

**Stream 游戏推荐系统**

**项目总结报告**



|  |  |
| --- | --- |
| 学 院： | 软件学院 |
| 专 业： | 软件工程 |
| 组 别： | 第二组 |

二零二五 年 五 月

目录

[1绪论 2](#_Toc200055149)

[1.1项目背景 2](#_Toc200055150)

[1.2相关技术现状 3](#_Toc200055151)

[1.3项目主要开发工作 3](#_Toc200055152)

[1.4项目组成员、分工及完成情况 6](#_Toc200055153)

[2项目策划、分析与设计工作情况 7](#_Toc200055154)

[2.1项目立项计划及完成情况 7](#_Toc200055155)

[2.1.1项目立项计划 7](#_Toc200055156)

[2.1.2项目完成情况 8](#_Toc200055157)

[2.2系统需求分析工作和结果 11](#_Toc200055158)

[2.2.1需求分析工作概述 11](#_Toc200055159)

[2.2.2系统需求分析结果 11](#_Toc200055160)

[2.3系统设计工作及结果 13](#_Toc200055161)

[2.3.1系统架构设计 13](#_Toc200055162)

[2.3.2数据库设计 14](#_Toc200055163)

[2.3.3功能模块设计 14](#_Toc200055164)

[2.3.4人机界面设计 15](#_Toc200055165)

[2.4系统具体实现工作及结果 15](#_Toc200055166)

[2.4.1前端实现 15](#_Toc200055167)

[2.4.2后端实现 16](#_Toc200055168)

[3系统部署与测试工作情况 17](#_Toc200055169)

[3.1系统部署与调试 17](#_Toc200055170)

[3.1.1部署流程 17](#_Toc200055171)

[3.1.2详细步骤 17](#_Toc200055172)

[3.2系统测试 21](#_Toc200055173)

[3.2.1测试概述 21](#_Toc200055174)

[3.2.2测试环境 21](#_Toc200055175)

[3.2.3测试内容及结果 21](#_Toc200055176)

[4项目管理工具使用情况 24](#_Toc200055177)

[5项目讨论与体会 24](#_Toc200055178)

[5.1对项目过程的体会 24](#_Toc200055179)

[5.2项目管理过程优点与不足 25](#_Toc200055180)

[6参考资料 25](#_Toc200055181)

# 1绪论

## 1.1项目背景

2024年夏季，国产3A游戏大作《黑神话：悟空》凭借其精良的制作水准、引人入胜的故事情节和浓厚的中国文化元素，在全球范围内掀起了一股热潮，成为游戏行业和大众关注的焦点。该游戏的成功不仅获得了国内外玩家的高度评价，还彰显了国产游戏在国际市场上的竞争力，使人们重新审视了游戏的文化价值和情感共鸣作用。

然而，在当今复杂多样的游戏市场中，玩家在面对众多游戏作品时往往感到信息过载，难以在质量参差不齐的众多选择中找到真正适合自己的优质游戏。市场中不仅有像《黑神话：悟空》这样的精品佳作，还充斥着大量依靠心理诱导机制牟利的低质量“快餐”游戏，虽然这些游戏可以吸引用户，但却未能真正满足用户的兴趣和需求，反而消耗了大量的时间和金钱。

在这样的市场环境下，如何根据玩家的个性化需求和兴趣，为其提供精准的游戏推荐，成为一个亟待解决的问题。传统的推荐方法主要基于简单的兴趣标签或购买历史，往往忽略了用户行为的深层次模式，导致推荐效果欠佳。

为此，本项目应运而生，旨在构建一个智能化的游戏推荐平台。该平台不仅采用先进的数据分析和推荐算法，深度挖掘和理解用户的兴趣偏好和行为数据，提供个性化的高质量推荐服务，同时还具备丰富的功能模块。平台为用户提供游戏浏览、游戏排行榜查看、游戏搜索、个性化推荐、收藏和评论等互动功能，帮助用户快速定位符合其需求的游戏。同时，用户还可以通过反馈和客服功能，提升用户体验。管理员则可以管理游戏库内容、优化推荐算法、管理广告内容和审核评论等，确保平台运行的质量和安全性。

通过这些功能，本项目致力于为用户打造一个高效、个性化的游戏推荐环境，帮助他们在纷繁复杂的市场中快速找到优质游戏，从而提升用户体验，为游戏行业的可持续发展贡献力量。

## 1.2相关技术现状

国外的游戏推荐系统起步较早，许多指明平台已经在智能推荐方面进行了深入探索和应用。例如：

1. Steam

Steam平台作为全球最大的PC游戏发行平台之一，采用基于用户购买记录、游戏评价和游玩时长的推荐算法，为玩家推荐个性化的游戏内容。Steam使用协同过滤、内容推荐和矩阵分解等技术，结合用户的行为数据，向用户推荐潜在感兴趣的游戏。此外，Steam引入了社交推荐的元素，借助好友和社区互动来优化推荐效果。

1. PlayStation 和 Xbox

索尼的PlayStation和微软的Xbox平台均利用用户数据、游玩记录、成就系统等数据来推荐游戏。通过机器学习和数据分析，这些平台能够识别出玩家的偏好，推荐符合玩家喜好的游戏类型和内容。此外，推荐系统还考虑到了新游戏的推广需求，帮助新游戏获得更多的曝光机会。

1. Netflix 和 Amazon 的推荐技术应用

虽然不属于游戏领域，但Netflix和Amazon的推荐算法也代表了国际上推荐技术的高水平。Netflix采用深度学习和强化学习结合的推荐模型，而Amazon则基于用户行为、商品相似性、协同过滤等技术进行个性化推荐。这些技术在推荐领域的先进性为游戏推荐系统提供了参考。

## 1.3项目主要开发工作

本项目核心内容为开发一个多功能的游戏推荐系统，具体包括用户注册、登录等基础用户功能，以及首页轮播推荐、排行榜、AI 游戏推荐助手等游戏推荐功能。

技术实现上，前端基于 React + Tailwind CSS 实现响应式界面；后端使用 Next.js 构建 API，集成 Supabase 数据库存储用户与游戏数据；AI 模块，通过 Vercel AI SDK 连接 LLM，实现智能对话与游戏推荐。

项目需要实现的功能大致如下表所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 功能名称 | 输入 | 功能详细描述 | 输出 |
| 注册 | 用户名、邮箱、密码、验证码 | 注册时，用户的角色类别默认为客户。用户需要输入系统所提供的验证码，这是为了防止机器暴力破解系统，保证系统的安全性。除此以外，用户还需要输入一些系统要求用户所输入的信息。同时，注册界面需要做的美观，增强用户的体验。 | 注册的结果 |
| 登录 | 用户名、  密码 | 用户如果想要进入网站主页访问系统，则必须先登录。登录时，用户需要输入自己的用户名和密码。 | 登录的结果 |
| 退出登录 | 无 | 用户在登录以后，如果想要切换另外一个账号登录系统，或者退出登录状态，用户需要选择系统中提供的退出登录的选项。 | 退出登录的结果 |
| 个人中心管理 | 用户ID、个人信息数据 | 管理用户个人资料，包括基本信息修改、头像更新、隐私设置等 | 更新后的个人信息 |
| 修改账号信息 | 用户ID、修改项、新数据 | 修改用户账号相关信息，如密码、邮箱、手机号等 | 修改状态、更新后的账号信息 |
| 搜索游戏 | 搜索关键词、筛选条件 | 根据关键词搜索游戏，支持标签筛选、价格区间筛选等 | 搜索结果列表、搜索统计信息 |
| 查看游戏推荐 | 用户ID、浏览历史、用户偏好 | 基于用户兴趣和行为特征推荐游戏 | 推荐游戏列表、推荐理由 |
| 收藏游戏 | 用户ID、游戏ID | 添加/取消游戏收藏，管理收藏列表 | 收藏操作状态、更新后的收藏列表 |
| 评分功能 | 用户ID、游戏ID、评分值 | 对游戏进行评分，更新游戏总体评分 | 评分结果、更新后的游戏评分信息 |
| 评论功能 | 用户ID、游戏ID、评论内容 | 发表游戏评论，查看其他用户评论 | 评论发布状态、评论列表 |
| 智能客服 | 用户问题、会话上下文 | 提供智能化的客服服务，解答用户问题 | 答复内容、相关推荐 |
| 更新游戏库 | 游戏信息、更新类型 | 新增、修改、下架游戏信息 | 更新状态、操作日志 |
| 管理广告内容 | 广告内容、投放设置 | 管理平台广告，包括添加、修改、下架广告 | 广告操作状态、广告统计数据 |
| 更新推荐算法 | 算法参数、训练数据 | 更新和优化游戏推荐算法 | 算法更新状态、性能评估报告 |
| 管理注册用户 | 用户ID、操作类型 | 管理用户账号，包括封禁、解封、权限管理等 | 操作结果、用户状态更新 |
| 审核用户评论 | 评论ID、审核结果 | 审核用户发布的评论内容 | 审核状态、处理结果 |
| 统计分析 | 分析维度、时间范围 | 生成平台运营数据分析报告 | 统计报表、数据可视化图表 |
| 收藏排序 | 排序规则、数据范围 | 对用户收藏的游戏进行智能排序 | 排序结果、排序依据说明 |

## 1.4项目组成员、分工及完成情况

1. 张琛

* 分工：
  + 统筹管理项目进度
  + 调研AI模块并实现
  + 配置数据库
* 完成情况：完成

1. 杨骑骏

* 分工：
  + 实现收藏模块
  + 实现个人信息模块
  + 编写部署手册
* 完成情况：完成

1. 田启泽

* 分工：
  + 实现排行榜模块
  + 参与编写测试文档并执行测试
  + 编写用户使用手册
* 完成情况：完成

1. 孙逍遥

* 分工：
  + 实现游戏详情模块
  + 实现搜索结果模块
* 完成情况：完成

1. 吴佳昊

* 分工：
  + 实现游戏商城模块
  + 参与编写测试文档并执行测试
* 完成情况：完成

# 2项目策划、分析与设计工作情况

## 2.1项目立项计划及完成情况

### 2.1.1项目立项计划

1. 项目背景与立项原因

在游戏行业日益繁荣、用户面临选择困难的背景下，为应对市场上信息过载与推荐质量参差不齐的问题，本项目旨在开发一款智能化的游戏推荐平台——**Stream 游戏推荐系统**，通过 AI 技术和用户行为建模，为用户提供精准、高效的游戏推荐体验。

1. 项目目标

完成用户系统开发，支持注册、登录、个人中心（收藏、评分、评论）功能，确保用户数据安全存储。

实现动态游戏推荐功能，包括首页轮播图与滑动交互、智能排行榜、AI 聊天交互界面等，从而实现个性化的游戏咨询。

1. 核心功能模块

* 功能模块
  + 用户系统：
    - 注册/登录
    - 个人中心：收藏管理、评分记录、评分历史
  + 游戏展示与推荐：
    - 首页动态轮播
    - 多维度排行榜：按照评分、发行时间对游戏进行排行
    - 模糊搜索：支持游戏名称、标签、简介关键词搜索
  + AI交互：
    - 了解界面：通过自然语言提问获取游戏推荐
    - 推荐解释：展示推荐逻辑
  + 技术实现：
    - 响应式前端：适配PC端主流分辨率，支持多种浏览器
    - 后端服务：基于NextJS的API接口和Supabase后端服务存储用户行为与游戏元数据
* 非功能需求
  + 性能：主界面加载时间 <= 1.5s, API响应延迟 <= 300ms
  + 可维护性：提供完整的日志系统与错误监控

1. 技术选型

* 前端：React + NextJS + TailwindCSS
* 后端：Supabase + NextJS API Routes
* AI服务：Vercel AI SDK
* 托管平台：Vercel

1. 开发计划与周期

* 项目总周期：13周
* 阶段划分：
  + 需求设计与分析（第1-2周）
  + 核心功能开发（第3-8周）
  + 测试与优化（第9-11周）
  + 部署与上线（第12-13周）

### 2.1.2项目完成情况

1. 功能实现情况

| **功能模块** | **完成情况** | **备注** |
| --- | --- | --- |
| 注册/登录系统 | ✅ 完成 | 已实现表单验证机制 |
| 用户中心 | ✅ 完成 | 支持资料编辑等 |
| 游戏推荐系统 | ✅ 完成 | 实现了轮播推荐、排行榜、搜索等 |
| AI 推荐助手 | ✅ 完成 | 与 Vercel AI SDK 集成，支持自然语言推荐 |
| 收藏/评分/评论 | ✅ 完成 | 数据同步正常，UI 交互顺畅 |
| 后台管理 | ✅ 完成 | 管理员可进行内容更新、评论审核 |
| 性能优化 | ✅ 完成 | 首页加载时间 < 1.5s，接口响应快 |
| 自动化测试 | ✅ 完成 | 覆盖主要流程，编写 20+ 测试用例 |
| 压力测试与验收 | ✅ 完成 | 模拟高并发访问，系统稳定 |
| 生产环境部署 | ✅ 完成 | 项目已部署至 Vercel，运行稳定 |

1. 团队任务完成情况

**阶段一：**需求分析与设计（2周）

* 第 1 周：
  + 调研用户需求，初步完成项目计划书
  + 分析竞品 steam 平台
  + 初步技术选型
* 第 2 周：
  + 继续讨论改进，完成项目计划书

**阶段二：**核心功能开发（6周）

* 第 3 周：
  + 初始化 Git 仓库，配置 CI/CD 流程
  + 搭建 Next.js 项目框架
  + Supabase 后端服务搭建
* 第 4 周：
  + 用户系统开发（注册/登录）
  + 个人中心基础 UI
* 第 5-6 周：
  + 首页轮播图推荐功能
  + 游戏收藏/评分交互
  + 协同过滤算法开发
* 第 7 周：
  + AI 聊天模块开发
  + Vercel AI SDK 接入
* 第 8 周：
  + 模糊搜索功能
  + 响应式布局适配
  + 性能优化

**阶段三：**测试与优化（3周）

* 第 9 周：
  + 单元测试（用户模块/API）
  + E2E 测试（全流程覆盖）
  + 安全测试
* 第 10 周：
  + 压力测试
  + 性能调优
  + 多分辨率适配验证
* 第 11 周：
  + 用户验收测试
  + 收集反馈并优化 UI/UX
  + 编写用户手册/运维文档

**阶段四：**部署与上线（1周）

* 第 12-13 周：
  + 生产环境部署（Vercel + Supabase）
  + 项目总结报告
  + 结题答辩准备

1. 项目亮点

* **AI 交互体验良好**：通过大语言模型，提升了用户推荐解释的透明度和互动性；
* **前后端分离与自动部署**：通过 Vercel 实现持续集成，部署效率高；

1. 问题与改进方向

| **问题** | **解决措施/建议** |
| --- | --- |
| 第三方 AI 接口速率偶有波动 | 可后续本地化模型部署或设置缓存机制 |
| Supabase 存储速度存在瓶颈 | 后续考虑更换为自建数据库 + Redis 缓存 |
| 功能模块分支合并期间偶有冲突 | 建议进一步细化分支权限与提交规范 |

## 2.2系统需求分析工作和结果

### 2.2.1需求分析工作概述

1. 分析目的

本次需求分析工作的目标是明确系统的功能边界、用户需求和技术实现约束，确保开发团队与用户对系统的期望达成一致。通过对功能性需求、非功能性需求、数据需求、用户特性与系统接口的细致分析，为后续的系统设计、编码与测试提供可靠依据。

1. 分析过程

* 通过调研主流平台的推荐机制，明确目标定位
* 与用户角色模拟反馈进行访谈与头脑风暴，提炼真实使用场景
* 编写用例图、ER图、活动图、时序图，辅助分析系统逻辑

### 2.2.2系统需求分析结果

1. 系统概述

本项目旨在通过技术开发与用户需求结合，构建一个高效、智能的游戏推荐系统，具体目标如下：

* 完成用户系统开发，支持注册、登录、个人中心（收藏、评分、评论）功能，确保用户数据安全存储。
* 实现动态游戏推荐功能，包括首页轮播图与滑动交互、智能排行榜、AI 聊天交互界面等，从而实现个性化的游戏咨询。

通过智能推荐与交互设计，解决用户在游戏选择中的信息过载问题，提供个性化、实时的游戏发现体验。主要面向游戏爱好者和新手玩家，通过融合 AI 对话与传统推荐算法，提升推荐的灵活性与用户参与度，并基于 Supabase 的实时数据同步，确保推荐结果动态更新。

1. 功能性需求概览

**主要功能模块**如下：

| **功能模块** | **说明** |
| --- | --- |
| 用户注册与登录 | 用户身份认证与权限控制 |
| 游戏浏览 | 游戏详情查看、排行榜等功能 |
| 搜索系统 | 支持关键词、标签、筛选条件的游戏检索 |
| 用户交互 | 支持评论、评分、收藏与管理收藏夹 |
| AI 对话客服 | 接入智能客服，处理推荐咨询与问题反馈 |
| 用户中心 | 管理个人信息、偏好设置、历史行为等 |
| 排行榜展示 | 按评分/时间等排序展示流行游戏 |
| 接口集成 | 数据库接口、推荐服务接口、人机交互界面 |

1. 非功能性需求分析

| **类别** | **关键指标/要求** |
| --- | --- |
| 性能 | 推荐响应时间 < 1 秒；推荐计算时间 < 5 秒；支持 ≥5 并发用户请求 |
| 安全性 | 使用 HTTPS、支持用户访问控制、防御 SQL 注入与 XSS 攻击 |
| 可用性 | 故障恢复 < 10 分钟；界面友好；支持技术支持通道 |
| 可扩展性 | 模块可热插拔；支持 AI 服务替换与游戏数据扩展 |
| 可移植性 | 支持 Windows、macOS、Linux 和移动端浏览器 |

1. 数据需求与模型

* **ER 图** 建模了用户、游戏、评分、评论、收藏等核心实体之间的关系；
* 系统需存储结构化数据（用户资料、评分记录等）与非结构化数据（评论文本）；

1. 用户特性分析

* 用户特征呈现出“泛兴趣+高个性”特征，对推荐系统的响应速度和透明度提出更高要求
* 面向对象包括：
  + 普通玩家：关注发现优质游戏、互动反馈
  + 核心玩家：更倾向于精细化筛选和行为驱动推荐

1. 系统边界与接口

* 系统边界明确定义数据采集范围、推荐计算范围与展示层功能；
* 接口设计方面包括：
  + 人机交互接口：登录/注册/行为提交；
  + 后端接口：数据库读写、AI 推荐 API；

## 2.3系统设计工作及结果

### 2.3.1系统架构设计

设计了表示层、业务逻辑层、数据访问层和数据源层四层架构。表示层采用 Nextjs 前端框架，负责用户界面展示和交互，处理用户输入并发送请求到后端，同时接收后端数据进行渲染。业务逻辑层基于 Nextjs 路由 / 服务器行为，接收前端请求，实现用户认证、数据处理、游戏评分排序和 AI 功能交互等逻辑。数据访问层利用 Supabase 客户端抽象与 Supabase PostgreSQL 数据库及 Storage 的交互，为业务逻辑层提供数据访问接口。数据源层包括 Supabase PostgreSQL 数据库存储结构化数据和 Supabase Storage 对象存储服务存储非结构化数据。

### 2.3.2数据库设计

根据系统功能需求，设计了 ER 图，确定了用户（user）、游戏（game）、评论（comment）、收藏（collections）、评分（rating）等实体及其之间的关系。用户与游戏之间存在多对多的评论、收藏和评分关系，评论和收藏由用户发出指向游戏，评分也是用户对游戏的操作。

详细定义了 user、game、comment、collections 和 rating 表的数据字典。user 表包含 U\_id（UUID 类型，用户唯一标识）、username（varchar 类型，用户名）、password（text 类型，用户密码）、signature（text 类型，用户个性签名）等字段。game 表有 G\_id（UUID 类型，游戏唯一标识）、G\_name（varchar 类型，游戏名称）、G\_time（Date 类型，游戏发行日期）、description（text 类型，游戏简介）、style（text 类型，游戏风格）、avg\_rating（float 类型，游戏平均评分）等字段。comment 表包含 U\_id、G\_id、time（Date 类型，评论发布时间）、content（text 类型，评论内容）、U\_name 等字段。collections 表记录 U\_id 和 G\_ids（text 类型，半结构化形式的游戏 id 串）。rating 表包含 U\_id、G\_id 和 score（float 类型，用户评分）字段。

### 2.3.3功能模块设计

将系统功能划分为用户管理模块、游戏推荐模块、游戏信息模块、用户交互模块、系统支撑模块和 AI 模块。用户管理模块包括用户注册、登录、认证、密码管理和个人资料管理等功能。游戏推荐模块涵盖基于游戏特征推荐、生成推荐列表、展示推荐游戏、动态调整推荐和提供推荐理由等功能。游戏信息模块有游戏搜索、分类、详情展示和排行榜生成功能。用户交互模块允许用户对游戏评分、评论、收藏和反馈。系统支撑模块涉及 API 管理、数据存储、缓存管理、负载均衡、异常处理和日志记录等功能。AI 模块则包括智能客服、个性化推荐优化和用户行为分析等功能。

为每个功能模块编写了详细的设计决策，明确各功能的实现方式和目标。例如，用户注册功能支持用户创建新账户，记录用户名、密码等信息；基于游戏特征推荐功能通过分析游戏特点和用户喜好为用户推荐相关游戏等。

### 2.3.4人机界面设计

设计了注册 / 登录页面、主页面、游戏详情页、账户管理页、收藏页、智能客服页、排行榜页、搜索结果页等多种界面。注册 / 登录页面提供用户注册 / 登录的 UI 接口；主页面以轮播图展示热门游戏，支持推荐列表滑动交互；游戏详情页展示评分、评论、游戏相关内容，并支持收藏与评分；账户管理页展示和修改用户个人信息；收藏页展示用户收藏的游戏；智能客服页为用户与 AI 对话的窗口；排行榜页按照评分对游戏进行排名；搜索结果页展示搜索结果。

提供了各界面的截图，展示了界面布局和元素。同时，描述了界面对象与操作，如注册页填写注册表格、跳转至登录页面，登录页填写登录表格、跳转至注册页面，游戏详情页的推荐游戏列表点击跳转、评论区域输入文本发表评论等操作，明确了用户与系统的交互方式。

## 2.4系统具体实现工作及结果

### 2.4.1前端实现

使用 React + Tailwind CSS 构建响应式界面，创建了注册和登录页面。在注册页面，用户可以输入用户名、邮箱、密码、验证码等信息进行注册，系统会将注册信息发送至后端进行处理，并将用户角色默认设置为客户，同时验证验证码以防止机器暴力破解，确保系统安全。登录页面允许用户输入用户名和密码进行登录验证，登录成功后可进入系统主页。

实现了主页面的轮播图展示热门游戏，通过 Next.js 的 API 调用获取游戏数据，并利用 JavaScript 实现轮播图的自动播放和手动切换功能，支持滑动交互查看推荐游戏列表，为用户提供了直观的视觉体验和便捷的游戏浏览方式。

开发了游戏详情页，展示游戏的详细信息，包括评分、评论、文字介绍、图片、视频等。通过从后端获取游戏数据，使用 React + Tailwind CSS 组织页面布局和内容展示，同时实现了收藏与评分功能，用户可以点击收藏按钮将游戏加入收藏夹，对游戏进行评分并提交评分数据至后端。

实现了个人中心页面，用户可以查看和修改个人信息，包括头像、昵称、签名等。收藏页展示了用户收藏的所有游戏，通过从后端获取收藏数据并渲染页面，用户可以方便地管理自己的收藏游戏。

集成了智能客服功能，创建了与 AI 对话的界面。通过调用 Vercel AI SDK 连接大语言模型，用户可以在智能客服页输入问题，系统将问题发送至 AI 服务并获取答复内容，同时提供相关推荐游戏，实现智能化的客户服务。

### 2.4.2后端实现

基于 Next.js 构建 API，实现了用户注册、登录、退出登录、修改密码和个人中心管理等功能。在注册接口中，对用户输入的用户名、密码等信息进行验证和处理，将其存储到 Supabase 数据库中。登录接口验证用户提供的用户名和密码，生成相应的会话或令牌以便用户后续访问系统。退出登录功能销毁用户会话或令牌，使用户安全退出系统。修改密码功能通过验证用户身份和旧密码，更新数据库中的密码信息。个人中心管理功能允许用户查看和编辑自己的个人信息，如头像、昵称、签名等，并将更新后的信息保存到数据库。

设计了 Supabase 数据库结构，创建了游戏相关的表，用于存储游戏的名称、发行日期、简介、风格、平均评分等信息。通过 Next.js API Routes 开发了游戏数据管理接口，实现了添加新游戏、修改游戏信息、下架游戏等功能。当游戏数据更新时，后端通过 API 响应前端请求，将更新后的数据返回给前端进行展示。

开发了基于协同过滤、内容推荐等算法的游戏推荐功能。根据用户的浏览历史、收藏记录、评分数据等行为数据，计算用户与其他用户的相似度，以及游戏之间的相似度，为用户推荐相关游戏。将推荐算法集成到后端服务中，通过 API 接口向前端提供推荐游戏列表和推荐理由，实现个性化的游戏推荐服务。

实现了游戏搜索功能，支持用户通过关键词搜索游戏名称、标签、简介等信息。后端接收到搜索请求后，利用 Supabase 数据库的查询功能，对游戏数据进行模糊匹配搜索，根据搜索关键词的相关度返回搜索结果列表，并通过 API 将搜索结果发送至前端进行展示，同时提供搜索统计信息以便用户了解搜索结果的数量和分布等。

# 3系统部署与测试工作情况

## 3.1系统部署与调试

### 3.1.1部署流程

1. 代码托管

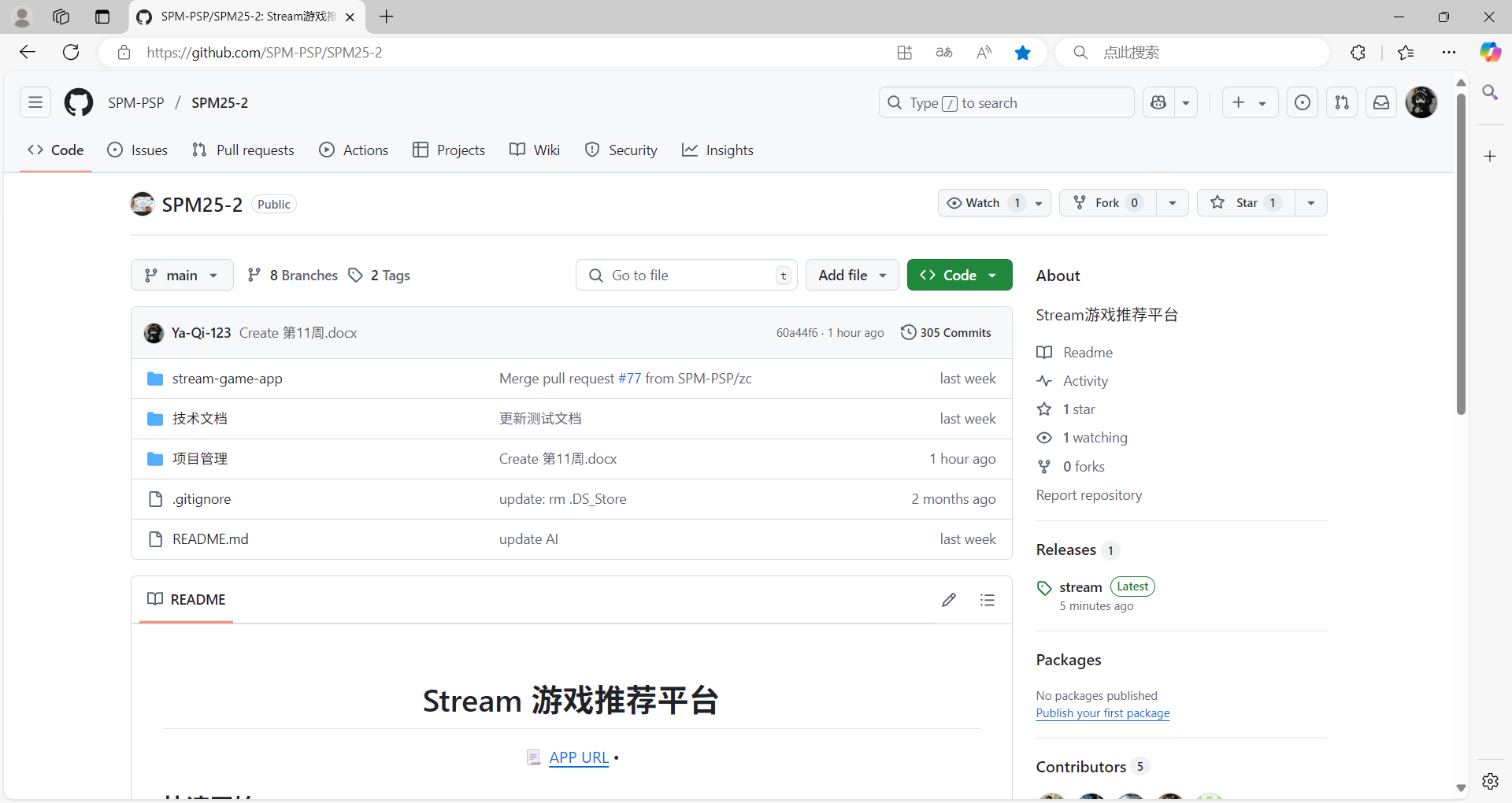
* 使用Github仓库统一管理源码（仓库地址：<https://github.com/SPM-PSP/SPM25-2>）
* Release版本打包为stream-game-app.zip供生产环境使用

1. 本地部署

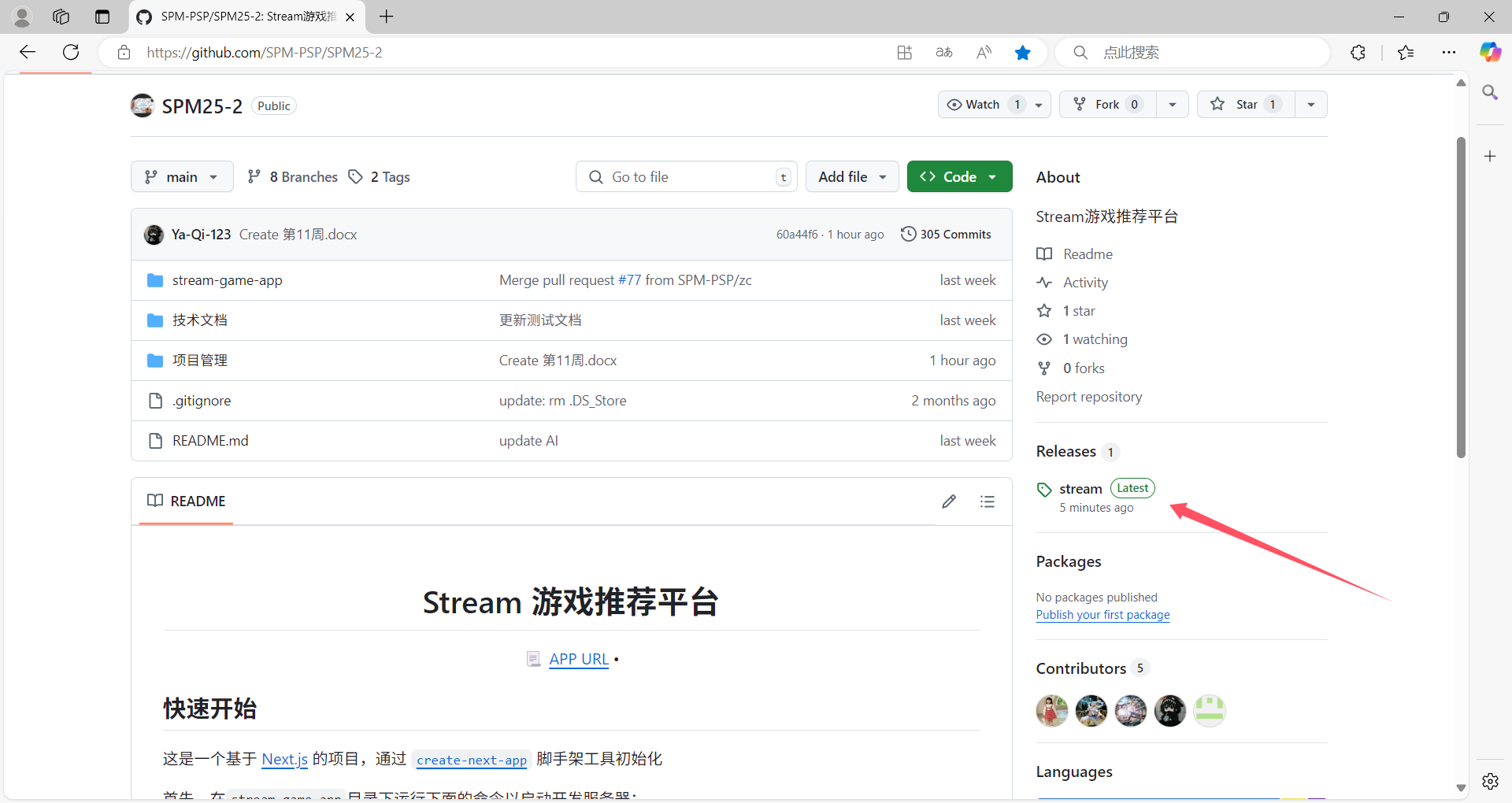
* 解压Release包：unzip stream-game-app.zip -d ./app
* cd ./app
* 安装依赖：npm install
* 启动服务：npm start
* 访问<http://localhost:3000>验证部署成果

### 3.1.2详细步骤

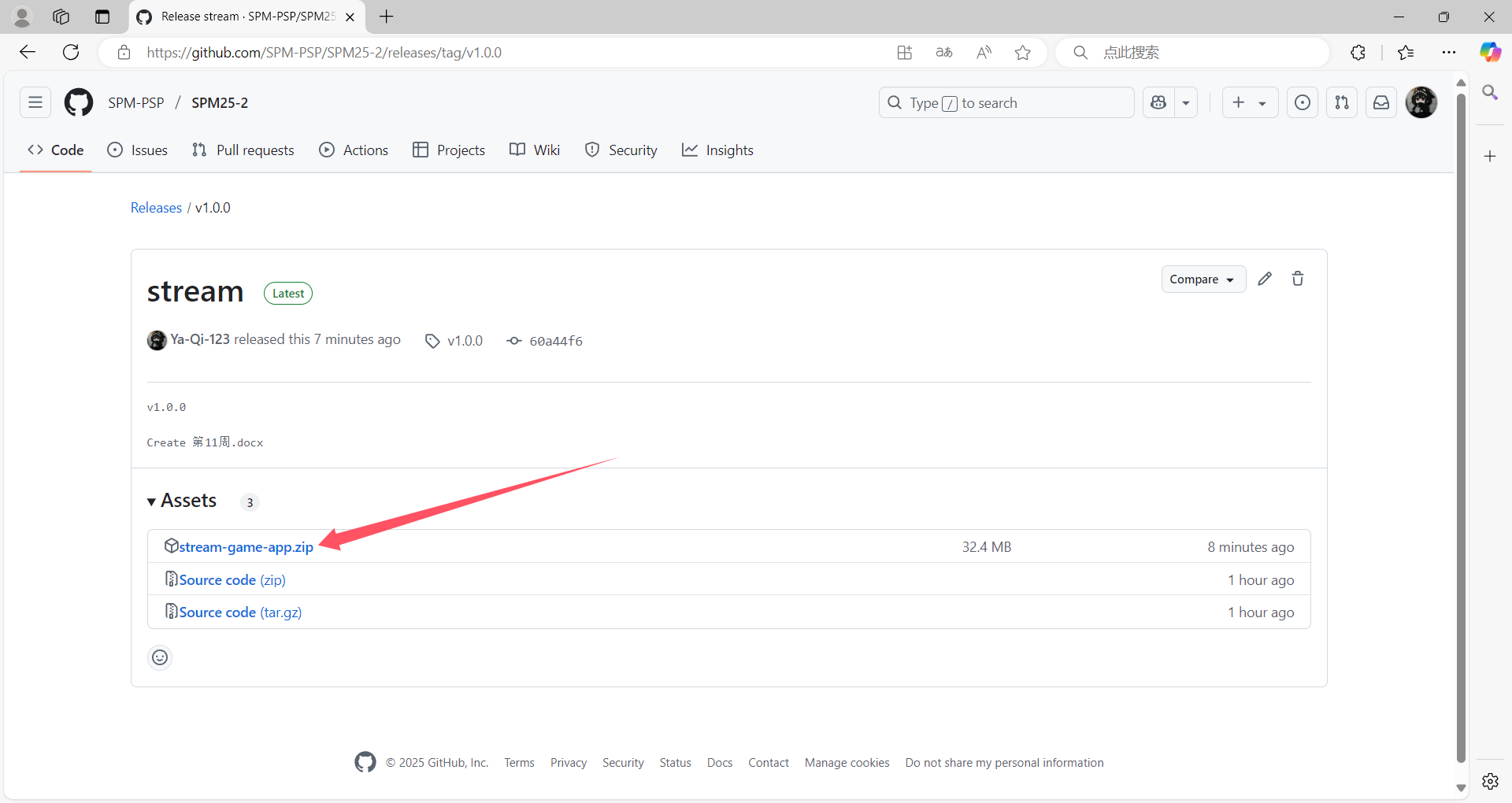
1. 登录Github([SPM-PSP/SPM25-2: Stream游戏推荐平台](https://github.com/SPM-PSP/SPM25-2))



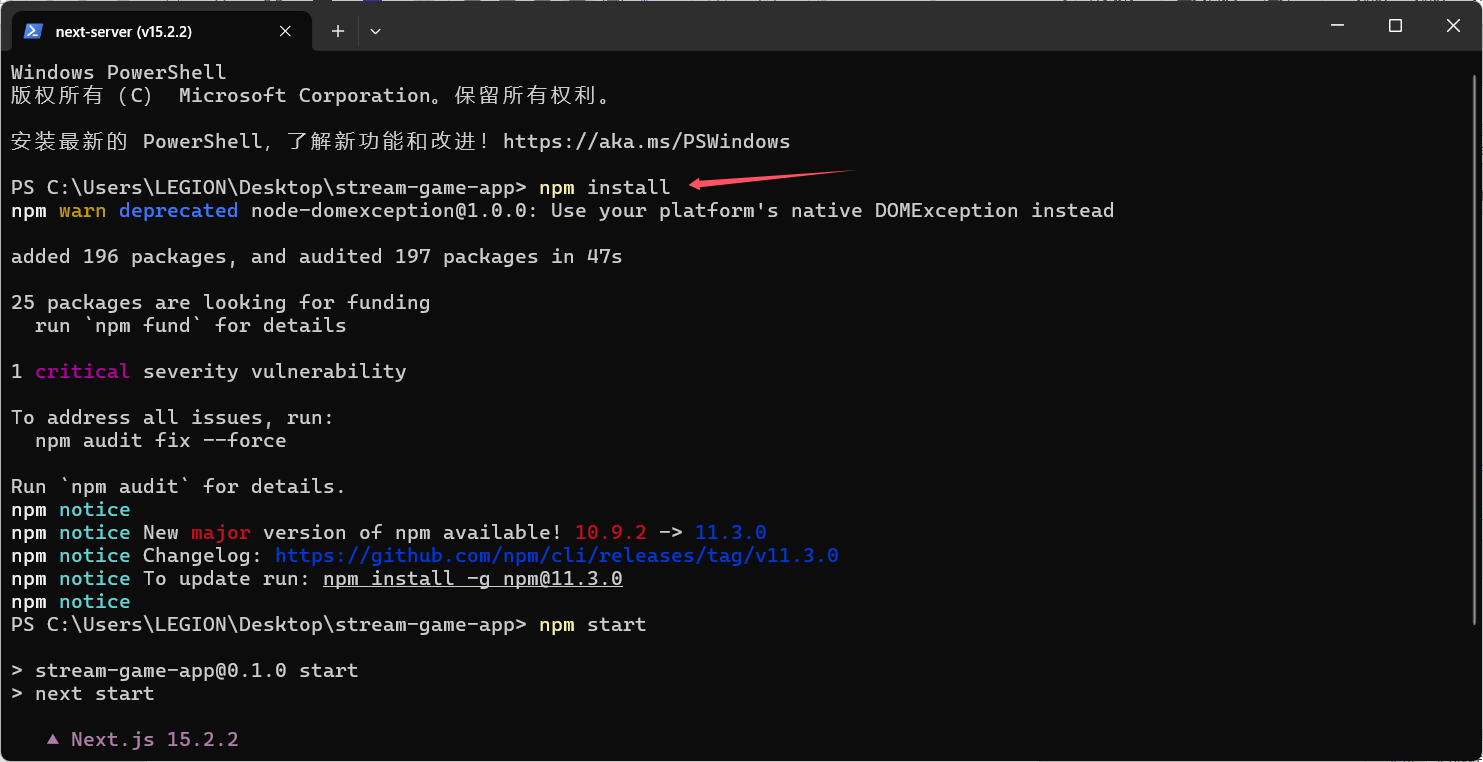
1. 点击Releases



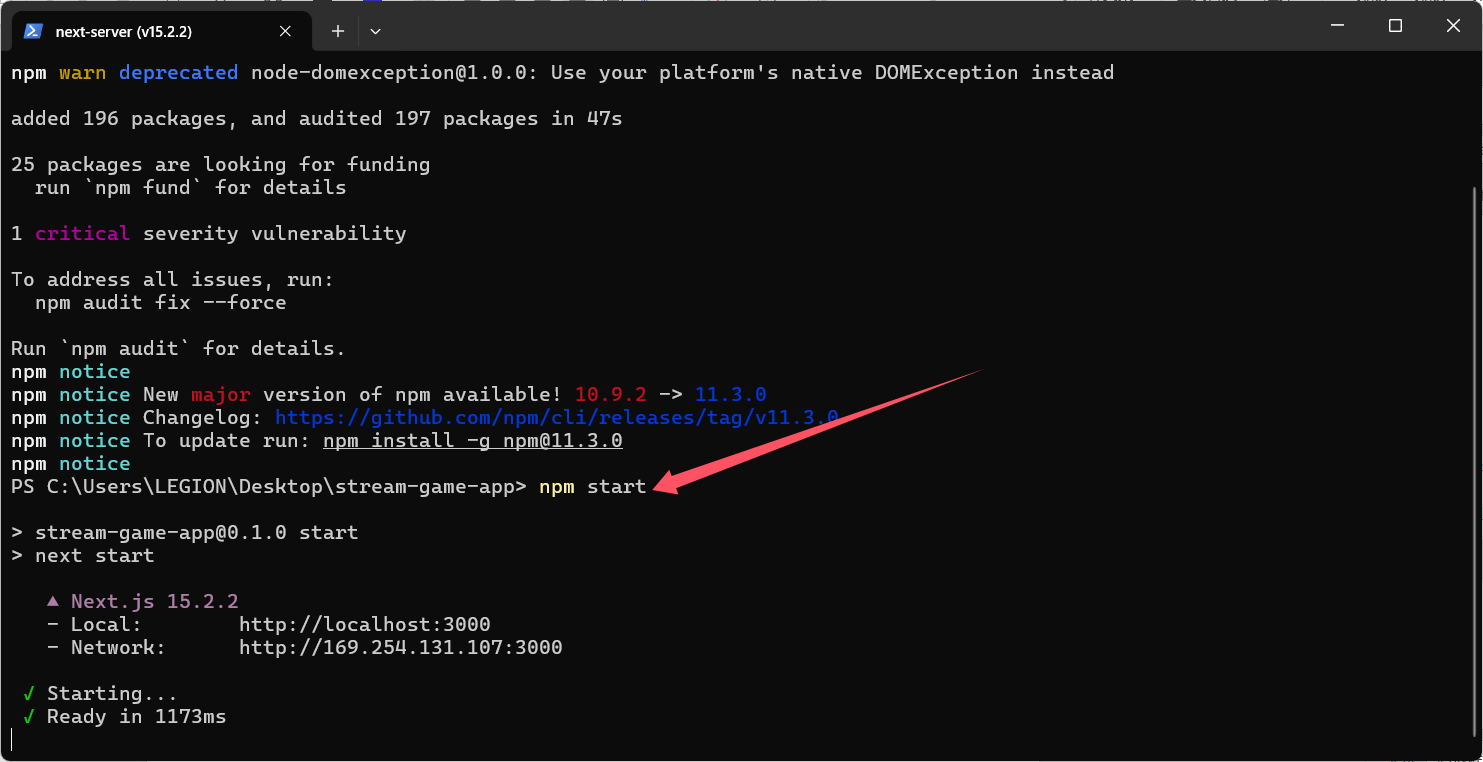
1. 下载stream-game-app



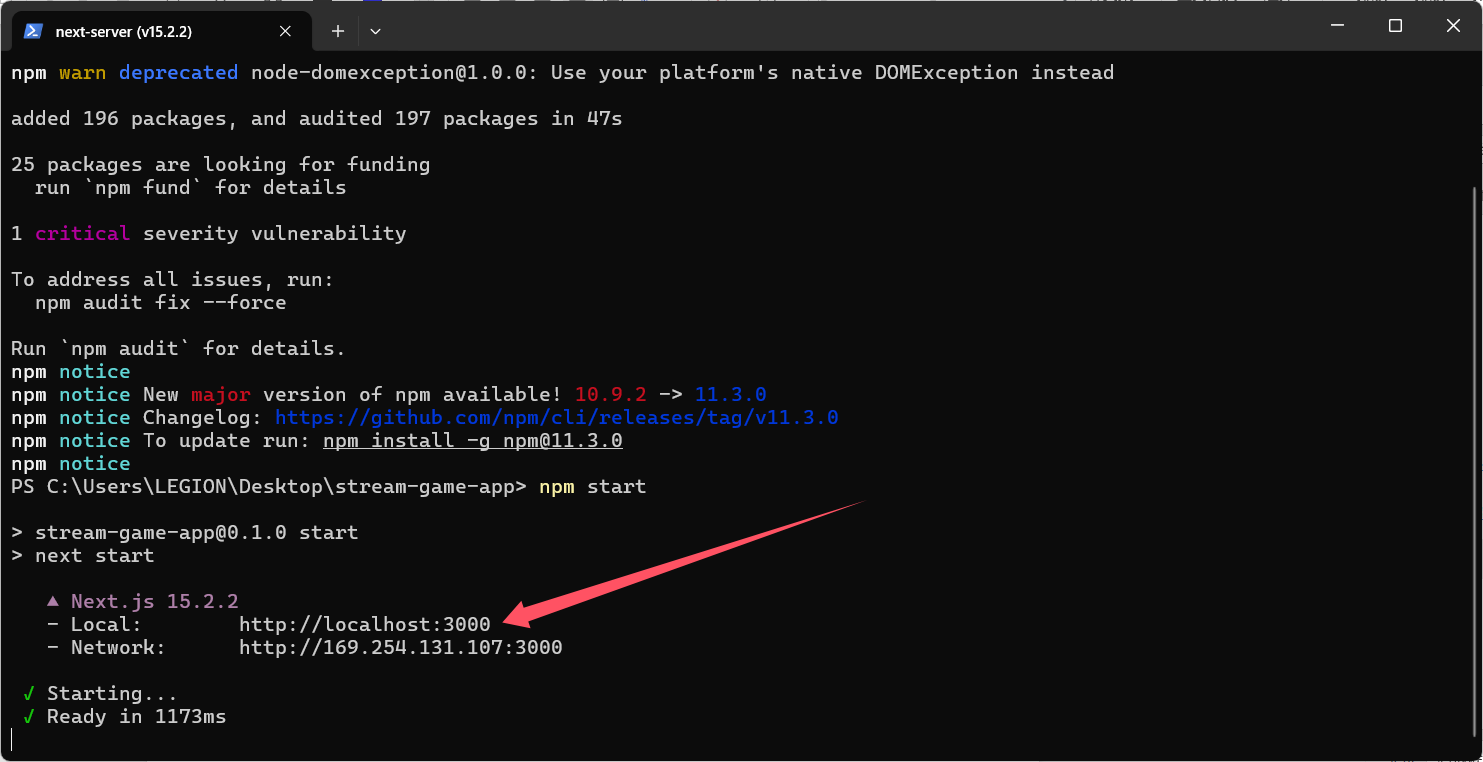
1. 解压文件夹到本地目录
2. 输入npm install



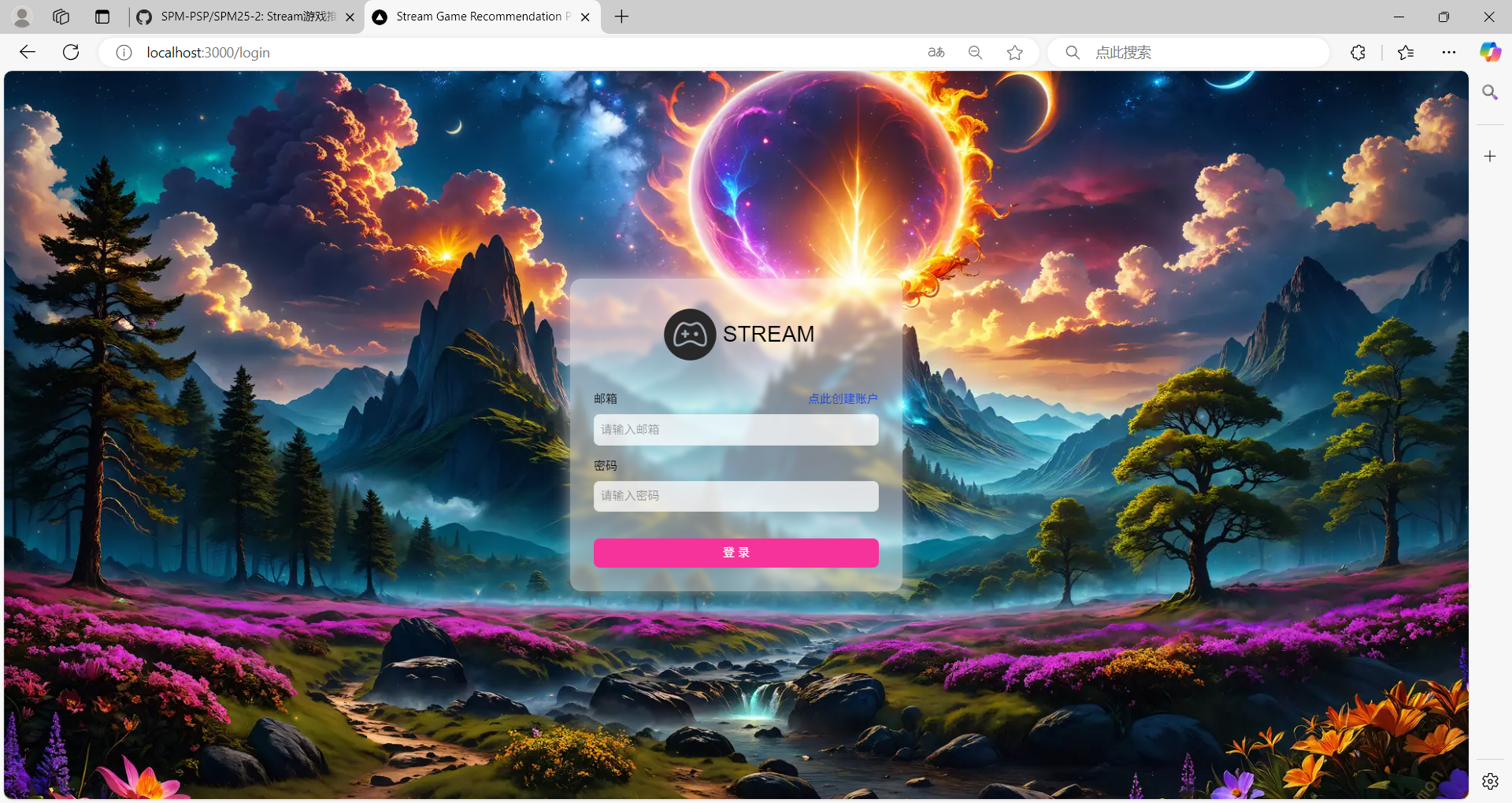
1. 输入npm start



1. 在浏览器中打开网址



看到如下页面则部署成功：



## 3.2系统测试

### 3.2.1测试概述

系统测试阶段围绕Stream游戏推荐系统的功能、性能、安全性及用户体验展开，覆盖了需求规格说明书和系统设计说明书中的所有核心需求。测试目标包括：

* 验证功能模块的正确性和完整性
* 保障数据安全和接口稳定性
* 检查文档与实现的一致性

### 3.2.2测试环境

1. 硬件：个人电脑（CPU：i7以上；内存：16GB；网络：稳定带宽）。
2. 软件：Windows11/macOS/Linux；Chrome 120+/Firefox 115+/Safari 16+
3. 测试工具：Postman；JMeter；Selenium；
4. 数据库：Supabase PostgreSQL

### 3.2.3测试内容及结果

1. 测试内容
2. 功能测试：

|  |  |  |
| --- | --- | --- |
| 功能 | 功能目标 | 测试结果 |
| 登录 | 用户输入正确的账号密码后登录；账号密码输入错误时，系统提示“用户不存在”。 | √ |
| 注册 | 用户在“注册账号”界面输入邮箱后，还需要前后两次秘密输入一致。 | √ |
| 游戏商城 | 游戏商城顶部有游戏海报轮播，向下滑动能查看所有游戏，点击任一游戏后能进入该游戏的详情页面。 | √ |
| 游戏详情 | 游戏视频、海报轮播可供用户查看，用户可以对游戏打分、评论、收藏 | √ |
| 我的收藏 | 展示用户在“游戏详情”页面收藏的游戏，用户可以拖动游戏改变排序。 | √ |
| 排行榜 | 提供“发售时间”和“游戏评分”两种排行供用户参考 | √ |
| Stream  AI | 用户可以提问相关问题，AI进行解答 | √ |
| 账户管理 | 用户可以在这里修改个人昵称和签名 | √ |

1. 性能测试：

|  |  |  |
| --- | --- | --- |
| 测试模块 | 性能目标 | 测试结果 |
| 登录/注册 | 用户输入正确的邮箱密码后，两秒内跳转至游戏商城。 | √ |
| 页面加载 | 游戏详情页、排行榜、收藏夹等核心页面2秒内完成加载。 | √ |
| 并发性 | 多人同时操作时，无明显卡顿出现。 | √ |

1. 兼容性测试：

* 跨平台：在Windows/Mac系统下Edge/Safari等浏览器均可正常访问

1. 测试结果

|  |  |  |
| --- | --- | --- |
| 测试项 | 通过率 | 主要问题 |
| 功能测试 | 100% | 无明显功能缺陷。 |
| 性能测试 | 95% | AI问答响应较慢，测试方法可以进一步完善。 |
| 兼容性测试 | 100% | 无显示问题。 |

1. 问题及改进方案
2. AI客服响应慢：优化框架和AI调用方法，降低AI问答时的延迟。
3. 性能测试：优化测试方法；在计时方面，可以写一个程序计时提高测试的准确性。
4. 总结
5. 所有核心功能完成，可以正常使用。
6. 主要性能指标达标。
7. 能够在不同浏览器中正常显示。

# 4项目管理工具使用情况

|  |  |  |
| --- | --- | --- |
| **工具名称** | **类型** | **核心用途** |
| **GitHub** | 代码管理与协作 | - 代码版本控制（分支管理/合并请求）  - Issue跟踪与任务分配  - CI/CD自动化部署 |
| **VSCode** | 开发环境 | - 代码编写与调试  - 插件扩展（ESLint/Prettier）  - Git集成操作 |
| **Supabase** | 后端即服务(BaaS) | - 数据库托管（PostgreSQL）  - 用户认证管理  - 实时API生成 |

# 5项目讨论与体会

## 5.1对项目过程的体会

在开发Stream游戏推荐系统的过程中，团队深刻体会到以下关键点：

1. 技术整合的挑战与机遇

采用NextJS框架显著提升了开发效率，但初期在Supabase数据库与Vercel AI SDK的集成上遇到兼容性问题。通过查阅官方文档和社区案例，团队成功实现数据流与AI模块的协同，验证了线代技术栈的灵活性。

1. 模块化开发的协作痛点

尽管成员分工明确，但模块间接口定义未在开发初期充分对齐，导致联调阶段出现数据格式不一致问题。

1. 敏捷实践的局限性

项目采用任务拆分制，但未严格执行每日站会，未来需强化迭代评审机制，确保需求闭环。

## 5.2项目管理过程优点与不足

1. 优点：

* **分工合理性**：成员技能与任务高度匹配，任务完成效率高
* **技术选型前瞻性**：TailwindCSS显著提升前段开发效率，响应式设计覆盖多端需求；Supabase简化了用户行为数据的实时处理
* **文档完备性**：用户手册、部署文档及测试文档覆盖全生命周期，降低交付风险。

1. 不足：

* **沟通机制待完善**：模块间依赖仅通过文档传递，缺乏定期技术对齐会
* **进度监控颗粒度不足**：甘特图未细化到子任务层级，影响整体测试排期
* **测试阶段压缩**：系统测试仅预留一周，导致部分边缘场景未充分覆盖，上线后容易出现错误

# 6参考资料

[1] 项目立项计划文档，见文件“项目管理/项目计划书”

[2] 系统需求规格说明书，见文件“技术文档/需求规格说明书”

[3] 系统概要设计文档，见文件“技术文档/系统设计说明书”

[4] 系统测试文档，见文件“技术文档/测试设计”

[5] 用户文档说明书，见文件夹“用户手册”

[6] 项目管理过程文档，见文件夹“项目管理”