Отчёт

Планируемая архитектура не сильно отличается от концептуальной схемы, изложенной в техническом задании.

Фактическое же исполнение немного отличается. LLM модель взаимодействует напрямую с базой данных и свой ответ сохраняет там же, а веб-приложение принимает всё из базы данных, включая и сообщения LLM, а оператор редактирует сообщения и отправляет их обратно в бд, с помощью которой и происходит обратная отправка сообщений в бота.

Стек использованных технологий:

Backend: Python (pyTelegramBotAPI, psycopg2, redis)

Frontend (модерация): Streamlit

База данных: PostgreSQL

Контейнеризация: Docker

Из ИИ инструментов я использовал DeepSeek для поиска информации об ошибках, отладке кода и генерации стилей для streamlit-а, В основном же использовал ollama для генерации кода для ответов на запросы. Модель LLM использовал llama3, т.к. модели ранних версий плохо работают с русским языком и отвечают на других. Промпт я использовал универсальный, который подойдёт для ботов разной направленности.

Инструкция по запуску: После распаковки архива в командной строке нужно дойти до нужной директории корневой папки проекта и вызвать команду “docker compose up –build”. После этого в командной строке появится ссылка Local URL, по которой откроется web-интерфейс. Общение с ботом производится через бота “[t.me/genai\_kirdrey\_bot](http://t.me/genai_kirdrey_bot)” или же сменой токена бота в файле [main.py](http://main.py).

(Установка последней версии llama3 может занять некоторое время, но модель будет значительно умнее)

Всё перечисленное из полноценного комплекса было выполнено в полном объёме.