`   
3. 解压后执行 `npm install`   
4. 配置 `.env` 文件（Supabase URL+Key）   
5. 启动服务： `npm start`   
6. 访问 http://localhost:3000   
7. 验证收藏拖拽/个人信息修改功能

* **排错附录：**  
  增加常见问题解决方案（如npm install报错处理、端口冲突修复）。

**1.3 实施过程问题及处理**

|  |  |  |
| --- | --- | --- |
| **问题** | **原因** | **解决方案** |
| 收藏列表刷新后排序丢失 | 前端缓存未持久化到数据库 | 增加localStorage过渡层 + 定时同步Supabase |
| 邮箱修改时JWT失效 | Supabase安全策略强制会话刷新 | 增加模态框提示“修改后需重新登录” |
| 移动端拖拽误触率高 | 触摸事件冲突 | 引入react-dnd-touch-backend优化移动体验 |
| Windows部署npm install失败 | node-sass版本兼容性问题 | 锁定依赖版本 |

**1.4 项目体会与评价**

**技术收获**：

* 掌握Supabase实时数据同步机制，成功实现收藏状态跨设备即时更新
* 深入理解拖拽排序的算法设计。

**项目评价**：

* Stream游戏推荐系统作为全栈式技术实践项目，在指定周期内成功实现了智能推荐引擎、用户行为交互系统、实时数据管理等核心功能。核心功能全部上线，技术选型具有前瞻性，但在高并发处理和AI响应优化方面仍有提升空间。