



# 暑期编程实训大作业

小组成员：

彭杰	叶志强	薛文杰	姚昊天	黄俊鹏	叶盛举
----	-----	-----	-----	-----	-----

文档链接：[📖 暑期编程实训大作业](#)

Github项目链接：<https://github.com/isexwj/NutriTrack>

- 5区105 答疑时间 9:00-11:30
- 2号-6号 **6号**晚上24:00前提交项目

## 一、项目要求

- 【任务 1】 Web 项目开发：自选题目，开发一款 Web 应用，技术栈Vue + SpringBoot，数据库不限。不少于 5 个核心页面。  
成果提交要求：源码、Git 提交记录及仓库链接、接口文档、部署说明。
- 【任务 2】 基于大模型的 Web 项目开发：使用大模型技术，将其他技术栈的项目（如 React、Django 等），转化为课程所讲技术栈—Vue + SpringBoot，数据库不限。不少于 5 个核心页面。  
成果提交要求：源码、技术栈转换方案、Git 提交记录及仓库链接、接口文档、部署说明。

### 评分依据

- 选题创新度（10%）
  - 是否结合前沿技术、热点问题或实际需求；
  - 题目是否具有新颖性和价值；
  - 是否体现团队自主思考与独创性。
- 程序复杂度（40%）
  - 功能点的数量与覆盖面；

- 技术难度；
- 代码质量（结构清晰、模块化、可读性强、注释合理）；
- 程序稳定性与性能；
- 项目管理质量（20%）
  - Git 使用情况（提交记录完整、分支管理合理、团队协作有迹可循）；
  - 文档规范性；
  - 任务分工是否合理。
- 视频或图片展示效果（30%）
  - 演示内容完整，功能展示清晰，亮点突出；
  - 界面设计美观，交互体验良好（用户体验）。

注：最终参与项目汇报的三个组会在评分的基础上多加2-3分。

可以跳出限制，只要完成了一个项目即可

## 二、项目目标

- 1. 熟悉git命令
- 2. 尽量结合AI，学习一些新的知识
- 3. ……（大家补充）

## 三、项目计划

Gantt



## 四、大家的想法（自由讨论）

- 1. AI工具箱？（大杂烩，可以多类AI工具；如AI面试官[上传简历进行语音面试]、大模型网页搜索等借助多个MCP服务实现的各类应用）
- 2. 阅读（书籍管理、阅读计划、阅读报告、进度跟踪、划线笔记、ai分析推荐）？
- 3. 网页端-饮食健康社区

AI工具箱？（大杂烩，可以多类AI工具）



彭杰 3:49 PM Sep 1

大概的想法就是每个人都可以完成一个 AI 应用，成为一个单独页面，耦合性低。亦或者单个界面有多个功能，如同常见的

4. 小美团-结合高德api，展示地图上的吃喝玩乐，并根据需求推荐
5. 智能菜谱与营养规划师----输入剩余食材或图片，生成菜谱和营养/热量计划（饭菜热量识别、食材输入/识别、菜谱生成、热量营养分析、购物清单、饮食日历。）
6. 或者，能不能把2-5中涉及AI的部分单独提出来，每个功能作为一个页面，最终合成为1，整合为一个项目🤔🤔，感觉可以试试！

五、任务分配

A、数据库设计

1. 用户 (User): 系统的基础，所有操作的主体。
2. 饮食记录 (Meal Record): 核心实体，由用户创建，记录每一餐的详情。
3. 记录图片 (Meal Image): 一条饮食记录可以包含多张图片，所以需要单独一个表来存储。
4. AI健康分析 (AI Health Analysis): 针对每一条饮食记录生成的健康评分和建议。这可以是一对一的关系。
5. 通知 (Notification): 用于实现饭点提醒等功能。

它们之间的关系如下：

- 一个 用户 可以有多条 饮食记录 。（一对多）
- 一条 饮食记录 只能属于一个 用户 。
- 一条 饮食记录 可以有多张 记录图片 。（一对多）
- 一条 饮食记录 有一份对应的 AI健康分析 报告。（一对一）
- 一个 用户 可以接收多条 通知 。（一对多）

1、用户表 users

存储用户注册和个人信息。

字段名	数据类型	约束/索引	备注
id	BIGINT	PRIMARY KEY , AUTO_INCREMENT	用户唯一ID
username	VARCHAR(50)	UNIQUE , NOT NULL	用户名，用于登录
password	VARCHAR(255)	NOT NULL	加密后的密码
email	VARCHAR(100)	UNIQUE	邮箱
phone_number	VARCHAR(20)	UNIQUE	手机号
nickname	VARCHAR(50)		用户昵称
avatar_url	VARCHAR(255)		头像图片URL

DeepSeek 界面，但是需要完成5个界面，这是个问题

文杰

薛文杰 4:28 PM Sep 1

每个人的 AI 应用大概是什么内容，能举个例子吗

彭杰

彭杰 4:37 PM Sep 1

感觉可以自由发挥，比如对话机器人、行程规划助手等各类 Agent

彭杰

彭杰 4:39 PM Sep 1

但是这样就有点不足，给人感觉就是提示词和调用的工具不一样；好处就是每个人都可以进一步去学习 AI 应用的开发，也可以自由发挥

网页端-饮食健康社区

文杰

薛文杰 5:05 PM Sep 1

核心功能，类似于做的笔记作业，记录每天三餐的情况记录；加上是否选择分享到社区（类似于朋友圈）；以及 AI 分析分享到社区的记录的健康度（类似于每天微信步数排行榜）

一、登陆注册全套逻辑（可添加 Redis 验证码），完成后可做 baseline，后续开发建立新的分支。

二、首页仪表盘概览，统计尽些时间的一日三餐情况，哪顿没打卡等（需设置定时任务，每次到饭点，页面需有消息提示）。

三、个人健康饮食记录，类似于笔记，上方可分“全部”、“早餐”、“午餐”、“晚餐”，展示用户的饮食情况，文字+图片（类似于朋友圈）

四、还是个人健康记录 那个页面，添加 AI 总结功能，效果类似于现在共享文档侧边评论区一样，设置成 AI 助手的形式。

五、社区记录，和个人健康饮食记录页面类似，只不过加载的内容变成了所有分享到社区的饮食记录，同时也添加一个 AI 功能，统计当天的社区记录（类似于微信朋友圈步数排行榜），位置可放在当前页，也可放到首页仪表盘。

五个核心功能，加上数据库设计，以及后续可能有新功能开发、解决冲突报错问题等，算是六个人的工作量。

Expand

彭杰

彭杰 5:08 PM Sep 1

我觉得也可以

status	ENUM('active', 'inactive', 'suspended', 'deleted')	NOT NULL , DEFAULT 'active' , INDEX	核心字段: 用户状态。 active :活跃, inactive :未激活, suspended :被封禁, deleted :已注销
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	注册时间
updated_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	信息更新时间
deleted_at	TIMESTAMP	NULLABLE	核心字段: 执行注销操作的时间, status 不为 deleted 时为 NULL

2、饮食记录表 meal\_records

核心业务表，记录用户每一餐的详情。

字段名	数据类型	约束/索引	备注
id	BIGINT	PRIMARY KEY , AUTO_INCREMENT	记录唯一ID
user_id	BIGINT	FOREIGN KEY (users.id) , INDEX	关联的用户ID
meal_type	ENUM('breakfast' , 'lunch', 'dinner', 'snack')	NOT NULL	餐别 (早餐, 午餐, 晚餐, 加餐)
description	TEXT		用户填写的文字描述
calories	INT UNSIGNED	NULLABLE	(新增) 估算的热量 (单位: 大卡 kcal)
rating	DECIMAL(2, 1)	NULLABLE	(新增) 用户评分 (例如 4.5, 5.0)
record_date	DATE	NOT NULL , INDEX	记录的日期
is_shared	BOOLEAN	DEFAULT FALSE , INDEX	是否分享到社区
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	创建时间
updated_at	TIMESTAMP		更新时间

智能菜谱与营养规划师----输入剩余食材...



薛文杰 8:49 PM Sep 1  
这个的意思相当于一个工作流吗？这样页面会不会少很多呀

或者，能不能把2-5中涉及AI的部分单独...



薛文杰 8:41 PM Sep 1  
这个的意思是将 2-5 的 AI 部分合成一个，当成 1 吗？



彭杰 9:43 PM Sep 1  
是的，一个功能对应一个界面，这样的话页面数量也满足要求



薛文杰 10:27 PM Sep 1  
感觉怪怪的



薛文杰 10:33 PM Sep 1  
感觉 1 和现在 kimi 里的工具挺像的，每个人都做一个独立的工具会不会任务量太大了，就算最后做好了会不会也只是图工作量大但是效果很一般的情况呀



		DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAM P	
--	--	--	--

3、记录图片表 meal\_images

存储与饮食记录关联的图片，与 meal\_records 是一对多关系。

字段名	数据类型	约束/索引	备注
id	BIGINT	PRIMARY KEY , AUTO_INCREMENT	图片唯一ID
record_id	BIGINT	FOREIGN KEY (meal_records.id ), INDEX	关联的饮食记录ID
image_url	VARCHAR(255)	NOT NULL	图片存储的URL
upload_order	INT	DEFAULT 0	图片排序，用于前端展示
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAM P	上传时间

4、AI健康分析表 ai\_health\_analysis

字段名	数据类型	约束/索引	备注
id	BIGINT	PRIMARY KEY , AUTO_INCREMENT	分析报告的唯一ID
user_id	BIGINT	FOREIGN KEY (users.id) , INDEX	关联的用户ID
analysis_date	DATE	NOT NULL , INDEX	分析的日期
--- 营养分析 (数值) ---			
health_score	DECIMAL(5, 2)		健康度评分 (例如 0-100.00)
carb_grams	DECIMAL(10, 2)		碳水化合物总克数
protein_grams	DECIMAL(10, 2)		蛋白质总克数
fat_grams	DECIMAL(10, 2)		脂肪总克数



--- 营养分析 (文字) ---			
nutrition_summary	TEXT		对应需求中的【营养分析】文字部分，例如分析三餐的营养均衡性等
--- 建议部分 ---			
health_suggestions	TEXT		对应需求中的【健康建议】，基于分析结果给出宏观建议
improvement_suggestions	TEXT		对应需求中的【改进建议】，给出具体的、可操作的饮食调整方案
--- 元数据 ---			
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	分析报告生成时间
updated_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	报告更新时间

5、通知表 notifications

为了支持点赞、评论等多种互动通知，并实现定位跳转，我们需要对表结构进行如下调整：

- 1. **扩展通知类型：**在 notification\_type 字段中加入 'like' 和 'comment' 类型。
- 2. **记录触发者：**需要知道是谁点赞或评论了你。为此，我们增加一个 sender\_user\_id 字段。
- 3. **明确通知目标：**为了实现跳转，通知必须知道它关联的是哪条“饮食记录”。为此，我们增加一个 target\_record\_id 字段。
- 4. **（可选）更精确的目标：**如果未来要做“回复评论”的通知，还可以加一个 target\_comment\_id。

这是更新后的 notifications 表设计：

字段名	数据类型	约束/索引	备注
id	BIGINT	PRIMARY KEY , AUTO_INCREMENT	通知唯一ID

user_id	BIGINT	FOREIGN KEY (users.id), INDEX	接收通知的用户ID (例如被点赞的人)
sender_user_id	BIGINT	FOREIGN KEY (users.id), INDEX	发送/触发通知的用户ID (例如点赞的人)
notification_type	ENUM(...)	NOT NULL	扩展后的类型: meal_reminder, system, like, comment
target_record_id	BIGINT	FOREIGN KEY (meal_records.id), INDEX, NULLABLE	核心字段: 关联的饮食记录ID, 用于跳转
content	VARCHAR(255)		通知内容摘要 (后端可动态生成)
is_read	BOOLEAN	DEFAULT FALSE, INDEX	是否已读
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	创建时间

现在，当社区发生互动时，后端逻辑如下：

场景1：用户A 点赞了 用户B 的饮食记录 (ID为101)

1. 后端在 record\_likes 表中插入一条点赞记录。
2. 同时，在 notifications 表中插入一条新通知：
  - user\_id: 用户B的ID (接收者)
  - sender\_user\_id: 用户A的ID (触发者)
  - notification\_type: 'like'
  - target\_record\_id: 101 (关联的记录)
  - content: 后端可以动态生成，如 " {用户A的昵称} 赞了你的记录"

场景2：用户C 评论了 用户D 的饮食记录 (ID为205)

1. 后端在 record\_comments 表中插入一条评论记录。
2. 同时，在 notifications 表中插入一条新通知：
  - user\_id: 用户D的ID
  - sender\_user\_id: 用户C的ID
  - notification\_type: 'comment'
  - target\_record\_id: 205
  - content: " {用户C的昵称} 评论了你: {评论内容截取} ..."

其他（系统场景）

1. 由一个预设的、虚拟的“系统用户”作为发送方。

前端如何实现跳转？

当用户在通知列表看到这条通知并点击时：

- 1. 前端获取到该条通知的 `target_record_id` (例如 `101`)。
- 2. 前端根据这个ID，跳转到饮食记录详情页，路由地址可以是 `/records/101`。
- 3. 页面加载后，根据ID请求该记录的详细数据进行展示。

6、点赞表 `record_likes`

这个表只用来记录哪个用户点赞了哪条饮食记录，结构非常简单。

字段名	数据类型	约束/索引	备注
<code>id</code>	<code>BIGINT</code>	<code>PRIMARY KEY</code> , <code>AUTO_INCREMENT</code>	点赞记录唯一ID
<code>record_id</code>	<code>BIGINT</code>	<code>FOREIGN KEY</code> ( <code>meal_records.id</code> ), <code>INDEX</code>	被点赞的饮食记录ID
<code>user_id</code>	<code>BIGINT</code>	<code>FOREIGN KEY</code> ( <code>users.id</code> ), <code>INDEX</code>	点赞的用户ID
<code>created_at</code>	<code>TIMESTAMP</code>	<code>DEFAULT</code> <code>CURRENT_TIMESTAMP</code> <code>P</code>	点赞时间
		<code>UNIQUE KEY</code> ( <code>user_id</code> , <code>record_id</code> )	<b>核心约束：</b> 确保一个用户对同一记录只能点赞一次

7、评论表 `record_comments`

这个表用于存储用户对饮食记录的评论，并为未来的“回复”功能预留了字段。

字段名	数据类型	约束/索引	备注
<code>id</code>	<code>BIGINT</code>	<code>PRIMARY KEY</code> , <code>AUTO_INCREMENT</code>	评论唯一ID
<code>record_id</code>	<code>BIGINT</code>	<code>FOREIGN KEY</code> ( <code>meal_records.id</code> ), <code>INDEX</code>	被评论的饮食记录ID
<code>user_id</code>	<code>BIGINT</code>	<code>FOREIGN KEY</code> ( <code>users.id</code> ), <code>INDEX</code>	发表评论的用户ID



content	TEXT	NOT NULL	评论的具体内容
parent_comment_id	BIGINT	FOREIGN KEY (itself), NULLABLE, INDEX	回复的哪条评论的ID。如果为NULL，则是主评论
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	评论时间
updated_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP	评论编辑时间

关于 parent\_comment\_id 的解释：

这是一个非常重要的字段，它实现了“楼中楼”或“threaded comments”的功能。

- 当一个用户直接评论一条饮食记录时，这个字段的值是 NULL。
- 当一个用户回复另一条评论时，这个字段的值就是被回复评论的 id。

### 数据库脚本

数据库构建和初始数据插入（已更新）

```
1  /*
2   Navicat Premium Dump SQL
3
4   Source Server      : tiger
5   Source Server Type : MySQL
6   Source Server Version : 50731 (5.7.31-log)
7   Source Host        : localhost:3306
8   Source Schema      : nutritrack
9
10  Target Server Type : MySQL
11  Target Server Version : 50731 (5.7.31-log)
12  File Encoding      : 65001
13
14  Date: 02/09/2025 17:42:12
15  */
16
17  SET NAMES utf8mb4;
18  SET FOREIGN_KEY_CHECKS = 0;
19
20  -- -----
21  -- Table structure for daily_diet_analysis
22  -- -----
23  DROP TABLE IF EXISTS `daily_diet_analysis`;
24  CREATE TABLE `daily_diet_analysis` (
25    `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '分析报告的唯一ID',
26    `user_id` bigint(20) NOT NULL COMMENT '关联的用户ID',
27    `analysis_date` date NOT NULL COMMENT '分析的日期',
28    `health_score` decimal(5, 2) NULL DEFAULT NULL COMMENT '健康度评分
29    (例如 0-100.00)',
30    `carb_grams` decimal(10, 2) NULL DEFAULT NULL COMMENT '碳水化合物总
31    克数',
32    `protein_grams` decimal(10, 2) NULL DEFAULT NULL COMMENT '蛋白质总
33    克数',
34    `fat_grams` decimal(10, 2) NULL DEFAULT NULL COMMENT '脂肪总克数',
35    `fiber_grams` decimal(10, 2) NULL DEFAULT NULL COMMENT '膳食纤维总克数',
36    `sugar_grams` decimal(10, 2) NULL DEFAULT NULL COMMENT '糖总克数',
37    `salt_grams` decimal(10, 2) NULL DEFAULT NULL COMMENT '盐总克数',
38    `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
39    `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',
40    PRIMARY KEY (`id`),
41    INDEX `user_id` (`user_id`),
42    INDEX `analysis_date` (`analysis_date`),
43    INDEX `created_at` (`created_at`),
44    INDEX `updated_at` (`updated_at`)
45  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COMMENT='每日饮食分析报告';
```

```
31     `fat_grams` decimal(10, 2) NULL DEFAULT NULL COMMENT '脂肪总克数',
32     `nutrition_summary` text NULL COMMENT '营养文字分析总结',
33     `health_suggestions` text NULL COMMENT '宏观健康建议',
34     `improvement_suggestions` text NULL COMMENT '具体饮食改进建议',
35     `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
COMMENT '报告生成时间',
36     `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP COMMENT '报告更新时间',
37     PRIMARY KEY (`id`) USING BTREE,
38     UNIQUE INDEX `uk_user_date` (`user_id`, `analysis_date`) USING
BTREE,
39     INDEX `idx_user_id` (`user_id`) USING BTREE,
40     INDEX `idx_analysis_date` (`analysis_date`) USING BTREE,
41     CONSTRAINT `fk_daily_diet_analysis_users` FOREIGN KEY
(`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE
NO ACTION
42 ) ENGINE = InnoDB AUTO_INCREMENT = 2      COMMENT = '用户的每日饮食汇总
分析报告' ROW_FORMAT = Dynamic;
43
44 -- -----
45 -- Records of daily_diet_analysis
46 -- -----
47 INSERT INTO `daily_diet_analysis` VALUES (1, 4, '2025-09-01',
75.50, 180.50, 95.20, 70.80, '您今天的早餐和午餐营养均衡，但晚餐的热量和
脂肪摄入较高，导致全天总热量略超标。蛋白质摄入充足，值得表扬。', '建议在享受
聚餐的同时，注意选择清淡一些的锅底和配菜，可以适当减少高脂肪肉类的摄入。保持
日常的运动习惯，有助于热量消耗。', '晚餐时可以尝试用蔬菜或菌菇类代替部分肉
类，增加饱腹感的同时降低脂肪摄入。如果感觉饥饿，可以选择一些低糖水果作为餐后
补充。', '2025-09-02 17:05:56', '2025-09-02 17:05:56');
48
49 -- -----
50 -- Table structure for meal_images
51 -- -----
52 DROP TABLE IF EXISTS `meal_images`;
53 CREATE TABLE `meal_images` (
54     `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '图片唯一ID',
55     `record_id` bigint(20) NOT NULL COMMENT '关联的饮食记录ID',
56     `image_url` varchar(255) NOT NULL COMMENT '图片存储的URL',
57     `upload_order` int(11) NOT NULL DEFAULT 0 COMMENT '图片排序，用于前
端展示',
58     `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
COMMENT '上传时间',
59     PRIMARY KEY (`id`) USING BTREE,
60     INDEX `idx_record_id` (`record_id`) USING BTREE,
61     CONSTRAINT `fk_meal_images_meal_records` FOREIGN KEY
(`record_id`) REFERENCES `meal_records` (`id`) ON DELETE CASCADE
ON UPDATE NO ACTION
62 ) ENGINE = InnoDB AUTO_INCREMENT = 29      COMMENT = '饮食记录的图片'
ROW_FORMAT = Dynamic;
63
64 -- -----
65 -- Records of meal_images
66 -- -----
67 INSERT INTO `meal_images` VALUES (5, 1, '1_20250901_1.jpg', 1,
'2025-09-02 17:34:07');
68 INSERT INTO `meal_images` VALUES (6, 1, '1_20250901_2.jpg', 2,
'2025-09-02 17:34:07');
69 INSERT INTO `meal_images` VALUES (7, 1, '1_20250901_3.jpg', 3,
'2025-09-02 17:34:07');
70 INSERT INTO `meal_images` VALUES (8, 2, '2_20250901_1.jpg', 1,
'2025-09-02 17:34:07');
```

```
71 INSERT INTO `meal_images` VALUES (9, 2, '2_20250901_2.jpg', 2,
72 '2025-09-02 17:34:07');
73 INSERT INTO `meal_images` VALUES (10, 2, '2_20250901_3.jpg', 3,
74 '2025-09-02 17:34:07');
75 INSERT INTO `meal_images` VALUES (11, 3, '3_20250901_1.jpg', 1,
76 '2025-09-02 17:34:07');
77 INSERT INTO `meal_images` VALUES (12, 3, '3_20250901_2.jpg', 2,
78 '2025-09-02 17:34:07');
79 INSERT INTO `meal_images` VALUES (13, 3, '3_20250901_3.jpg', 3,
80 '2025-09-02 17:34:07');
81 INSERT INTO `meal_images` VALUES (14, 4, '4_20250901_1.jpg', 1,
82 '2025-09-02 17:34:07');
83 INSERT INTO `meal_images` VALUES (15, 4, '4_20250901_2.jpg', 2,
84 '2025-09-02 17:34:07');
85 INSERT INTO `meal_images` VALUES (16, 4, '4_20250901_3.jpg', 3,
86 '2025-09-02 17:34:07');
87 INSERT INTO `meal_images` VALUES (17, 5, '5_20250902_1.jpg', 1,
88 '2025-09-02 17:34:07');
89 INSERT INTO `meal_images` VALUES (18, 5, '5_20250902_2.jpg', 2,
90 '2025-09-02 17:34:07');
91 INSERT INTO `meal_images` VALUES (19, 5, '5_20250902_3.jpg', 3,
92 '2025-09-02 17:34:07');
93 INSERT INTO `meal_images` VALUES (20, 6, '6_20250902_1.jpg', 1,
94 '2025-09-02 17:34:07');
95 INSERT INTO `meal_images` VALUES (21, 6, '6_20250902_2.jpg', 2,
96 '2025-09-02 17:34:07');
97 INSERT INTO `meal_images` VALUES (22, 6, '6_20250902_3.jpg', 3,
98 '2025-09-02 17:34:07');
99 INSERT INTO `meal_images` VALUES (23, 7, '7_20250902_1.jpg', 1,
100 '2025-09-02 17:34:07');
101 INSERT INTO `meal_images` VALUES (24, 7, '7_20250902_2.jpg', 2,
102 '2025-09-02 17:34:07');
103 INSERT INTO `meal_images` VALUES (25, 7, '7_20250902_3.jpg', 3,
104 '2025-09-02 17:34:07');
105 INSERT INTO `meal_images` VALUES (26, 8, '8_20250902_1.jpg', 1,
106 '2025-09-02 17:34:07');
107 INSERT INTO `meal_images` VALUES (27, 8, '8_20250902_2.jpg', 2,
108 '2025-09-02 17:34:07');
109 INSERT INTO `meal_images` VALUES (28, 8, '8_20250902_3.jpg', 3,
110 '2025-09-02 17:34:07');
111
112 -- -----
113 -- Table structure for meal_records
114 -- -----
115 DROP TABLE IF EXISTS `meal_records`;
116 CREATE TABLE `meal_records` (
117   `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '记录唯一ID',
118   `user_id` bigint(20) NOT NULL COMMENT '关联的用户ID',
119   `meal_type` enum('breakfast','lunch','dinner','snack') NOT NULL
120 COMMENT '餐别 (早餐, 午餐, 晚餐, 加餐)',
121   `description` text NULL COMMENT '用户填写的文字描述',
122   `calories` int(10) UNSIGNED NULL DEFAULT NULL COMMENT '估算的热量
123 (单位: 大卡 kcal) ',
124   `rating` decimal(2, 1) NULL DEFAULT NULL COMMENT '用户评分 (例如
125 4.5, 5.0)',
126   `record_date` date NOT NULL COMMENT '记录的日期',
127   `is_shared` tinyint(1) NOT NULL DEFAULT 0 COMMENT '是否分享到社区
128 (0:不分享, 1:分享)',
129   `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
130 COMMENT '创建时间',
131   `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON
132 UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',
```

```
107     PRIMARY KEY (`id`) USING BTREE,
108     INDEX `idx_user_id`(`user_id`) USING BTREE,
109     INDEX `idx_record_date`(`record_date`) USING BTREE,
110     INDEX `idx_is_shared`(`is_shared`) USING BTREE,
111     CONSTRAINT `fk_meal_records_users` FOREIGN KEY (`user_id`)
REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE NO ACTION
112 ) ENGINE = InnoDB AUTO_INCREMENT = 9      COMMENT = '用户饮食记录表'
ROW_FORMAT = Dynamic;
113
114 -- -----
115 -- Records of meal_records
116 -- -----
117 INSERT INTO `meal_records` VALUES (1, 4, 'breakfast', '一杯燕麦拿铁，
两片全麦面包，一个煎蛋。开启元气满满的一天！', 350, 4.5, '2025-09-01',
1, '2025-09-02 17:05:56', '2025-09-02 17:05:56');
118 INSERT INTO `meal_records` VALUES (2, 4, 'lunch', '公司楼下的轻食沙
拉，主要是鸡胸肉和蔬菜。', 450, 4.0, '2025-09-01', 0, '2025-09-02
17:05:56', '2025-09-02 17:05:56');
119 INSERT INTO `meal_records` VALUES (3, 4, 'dinner', '晚上和朋友聚餐，吃
了寿喜锅，有点放纵了~', 950, 5.0, '2025-09-01', 1, '2025-09-02
17:05:56', '2025-09-02 17:05:56');
120 INSERT INTO `meal_records` VALUES (4, 5, 'dinner', '自己做的番茄意面，
简单又健康。', 550, 4.5, '2025-09-01', 1, '2025-09-02 17:05:56',
'2025-09-02 17:05:56');
121 INSERT INTO `meal_records` VALUES (5, 4, 'breakfast', '今天起晚了，只
喝了一杯黑咖啡。', 20, 3.0, '2025-09-02', 0, '2025-09-02 17:05:56',
'2025-09-02 17:05:56');
122 INSERT INTO `meal_records` VALUES (6, 4, 'lunch', '点的外卖麻辣香锅，
好吃是好吃，但感觉热量爆炸。', 880, 4.0, '2025-09-02', 1, '2025-09-02
17:05:56', '2025-09-02 17:05:56');
123 INSERT INTO `meal_records` VALUES (7, 7, 'breakfast', '自制牛油果奶
昔，好喝！', 320, 5.0, '2025-09-03', 1, '2025-09-02 17:05:56', '2025-
09-02 17:05:56');
124 INSERT INTO `meal_records` VALUES (8, 6, 'lunch', '健身后的午餐，三文
鱼、西兰花和糙米饭，非常标准的健身餐。', 580, 5.0, '2025-09-02', 1,
'2025-09-02 17:05:56', '2025-09-02 17:05:56');
125
126 -- -----
127 -- Table structure for notifications
128 -- -----
129 DROP TABLE IF EXISTS `notifications`;
130 CREATE TABLE `notifications` (
131     `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '通知唯一ID',
132     `user_id` bigint(20) NOT NULL COMMENT '接收通知的用户ID',
133     `sender_user_id` bigint(20) NOT NULL COMMENT '触发通知的用户ID（系统
消息时为系统用户ID)',
134     `notification_type`
enum('meal_reminder','system','like','comment') NOT NULL COMMENT
'通知类型',
135     `target_record_id` bigint(20) NULL DEFAULT NULL COMMENT '关联的饮食
记录ID，用于跳转',
136     `content` varchar(255) NULL DEFAULT NULL COMMENT '通知内容摘要',
137     `is_read` tinyint(1) NOT NULL DEFAULT 0 COMMENT '是否已读（0:未读，
1:已读)',
138     `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
COMMENT '创建时间',
139     PRIMARY KEY (`id`) USING BTREE,
140     INDEX `idx_user_id_is_read`(`user_id`, `is_read`) USING BTREE,
141     INDEX `idx_sender_user_id`(`sender_user_id`) USING BTREE,
142     INDEX `idx_target_record_id`(`target_record_id`) USING BTREE,
143     CONSTRAINT `fk_notifications_meal_records` FOREIGN KEY
(`target_record_id`) REFERENCES `meal_records` (`id`) ON DELETE
```

```

CASCADE ON UPDATE NO ACTION,
144     CONSTRAINT `fk_notifications_receiver_users` FOREIGN KEY
      (`user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE
      NO ACTION,
145     CONSTRAINT `fk_notifications_sender_users` FOREIGN KEY
      (`sender_user_id`) REFERENCES `users` (`id`) ON DELETE CASCADE ON
      UPDATE NO ACTION
146 ) ENGINE = InnoDB AUTO_INCREMENT = 10      COMMENT = '通用用户通知表'
      ROW_FORMAT = Dynamic;
147
148 -- -----
149 -- Records of notifications
150 -- -----
151 INSERT INTO `notifications` VALUES (1, 4, 5, 'like', 3, '皮皮酱 赞了
      你的记录', 0, '2025-09-02 17:05:56');
152 INSERT INTO `notifications` VALUES (2, 4, 7, 'like', 3, '月之泣 赞了
      你的记录', 1, '2025-09-02 17:05:56');
153 INSERT INTO `notifications` VALUES (3, 5, 4, 'like', 4, '小王几 赞了
      你的记录', 0, '2025-09-02 17:05:56');
154 INSERT INTO `notifications` VALUES (4, 6, 4, 'like', 8, '小王几 赞了
      你的记录', 0, '2025-09-02 17:05:56');
155 INSERT INTO `notifications` VALUES (5, 6, 5, 'like', 8, '皮皮酱 赞了
      你的记录', 0, '2025-09-02 17:05:56');
156 INSERT INTO `notifications` VALUES (6, 4, 5, 'comment', 3, '皮皮酱
      评论了你：看起来也太好吃了吧！哪家店呀？', 0, '2025-09-02 17:05:56');
157 INSERT INTO `notifications` VALUES (7, 6, 9, 'comment', 8, '黑金派
      评论了你：太自律了！求一份详细的食谱参考下。', 0, '2025-09-02 17:05:56');
158 INSERT INTO `notifications` VALUES (8, 8, 1, 'system', NULL, '欢迎加
      入NutriTrack！开始记录你的第一餐，追踪健康生活吧！', 0, '2025-09-02
      17:05:56');
159 INSERT INTO `notifications` VALUES (9, 4, 2, 'meal_reminder',
      NULL, '晚餐时间到啦，别忘了记录今天吃了什么哦~', 0, '2025-09-02
      17:05:56');
160
161 -- -----
162 -- Table structure for record_comments
163 -- -----
164 DROP TABLE IF EXISTS `record_comments`;
165 CREATE TABLE `record_comments` (
166     `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '评论唯一ID',
167     `record_id` bigint(20) NOT NULL COMMENT '被评论的饮食记录ID',
168     `user_id` bigint(20) NOT NULL COMMENT '评论的用户ID',
169     `content` text NOT NULL COMMENT '评论内容',
170     `parent_comment_id` bigint(20) NULL DEFAULT NULL COMMENT '回复的父
      评论ID，NULL为一级评论',
171     `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
      COMMENT '评论时间',
172     `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON
      UPDATE CURRENT_TIMESTAMP COMMENT '评论更新时间',
173     PRIMARY KEY (`id`) USING BTREE,
174     INDEX `idx_record_id` (`record_id`) USING BTREE,
175     INDEX `idx_user_id` (`user_id`) USING BTREE,
176     INDEX `idx_parent_comment_id` (`parent_comment_id`) USING BTREE,
177     CONSTRAINT `fk_record_comments_meal_records` FOREIGN KEY
      (`record_id`) REFERENCES `meal_records` (`id`) ON DELETE CASCADE
      ON UPDATE NO ACTION,
178     CONSTRAINT `fk_record_comments_parent_comment` FOREIGN KEY
      (`parent_comment_id`) REFERENCES `record_comments` (`id`) ON
      DELETE CASCADE ON UPDATE NO ACTION,
179     CONSTRAINT `fk_record_comments_users` FOREIGN KEY (`user_id`)
      REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE NO ACTION

```

```
180 ) ENGINE = InnoDB AUTO_INCREMENT = 5      COMMENT = '饮食记录评论表（支持楼中楼）' ROW_FORMAT = Dynamic;
181
182 -- -----
183 -- Records of record_comments
184 -- -----
185 INSERT INTO `record_comments` VALUES (1, 3, 5, '看起来也太好吃了吧！哪家店呀?', NULL, '2025-09-02 17:05:56', '2025-09-02 17:05:56');
186 INSERT INTO `record_comments` VALUES (2, 3, 4, '就在公司附近新开的那家哦，下次一起去!', 1, '2025-09-02 17:05:56', '2025-09-02 17:05:56');
187 INSERT INTO `record_comments` VALUES (3, 8, 9, '太自律了！求一份详细的食谱参考下。', NULL, '2025-09-02 17:05:56', '2025-09-02 17:05:56');
188 INSERT INTO `record_comments` VALUES (4, 8, 6, '没问题，我晚点整理一下发给你~', 3, '2025-09-02 17:05:56', '2025-09-02 17:05:56');
189
190 -- -----
191 -- Table structure for record_likes
192 -- -----
193 DROP TABLE IF EXISTS `record_likes`;
194 CREATE TABLE `record_likes` (
195     `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '点赞记录唯一ID',
196     `record_id` bigint(20) NOT NULL COMMENT '被点赞的饮食记录ID',
197     `user_id` bigint(20) NOT NULL COMMENT '点赞的用户ID',
198     `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
199     COMMENT '点赞时间',
200     PRIMARY KEY (`id`) USING BTREE,
201     UNIQUE INDEX `uk_user_record` (`user_id`, `record_id`) USING
202     BTREE COMMENT '联合唯一索引，防止重复点赞',
203     INDEX `idx_record_id` (`record_id`) USING BTREE,
204     INDEX `idx_user_id` (`user_id`) USING BTREE,
205     CONSTRAINT `fk_record_likes_meal_records` FOREIGN KEY
206     (`record_id`) REFERENCES `meal_records` (`id`) ON DELETE CASCADE
207     ON UPDATE NO ACTION,
208     CONSTRAINT `fk_record_likes_users` FOREIGN KEY (`user_id`)
209     REFERENCES `users` (`id`) ON DELETE CASCADE ON UPDATE NO ACTION
210 ) ENGINE = InnoDB AUTO_INCREMENT = 6      COMMENT = '饮食记录点赞表'
211 ROW_FORMAT = Dynamic;
212
213 -- -----
214 -- Records of record_likes
215 -- -----
216 INSERT INTO `record_likes` VALUES (1, 3, 5, '2025-09-02 17:05:56');
217 INSERT INTO `record_likes` VALUES (2, 3, 7, '2025-09-02 17:05:56');
218 INSERT INTO `record_likes` VALUES (3, 4, 4, '2025-09-02 17:05:56');
219 INSERT INTO `record_likes` VALUES (4, 8, 4, '2025-09-02 17:05:56');
220 INSERT INTO `record_likes` VALUES (5, 8, 5, '2025-09-02 17:05:56');
221
222 -- -----
223 -- Table structure for users
224 -- -----
225 DROP TABLE IF EXISTS `users`;
226 CREATE TABLE `users` (
227     `id` bigint(20) NOT NULL AUTO_INCREMENT COMMENT '用户唯一ID',
228     `username` varchar(50) NOT NULL COMMENT '用户名，用于登录',
229     `password` varchar(255) NOT NULL COMMENT '重要：存储Bcrypt等算法加密后的密码哈希值，严禁存储明文！',
230     `email` varchar(100) NULL DEFAULT NULL COMMENT '邮箱，可用于登录或找回密码',
231     `phone_number` varchar(20) NULL DEFAULT NULL COMMENT '手机号，可用于登录或找回密码',
232     `nickname` varchar(50) NULL DEFAULT NULL COMMENT '用户昵称，在社区显示',
233 
```



```
227     `avatar_url` varchar(255) NULL DEFAULT NULL COMMENT '头像图片
URL',
228     `status` enum('active','inactive','suspended','deleted') NOT
NULL DEFAULT 'active' COMMENT '用户状态: active-活跃, ina-未激
活, suspended-被封禁, deleted-已注销',
229     `created_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
COMMENT '注册时间',
230     `updated_at` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP COMMENT '信息更新时间',
231     `deleted_at` timestamp NULL DEFAULT NULL COMMENT '执行注销操作的时
间, status不为deleted时为NULL',
232     PRIMARY KEY (`id`) USING BTREE,
233     UNIQUE INDEX `uk_username`(`username`) USING BTREE,
234     UNIQUE INDEX `uk_email`(`email`) USING BTREE,
235     UNIQUE INDEX `uk_phone_number`(`phone_number`) USING BTREE,
236     INDEX `idx_status`(`status`) USING BTREE
237 ) ENGINE = InnoDB AUTO_INCREMENT = 10      COMMENT = '用户信息表 (包含
软删除逻辑)' ROW_FORMAT = Dynamic;
238
239 -- -----
240 -- Records of users
241 -- -----
242 INSERT INTO `users` VALUES (1, 'system',
'$2a$10$e.ExV2pYm1514.e1XAn.2u0g8sLnsd/EMmTEUv362sS3d22Pyi7.q',
NULL, NULL, '系统通知', 'https://i.pravatar.cc/150?u=system',
'active', '2025-09-02 17:05:56', '2025-09-02 17:05:56', NULL);
243 INSERT INTO `users` VALUES (2, 'reminder',
'$2a$10$e.ExV2pYm1514.e1XAn.2u0g8sLnsd/EMmTEUv362sS3d22Pyi7.q',
NULL, NULL, '用餐提醒', 'https://i.pravatar.cc/150?u=reminder',
'active', '2025-09-02 17:05:56', '2025-09-02 17:05:56', NULL);
244 INSERT INTO `users` VALUES (3, 'suggestion',
'$2a$10$e.ExV2pYm1514.e1XAn.2u0g8sLnsd/EMmTEUv362sS3d22Pyi7.q',
NULL, NULL, '健康建议', 'https://i.pravatar.cc/150?u=suggestion',
'active', '2025-09-02 17:05:56', '2025-09-02 17:05:56', NULL);
245 INSERT INTO `users` VALUES (4, 'xwj',
'$2a$10$ejkSpdnmxFRYnS2PIfinM.tbSesjCaOV1rUT.f9hEMWULkeNj6FUS',
'xwj@example.com', '13800138001', '小王几',
'https://i.pravatar.cc/150?u=xwj', 'active', '2025-09-02
17:05:56', '2025-09-02 17:07:52', NULL);
246 INSERT INTO `users` VALUES (5, 'pj',
'$2a$10$e.ExV2pYm1514.e1XAn.2u0g8sLnsd/EMmTEUv362sS3d22Pyi7.q',
'pj@example.com', '13800138002', '皮皮酱',
'https://i.pravatar.cc/150?u=pj', 'active', '2025-09-02 17:05:56',
'2025-09-02 17:05:56', NULL);
247 INSERT INTO `users` VALUES (6, 'yht',
'$2a$10$e.ExV2pYm1514.e1XAn.2u0g8sLnsd/EMmTEUv362sS3d22Pyi7.q',
'yht@example.com', '13800138003', '樱花糖',
'https://i.pravatar.cc/150?u=yht', 'active', '2025-09-02
17:05:56', '2025-09-02 17:05:56', NULL);
248 INSERT INTO `users` VALUES (7, 'yzq',
'$2a$10$e.ExV2pYm1514.e1XAn.2u0g8sLnsd/EMmTEUv362sS3d22Pyi7.q',
'yzq@example.com', '13800138004', '月之泣',
'https://i.pravatar.cc/150?u=yzq', 'active', '2025-09-02
17:05:56', '2025-09-02 17:05:56', NULL);
249 INSERT INTO `users` VALUES (8, 'ysj',
'$2a$10$e.ExV2pYm1514.e1XAn.2u0g8sLnsd/EMmTEUv362sS3d22Pyi7.q',
'ysj@example.com', '13800138005', '艺术家',
'https://i.pravatar.cc/150?u=ysj', 'active', '2025-09-02
17:05:56', '2025-09-02 17:05:56', NULL);
250 INSERT INTO `users` VALUES (9, 'hjp',
'$2a$10$e.ExV2pYm1514.e1XAn.2u0g8sLnsd/EMmTEUv362sS3d22Pyi7.q',
'hjp@example.com', '13800138006', '黑金派',
```

```
'https://i.pravatar.cc/150?u=hjp', 'active', '2025-09-02
17:05:56', '2025-09-02 17:05:56', NULL);
251
252 SET FOREIGN_KEY_CHECKS = 1;
```

B、大概功能

六、GitHub操作要点

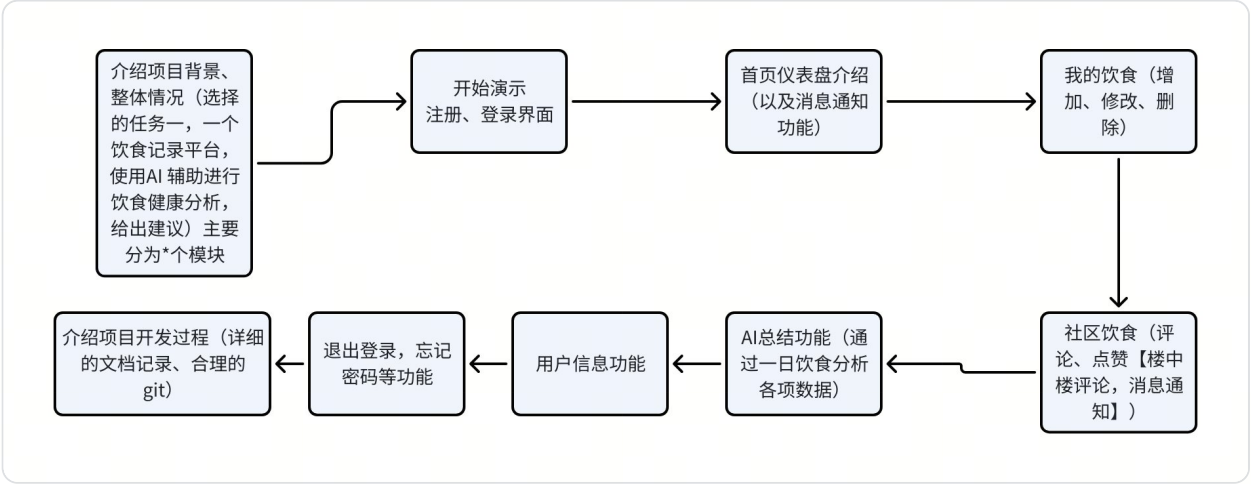
为了**保留合作记录**，我会为每一个任务设置一个**issue**，可以通过**我分配任务**的方式或者通过**认领**的方式 获取任务；同时，我设置仓库不能通过简单的**合并merge**来完成合并代码的操作，只能通过**pr**的方式来合并代码，这个过程中，需要注意，可能需要指定审批代码人（好像有写权限的人都可以，但是最好指定**组长**），同时，在pr描述里添加**Closes #issue\_number**，将issue任务关闭。

目前，**四个页面**的开发工作，在提交pr的时候，一定注意，**不要pr到master分支**，到时我会建立一个**integration**分支，需要手动指定**pr到这个分支**。

七、提交文件

1、演示视频

脚本



字幕：

```
Code block

1  老师好，我们小组选择的是任务一，完成了一个饮食记录平台，并且使用AI辅助进行饮食健康分析，给出建议。主要功能模块包括用户登录注册，以及用户信息的修改与注销，然后是饮食记录及其社区讨论功能，还有就是数据统计和AI分析功能。
2  接下来我将向您进行演示以具体介绍。
3  首先运行项目的后端与前端。
4  在浏览器中进入前端界面。这里我们首先进行注册，填写账号信息。
5  然后输入邮箱，通过邮箱收到的验证码进行绑定。
6  注册完成后进行登录，登陆进去可以看到首页仪表盘，显示了一些统计信息，由于是新注册的用户，所以信息并不丰富。
7  接下来，演示自我饮食记录的功能。
8  首先，添加一条饮食记录。
9  添加完成后，还能够对其进行修改。
10 也可以进行删除，这里我们新建一条记录。
11 删掉这条记录。
12 可以看到饮食记录的增删查改功能都是完成了的。
13 而在社区记录中，可以看到其他用户分享出来的饮食记录。
```

14 这里我们对之前饮食记录进行修改，使其分享到社区中。

15 修改后，可以看到我们的添加的记录了。

16 社区记录中也会显示对于饮食记录发布情况的统计，包括总记录数量、今日记录的数量。

17 此外，在社区中，还可以对其他人的记录进行点赞和评论。

18 也可以回复他人的评论。

19 接下来介绍AI分析助手的功能。

20 当今日的饮食记录有变动后，就会自动调用AI分析，生成今日饮食的数据分析报告，对饮食情况进行评分。

21 当然也可以手动调用分析报告的生成。

22 分析报告中包括营养分析，对碳水、蛋白质、脂肪进行计算和总结。

23 这一界面还可可视化了卡路里和健康指数的变化趋势。

24 分析报告中还包括健康建议、改进建议。

25 我们还实现了AI营养师助手，可以向他询问一些问题。

26 比如说我问他“中午应该吃什么”，他就能给出一些建议。

27 由于我们之前对他人的饮食记录进行了互动，这里我在另一个浏览器提前登陆了对应账号。

28 刷新界面后可以看到多了一些消息提醒。

29 这里的消息提醒就包括刚才的点赞提醒和评论提醒。

30 此外我们还开发了用餐提醒，可以看到由系统发送的午餐用餐消息。

31 最后我们演示对于用户的信息的修改功能，在个人资料界面可以修改用户的信息。

32 也设置了注销账号的功能，实现账号的软删除。

33 退出登录回到登录界面。

34 可以看到忘记密码功能，可以让用户通过绑定的邮箱进行密码重置。

35 在最后，将展示我们开发过程中的一些文档。

36 包括我们使用飞书文档去共同计划、设计需求和数据库，提高协作效率。

37 也进行了代码管理，可以看到一些协作的记录，包括开发分支、提出的一些改进需求和开发中提交的PR。

38 以上就是我们小组项目的全部内容，谢谢老师。

## 2、部署文档

使用README.md文件

## 3、接口文档

swagger无法直接导出文档。可以借用ApiFox工具。

1. 先将Swagger的数据导入到apifox在线网站上

如何从 Swagger 迁移至 Apifox (4 种方法)

<https://apifox.com/blog/how-to-migrate-from-swagger-to-apifox/>  
之前使用 Swagger 来管理 API，如何迁移到 Apifox？  
如何将现有的 Swagger 管理的 API 迁移到 Apifox？4 种方法等你实践！

2. 导出为想要的格式（感觉对于html的适配比较好）



<https://docs.apifox.com/export-data>  
导出数据 - Apifox 帮助文档  
导出数据 - Apifox 帮助文档

## 4、git提交记录（文字+图片，内容放在这个文档里）

核心：尽可能美观，多参考网上方法

文字导出可以参考 [链接](#)，也可以多多找几个方法。图片可以直接截图（示例如下，也可以其他美观视图）



fix: 处理日常饮食分析的空值情况	origin & feat/userinfo	今天 14:58	baiyun
feat(#18): 重构个人资料页面并添加新功能		今天 14:53	baiyun
Merge remote-tracking branch 'origin/integration' into integration		今天 14:10	薛文杰
修复整合后的bug、完善显示效果		今天 14:01	wady*
完善社区页面的餐食图片展示	origin & feat/bug_fix	今天 13:54	baiyun
更新图片文件夹的相对路径		今天 13:06	baiyun
合并		今天 12:26	baiyun
修复了打开编辑记录对话框时无法显示图片的问题。	../diet-record-crud	今天 12:14	xaoica
完善AI分析和今日概况接口调用		今天 12:12	baiyun
Merge remote-tracking branch 'origin/feature/diet-record-crud' into featu		今天 11:32	xaoica
完善AI请求鉴权		今天 11:23	baiyun

## 七、总结

### 1、git操作

场景一：完成了一次开发（本地都已经commit），想要最终PR到分支 integration 上去

可以先切换至 integration 分支上去，然后拉取最新内容；之后再切换回自己原先的分支，执行 git rebase integration，解决冲突后，然后直接push上自己的远程分支