Название проекта: Бот для нарешивания заданий

**Автор:** Боков Владислав

**Описание идеи:** существует проблема. При нарешивании заданий на сайтах или в сборниках замечаешь, что содержание заданий повторяется и ответ на это задание ты помнишь. Так же решать задания на сайтах не всегда бывает удобно, поскольку они не оптимизированы для работы с телефона, а также не позволяют следить за прогрессом и не сохраняют прогресс решения заданий.

**Архитектура проекта:**

**Модули и классы:**

1. **UserManager Class:**

Класс для управления пользователями и базой данных:

* register\_user: Регистрация нового пользователя или обновление существующего.
* update\_rating: Изменение рейтинга пользователя после решения заданий.
* fetch\_top\_users: Возвращает топ пользователей по рейтингу.

1. **TaskGenerator Class:**

Класс для генерации заданий по различным темам

1. **TheorySection Class:**

Класс для работы с теоретическими материалами:

* extract\_archive: Распаковка архива с материалами по заданиям и сохранение путей к файлам.
* get\_theory\_by\_task: Возвращает путь к файлу с теоретическим материалом по заданному номеру.

1. **BotHandler Class:**

Главный класс, отвечающий за общение с пользователем и обработку его запросов: Регистрация пользователей, обработка команд, генерация заданий, пересылка сообщений. Поддерживает отправку заданий, проверку ответов, ведение рейтинга, изучение теории и обратную связь с разработчиками.

**База данных(SQLite):**

Используется для хранения информации о пользователе. Содержит таблицу со столбцами:

1. **id:** уникальный идентификатор записи.
2. **telegram\_id:** уникальный идентификатор пользователя в Telegram.
3. **rating:** рейтинг пользователя

**Библиотеки и технологии:**

1. **Aiogram:** (>= 3.0)Основная библиотека для интеграции с Telegram API.
2. **SQLite:** СУБД для хранения данных о пользователях (рейтинга, истории сессий и прочих метрик). Zipfile: Стандартная библиотека Python для работы с архивированными файлами.
3. **re (Regular Expressions):** Используется для обработки регулярных выражений при поиске файлов в архиве и их сортировке.
4. **Os:** Стандартная библиотека Python для работы с операционной системой (создание папок, управление путями).

**Основные функции:**

1. **Начало общения:** (/start) Регистрирует пользователя в системе
2. **Решение заданий:** Пользователь выбирает тип задания, количество вопросов и приступает к решению. За правильные ответы начисляется рейтинг.
3. **Изучение теории:** Пользователь выбирает номер задания и скачивает учебный материал в виде PDF-документа.
4. **Обратная связь с разработчиком:** Пользователь может связаться с разработчиком по какому-либо вопросу
5. **Лидерборд:** Доступ к таблице лидеров по рейтингу, позволяющей увидеть успехи других пользователей.

