Лицей Академии Яндекса  
 Образовательная площадка «Супермалыш»

Проект WEB-сервер(бот+алиса)

Выполнил:

Гареев А. И.  
 Проверил:  
 Гильдин А. Г.

Уфа – 2025

Технологии и архитектура Telegram-бота для репетиторов и учеников Этот проект представляет собой Telegram-бота на Python, предназначенного для организации учебного процесса между репетиторами и учениками. Бот предоставляет функционал для регистрации пользователей, создания уроков, назначения домашних заданий и проверки работ.

Основные технологии Python 3.7+ (с поддержкой async/await)

python-telegram-bot v20.x — фреймворк для работы с Telegram Bot API

aiosqlite — асинхронная работа с SQLite

nest-asyncio — для корректной работы asyncio в некоторых средах (например, Jupyter)

datetime — работа с датой и временем

Архитектура и функционал Основные сущности Пользователи (users таблица):

telegram\_id, name, role (tutor/student), confirmed (подтвержден ли аккаунт)

Уроки (lessons таблица):

title, datetime, tutor\_id, homework, zoom\_link

Связь учеников и учителей (student\_teacher таблица)

Домашние задания (lesson\_students таблица):

homework\_done, feedback

Ключевые функции бота Регистрация и аутентификация:

Ученики и репетиторы регистрируются через /start.

Учителя могут подтверждать учеников.

Управление уроками:

Создание уроков (/create\_lesson).

Назначение учеников.

Добавление Zoom-ссылок.

Домашние задания:

Учителя добавляют задания (/homework).

Ученики отмечают выполнение (/my\_homework).

Учителя проверяют и оставляют комментарии (/check\_homework).

Интерактивные элементы:

Inline-клавиатуры для выбора ролей, уроков и учеников.

Динамические формы (ввод названия урока, даты, домашнего задания).

Работа с базой данных Используется SQLite с асинхронным доступом через aiosqlite.

Нет ORM — все запросы написаны вручную.

Примеры запросов:

python

Добавление пользователя

await db.execute("INSERT INTO users VALUES (?, ?, ?, ?)", (user\_id, name, role, confirmed))

Получение списка уроков

cursor = await db.execute("SELECT \* FROM lessons WHERE tutor\_id=?", (tutor\_id,)) 4. Особенности реализации Асинхронность:

Все операции (бота, БД) работают через async/await.

Конечные автоматы (FSM):

Состояния регистрации (register\_step).

Процесс создания урока (create\_lesson\_step).

Проверка домашних заданий (grading\_step).

Интеграция с Telegram API:

Обработка команд (CommandHandler).

Callback-кнопки (CallbackQueryHandler).

Динамические ответы (update.message.reply\_text).