程序设计实习作业报告

程序名称：PKU FIT

创作目的

在当代大学生活节奏日益加快的背景下，身体健康与心理调适已成为高校学生不可忽视的重要课题。**PKU FIT** 应运而生，旨在为北京大学的学生群体提供一个科学、高效、个性化的健身管理平台，助力同学们在繁重的学业之余，找到身心平衡的出口。

本程序不仅满足基础的健身记录与管理需求，更通过数据驱动、可视化设计和人性化交互，激发学生对运动的持续兴趣，帮助他们制定科学合理的训练计划，进而提升身体素质、增强意志力、塑造健康生活方式。

我们希望通过 **PKU FIT** 营造一个“人人热爱运动、人人参与健身”的校园氛围，为构建健康中国贡献“燕园力量”。

程序功能简介（PKU FIT）

PKU FIT 包含五个功能模块 致力于为北京大学学生提供科学高效的健身支持

1 主界面
用户可以浏览各类健身动作 像添加购物车一样将动作加入已选列表并保存 实现训练项目的灵活组合与管理

2 动作库
收录大量健身动作 每个动作包含难度 训练部位等信息 用户可以按需查找和筛选 用于制定个性化训练计划

3 计划生成
提供两种方式生成训练计划
一是通过 deepseek 智能模型 根据用户选择的训练部位和动作难度自动推荐训练方案
二是使用内置规则算法生成标准化计划 满足不同层次的训练需求

4 历史记录
记录每天的训练内容和锻炼部位 用户可以按日期查看训练记录 并通过图示回顾身体各部位的锻炼情况

5 身体数据
支持录入身高体重 系统自动计算 BMI 并以折线图形式展示身体数据变化趋势 便于用户长期追踪健康状况

项目各模块与类设计细节

---

app.py 是训练记录系统的主程序

主要功能是使用 PyQt5 创建应用主窗口，管理多个功能页面。集成了名为 LiteDataManager 的数据库管理器，负责本地 SQLite 数据库的数据读写。通过 QStackedWidget 控件实现页面切换，提升界面响应速度与结构清晰度。

提供统一接口 show_page(name) 用于跳转指定页面，并支持调用页面的 on_show 钩子函数完成初始化操作。另有 logout 方法用于注销用户并跳转回登录页。

TrainingApp 类继承自 QWidget。包含 stacked_widget（页面堆栈控件）、database_manager（LiteDataManager 实例，连接路径为 ./data/test2.db）、pages（页面名与页面对象的字典）等属性。页面包括 LoginPage、MainPage、ActionLibraryPage、PersonalizedWorkoutPage、BodyDataPage、ViewPage 和 zRecordPage。初始化时创建所有页面并添加到堆栈中，默认显示登录页。

程序入口配置日志输出至 ./data/app.log 和控制台，确保 ./data 目录存在，创建 QApplication 与 TrainingApp 实例，设置窗口标题“训练记录系统”和大小 800x600，启动事件循环。

整体结构清晰、功能模块化，便于维护与扩展。

---

action_library_page.py 模块功能为展示健身动作库并提供动作详情查看与跳转

FontConfig 类提供统一字体样式，包括标题、副标题、按钮、列表项和卡片标题字体。MuscleColorMap 类用于映射肌肉名称与对应颜色，用于训练部位彩色标识。

ColoredCardButton 继承 QPushButton，每个动作以彩色卡片形式展示，上方为彩色肌肉条，下方为动作名称与难度，支持交互反馈。

ActionDetailDialog 继承 QDialog，用于展示单个动作详情，包含高亮目标肌肉、描述和基本信息，使用 QTextEdit 支持多行显示。

ActionLibraryPage 是主页面类，构建整体界面结构，包含标题、设置按钮、肌肉颜色图例、滚动区域与返回按钮。加载 JSON 或数据库数据后，卡片式展示动作，点击可弹出详情框。

---

base.py 定义所有页面的基类 BasePage

继承 QWidget，提供 on_show 方法作为钩子函数，子类可重写用于页面切换时的初始化操作。统一接口有助于管理页面生命周期。

---

body_data_page.py 实现身体数据管理与可视化

BodyDataPage 继承 BasePage，构造函数接收 controller。界面包括日期选择器、时间范围过滤框、体重变化图表和表单输入区域。用户可录入身高体重，自动计算 BMI 并保存到数据库。图表通过 matplotlib 展示体重随时间变化曲线，支持过滤显示“最近 7 天”、“最近 30 天”或“全部数据”。

保存数据时调用 save_data，将 BodyStats 写入数据库。on_show 方法会加载最近数据并更新图表。图表根据日期自动调整刻度，美化样式、字体、网格、图例等。

---

login_page.py 提供用户登录与注册界面

LoginPage 继承 BasePage，使用 QTabWidget 提供登录与注册两个选项卡。登录卡支持用户名与密码验证，注册卡支持创建新用户。界面使用垂直布局与定制样式，提升视觉美感。

支持注册成功提示并跳转至登录页，失败提示报错。使用回车键可快速提交表单。逻辑调用 controller.database_manager 接口完成登录和注册操作。

---

main_page.py 是主界面，负责展示、选择、筛选与保存训练动作

定义 FontConfig 统一字体样式，CardButton 自定义按钮显示动作卡片，包括图片和文字标签。

MainPage 初始化时构建整体界面，顶部设置按钮用于登出。中部左侧为已选动作列表，右侧为动作库展示，支持关键词搜索。底部为四个导航按钮跳转页面。

点击动作卡片将添加至左侧列表。保存按钮可将选中动作存入数据库。搜索框支持模糊匹配并高亮结果。

---

personalized_workout_page.py 实现智能与普通训练计划生成界面

继承 BasePage。界面包括训练时长输入、目标肌肉与难度下拉框、生成按钮与结果显示区域。支持“AI 生成计划”与“普通生成计划”。

加载本地 JSON 动作库，筛选符合条件动作。AI 生成通过构造提示词并调用 OpenAI 接口生成 Markdown 格式计划文本。普通生成则按 5 分钟/动作估算填充动作计划，可重复使用动作。

生成后可一键添加至主界面训练清单。支持心肺与全身训练特殊处理。界面简洁，逻辑清晰。

---

record_page.py 实现训练记录的手动录入

继承 BasePage，界面包括日期选择器、训练动作与时间输入框、保存记录与查看记录按钮。保存记录时校验数据格式一致性，并写入数据库。支持回主页面与跳转历史记录页。

逻辑调用 LiteDataManager 接口保存数据。保存成功后提示并清空输入框。

---

view_page.py 用于展示历史训练记录和高亮肌肉图

包含自定义日历 TrainingCalendar，用透明绿色标记训练强度。
MuscleOverlayWidget 加载正反肌肉图，结合遮罩图高亮训练部位。

ViewPage 继承 BasePage，on_show 方法加载训练记录，统计训练强度并更新日历。点击日期后显示当日训练详情与对应肌肉图。界面含刷新与返回按钮，风格统一，交互友好。

---

data_interface/__init__.py

从 datamanager 模块导出 LiteDataManager、TrainingInfo 与 BodyStats 三个类，便于模块统一调用。

---

data_interface/datamanager.py 提供数据库接口 LiteDataManager

使用 sqlite3 管理数据库，包含用户表、训练记录表与身体数据表，支持用户注册登录、保存获取训练数据与身体数据。

TrainingInfo 与 BodyStats 为数据类，提供字典与 JSON 格式转换方法。登录验证使用 pbkdf2_sha256 加密密码。训练记录自动合并相同动作组数。图表数据支持按日期筛选展示。

---

actions/action_store.py 是控制台动作库管理工具

支持查看、添加、搜索、删除健身动作，动作数据保存在 fitness_library.json 中。通过命令行输入实现交互操作，数据结构为列表，动作信息包括名称、目标肌肉、器材、难度与描述。提供主菜单与操作入口函数。

---

actions/RuleBasedTrainingPlan.py 实现规则生成训练计划

读取本地动作库 JSON 文件，根据用户输入的训练时长、目标肌肉与难度筛选动作。每个动作估算为 5 分钟，按需填充足够数量的动作。

若难度筛选无匹配，则放宽条件；若仍无结果则提示用户。计划生成后在控制台打印动作序号、名称、难度与描述。

适合作为简单训练计划推荐工具使用，数据源由其他模块维护。

---

ui/_init_.py 统一导出页面各模块

小组成员分工情况

谷金忠：提出并完善项目核心思路，独立完成计划生成模块（涵盖普通生成与 AI 生成功能）、身体数据展示模块、动作库模块；主导各模块的大部分界面美化工作，设计并实现主界面的动作贴图展示。同时积极协助其他组员，推动整体项目进度。

李晨宇：在完善项目思路的基础上，搭建并优化了 UI 整体框架，设计并实现登录界面模块、主界面模块和记录模块；协助开发身体数据展示模块。负责合成 assets 文件夹中的人物图与肌肉标红图像，参与界面美化工作。完成 app.py 主程序结构编写，并在项目中期主导进行大幅度调整和修复，保障项目稳定推进。

周传智：协助完善项目构思，负责数据库模块和 `base.py` 模块的设计与实现；在项目初期阶段重构并清晰化代码结构，优化数据表示方式与前端设计逻辑。同时在多个模块开发中提供支持，有效推动团队协作效率。

项目总结与反思

本次项目中，我们学会了如何结合已有工具与资源进行高效编程，对不同目标的实现难度有了更清晰的认知。每位成员都深刻认识到自身在能力上的优势与不足，尤其在图形化编程方面收获颇丰，成功掌握了 Python 图形界面的开发方式。

未来项目中，可进一步拓展功能，例如：实现健身房人数实时爬取等实用模块。此外，我们也认识到目标设定的重要性——在本项目中期出现的大幅重构提醒我们，日后类似任务中应在前期尽量明确目标状态与功能边界，避免返工、提高效率。