Лицей Академии Яндекса  
 Образовательная площадка «Супермалыш»

Проект WEB  
**Телеграм бот для парсинга электронного дневника**

Ибатуллин Тимур

Проверил:  
 Гильдин А. Г.

Уфа – 2025

Задание проекта: написать телеграм бота на основе фреймворка aiogram, предназначенного для парсигда оценок и домашнего задания из электронного дневника.

Бот включает в себя следующие команды:

* /start - перезапуск бота
* /reg - регистрация/авторизация в системе
* /get\_marks - получение текущих оценок
* /get\_diary - получение расписания на выбранный день

Технологии и инструменты:

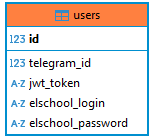
aiogram – асинхронный фреймворк для работы с Telegram Bot API.

aiohttp – асинхронные HTTP-запросы к электронному дневнику.

BeautifulSoup – парсинг HTML-страниц для извлечения данных.

aiosqlite + SQLAlchemy – асинхронная работа с базой данных SQLite и ORM-моделирование.

Данные пользователей храняться в однотабличной базе данных sqlite, имеющей следующую структуру:



Описание назначения полей таблиц базы данных:

Таблица users:

telegram\_id – id пользователя в телеграмме;

jwt\_token - JWT токен пользователя на сайте elschool;

elschool\_login - логин пользователя в электронном дневнике;

elschool\_password - пароль пользователя в электронном дневнике.

Обработка ошибок:

Все критические операции обернуты в try/except с логированием (logging.error).

Пользователь получает понятные сообщения об ошибках (например, "Ошибка регистрации").

Описание механизма сессий:

В проекте используется глобальный словарь sessions для хранения активных сессий пользователей. Это необходимо для эффективного взаимодействия с API электронного дневника.

Преимущества работы с сессиями:

* Избегает повторного ввода логина/пароля при каждом запросе.
* Сохраняет авторизационные куки и JWT-токен.
* HTTP-соединение не пересоздаётся для каждого запроса.

Проблема и её решение: при перезапуске бота сессии теряются (так как хранятся в оперативной памяти). Это решает Middleware, который выполняет проверку перед обработкой каждого сообщения/команды:

1. Проверяет, есть ли активная сессия для пользователя.
2. Если сессии нет, пытается восстановить её из базы данных:

* Загружает логин/пароль пользователя.
* Создаёт новую aiohttp-сессию и авторизуется в elschool.
* Сохраняет сессию в глобальный словарь sessions.