



EduBot – Документація (Dev + Deploy)

EduBot — це інтелектуальний асистент для університетів, який допомагає студентам, абітурієнтам та працівникам знаходити потрібну інформацію через Telegram-бота та вебінтерфейс.

Ця документація охоплює дві частини: -  **Dev версія** — інструменти для embedding, векторизації та візуалізації знань -  **Deploy версія** — готовий до використання Telegram-базований бот із Gradio аналітикою

Структура проєкту

```
EduBot/ |— main/ # Dev: ноутбуки, скрипти, knowledge-base | |—  
main.ipynb | |— visualize_vectors.py | |— knowledge-base/ | |—  
vector_db/ | |— .env | |— requirements.txt / environment.yml |  
|— README.md |— insurellm-gpt/ # Deploy: Telegram bot + Gradio  
| |— app.py | |— database.py | |— knowledge-base/ | |—  
requirements.txt | |— .env | |— README.md |— chat-log-  
inspector/ # Другий Hugging Face Space (аналітика) | |— app.py |  
|— database.py | |— requirements.txt | |— .env | |— README.md
```

Dev версія (embedding & visualization)

◆ Призначення

- Створення векторного простору на базі знань
- Візуалізація схожості документів через PCA / UMAP

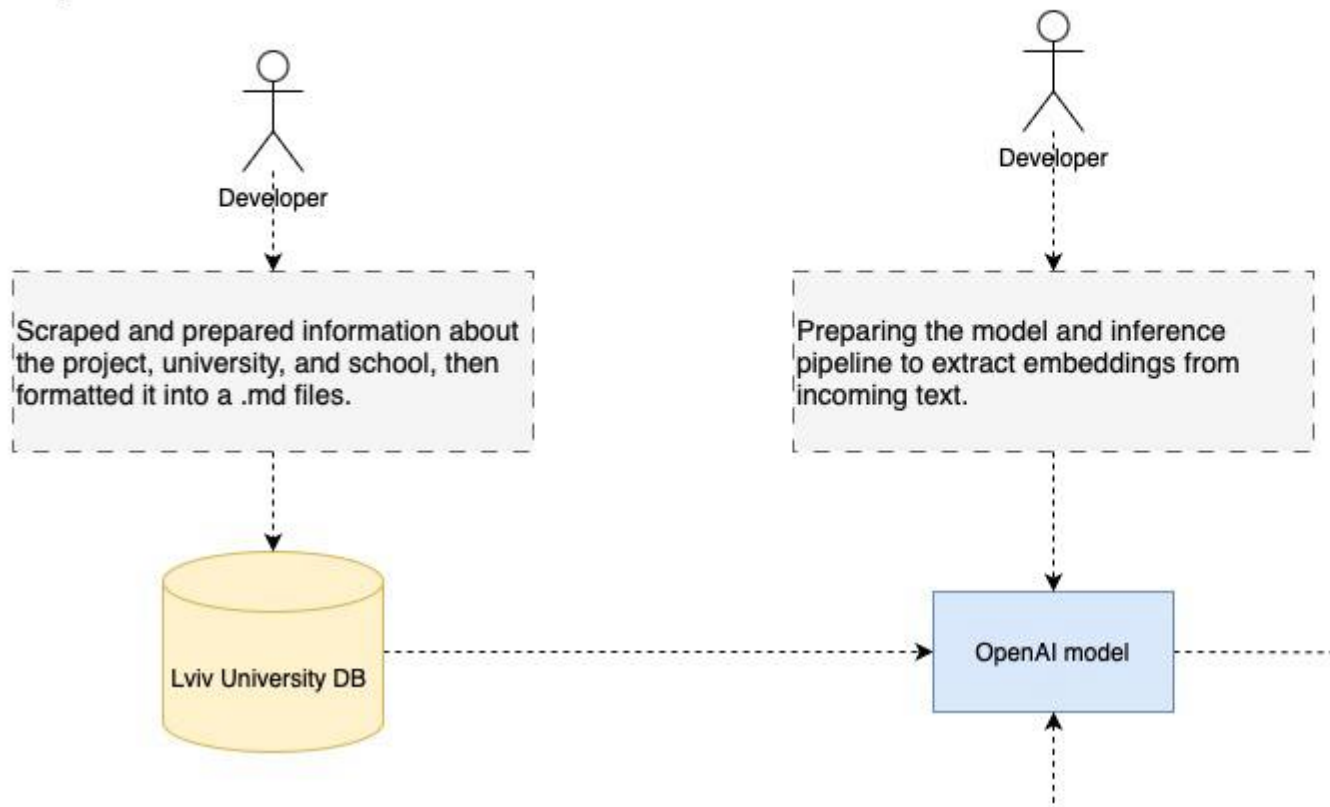
◆ Як працює

Markdown → LangChain → Embedding → Chroma → Візуалізація

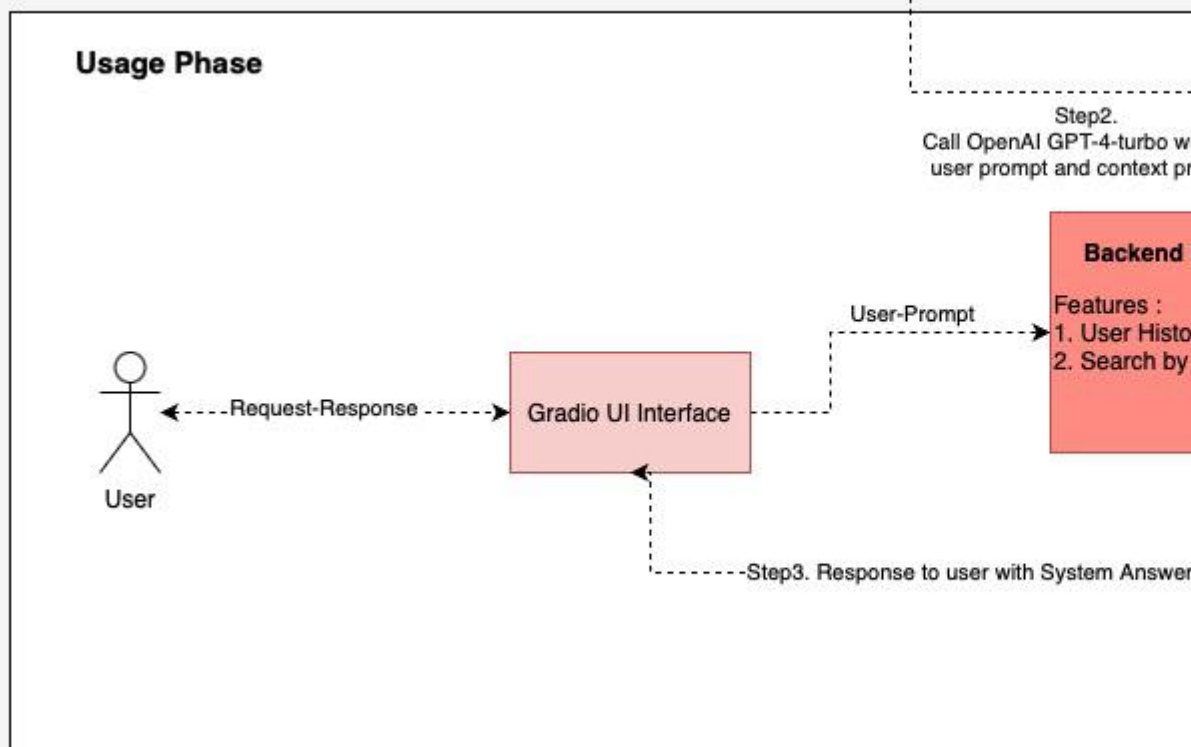
🧠 Загальна логіка Dev-процесу

Main Diagram

Preparation Phase



Usage Phase

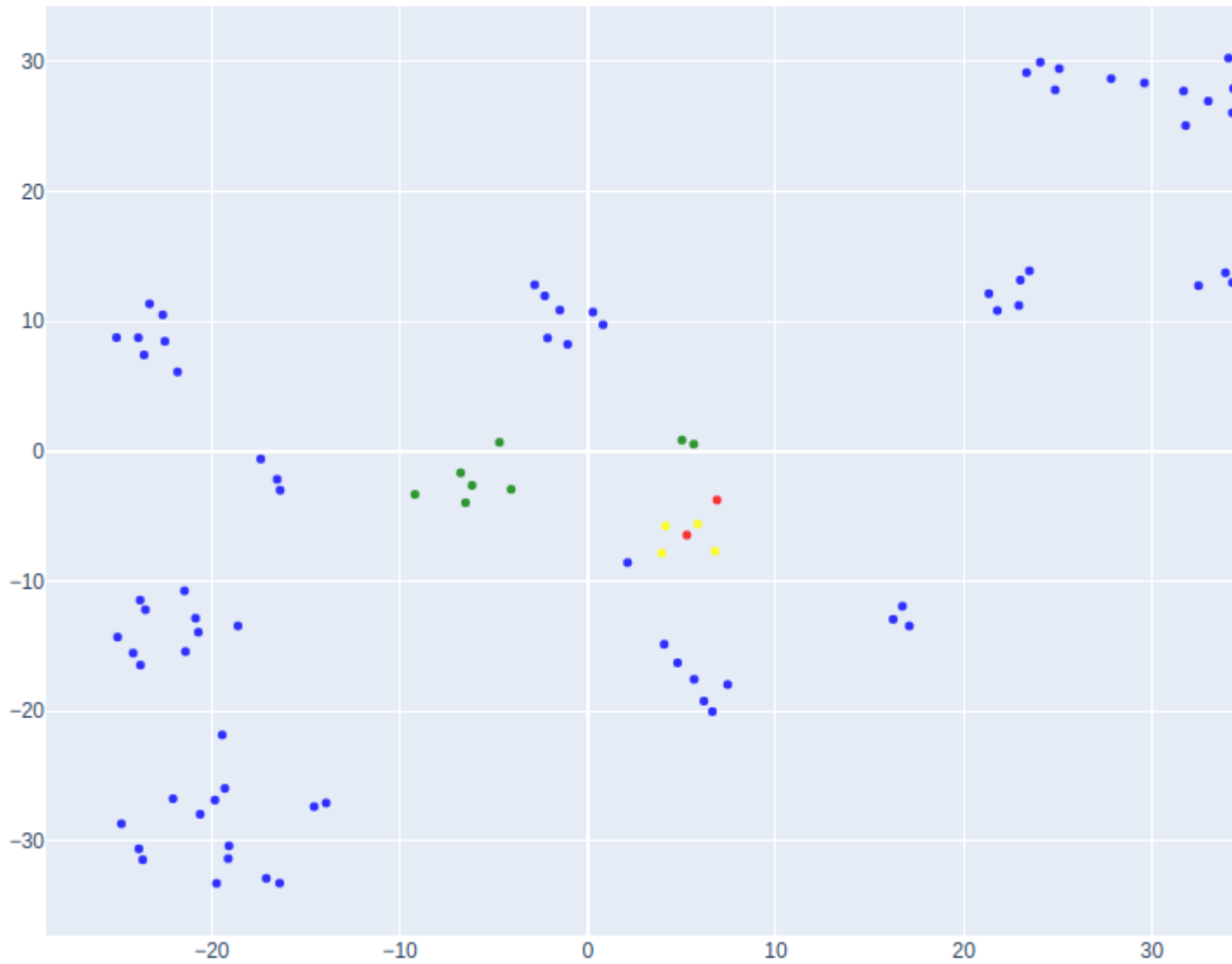


Візуалізація векторів

Візуалізація embedding-даних допомагає інтерпретувати структуру знань.

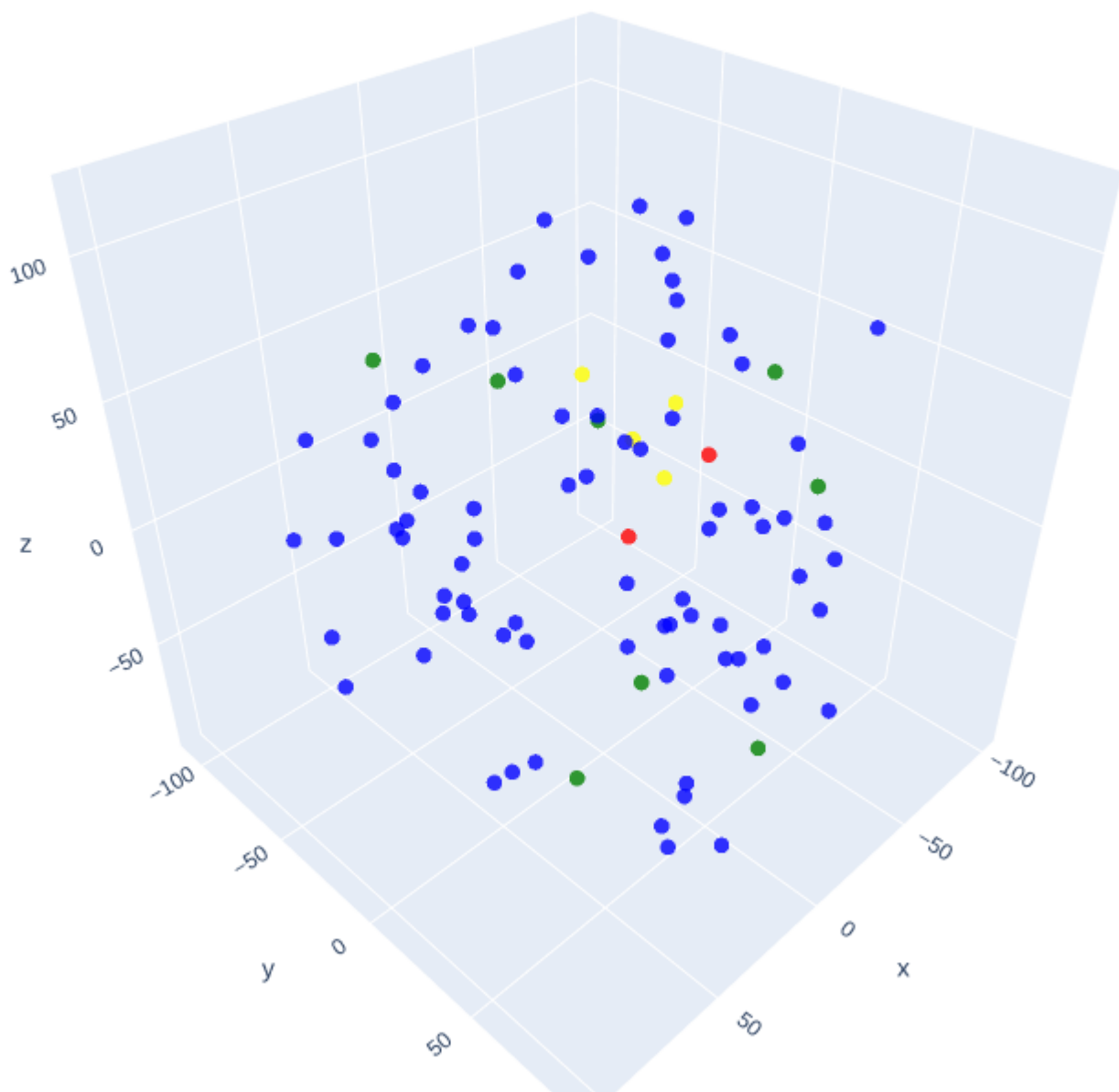
2D-проекція

2D Chroma Vector Store Visualization



📍 3D-проекція

3D Chroma Vector Store Visualization



🚀 Deploy версія (Hugging Face)

◆ Telegram-бот

- Запускається з `app.py`
- Реалізовано через `pyTelegramBotAPI`

- Отримує повідомлення → додає до історії → надсилає у LangChain chain → повертає відповідь
- Обмеження на мову (українська)
- Зберігає повідомлення до `chat_logs.db`

Приклад Telegram-чату

EduBot
bot



/start 6:22 PM ✓✓



Привіт, Matthew! 🙌 Я - ваш бот. Як могу допомогти? 6:22 PM



де 2 гуртожиток 6:22 PM ✓✓



Гуртожиток №2 знаходиться за адресою: вул. Медової Печери, 39а. 6:22 PM



дай розклад мехмату 6:24 PM ✓✓

Я не могу надати розклад мехмату. Але ти можеш переглянути його [тут](<https://new.mmf.lnu.edu.ua/students/schedule>).

Механіко-математичний факультет
Розклад - Механіко-математичний факультет
Розклад занять, заліків, іспитів, перездач та державних екзаменів механіко-математичн...



6:24 PM



тепер економічного 6:24 PM ✓✓

Ви можете переглянути розклад занять економічного факультету за цим посиланням: [Розклад економічного факультету] (<https://econom.lnu.edu.ua/students/schedule/schedule-zanjatja-denna>).

Економічний факультет
Розклад занять денної форми навчання - Економічний факультет
УВАГА!!! Заняття розпочинаються 10 лютого 2025 р. за ЗНАМЕННИКОМ II семестр 2024-2...



6:24 PM



розкажи про свою архітектуру 6:37 PM ✓✓

EduBot використовує архітектуру RAG (Retrieval-Augmented Generation), яка поєднує базу знань ЛНУ із генеративною моделлю. Для обробки запитів і роботи з текстовими даними застосовується LangChain, а для формулювання відповідей природною мовою використовується OpenAI API.

Вся інформація про університет зберігається у вигляді структурованих файлів Markdown, що дозволяє швидко знаходити та формувати відповіді на запитання користувачів.



6:37 PM







Write a message...

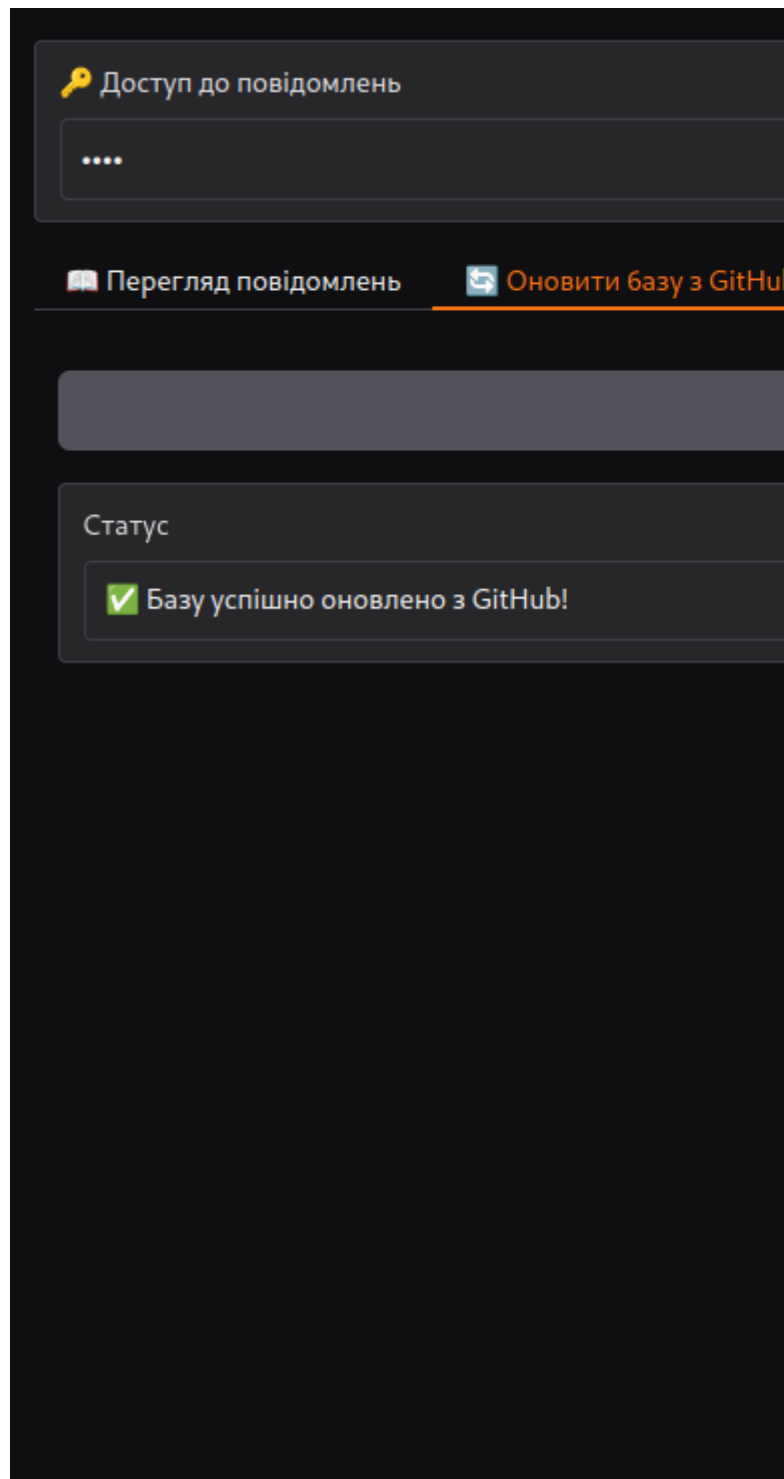
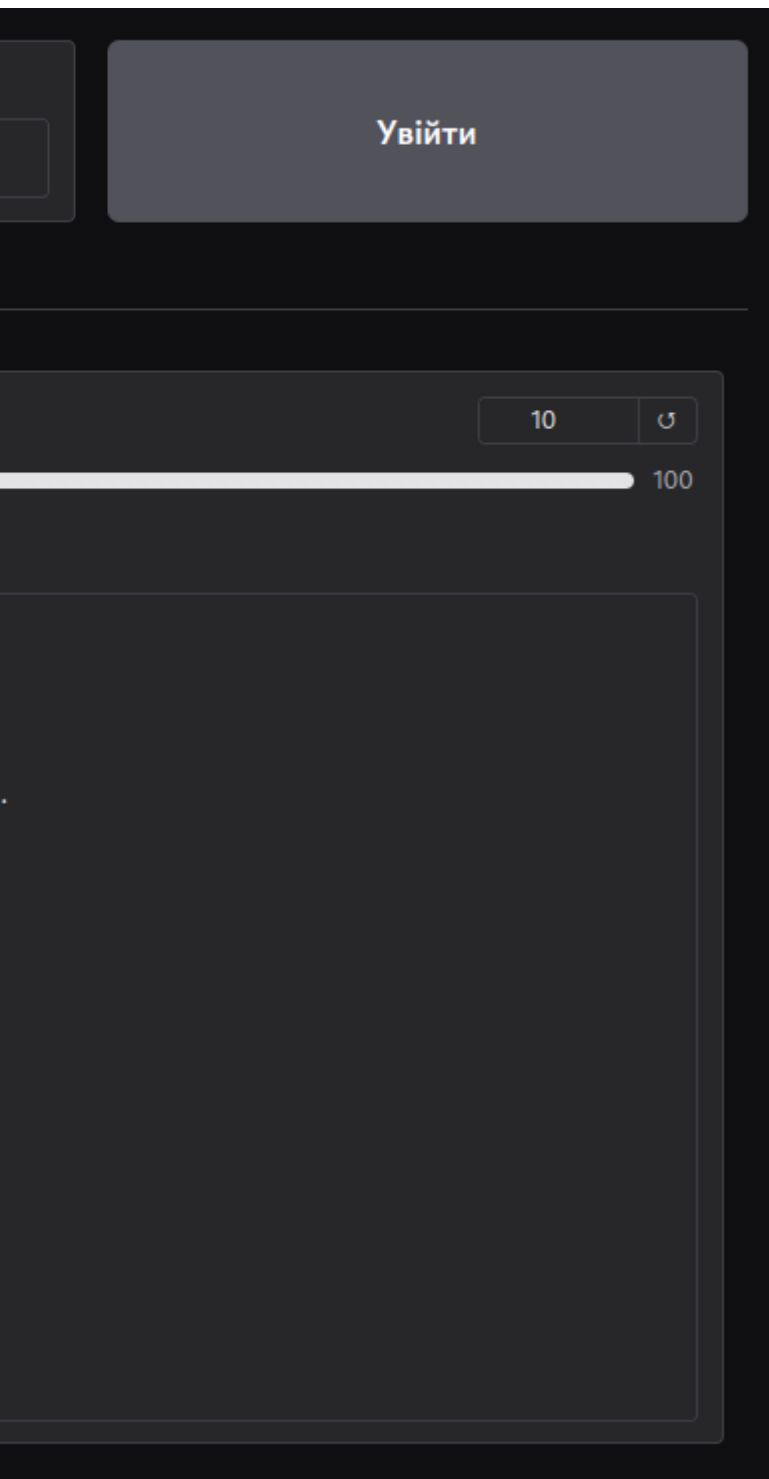
◆ База знань (спільна з Dev)

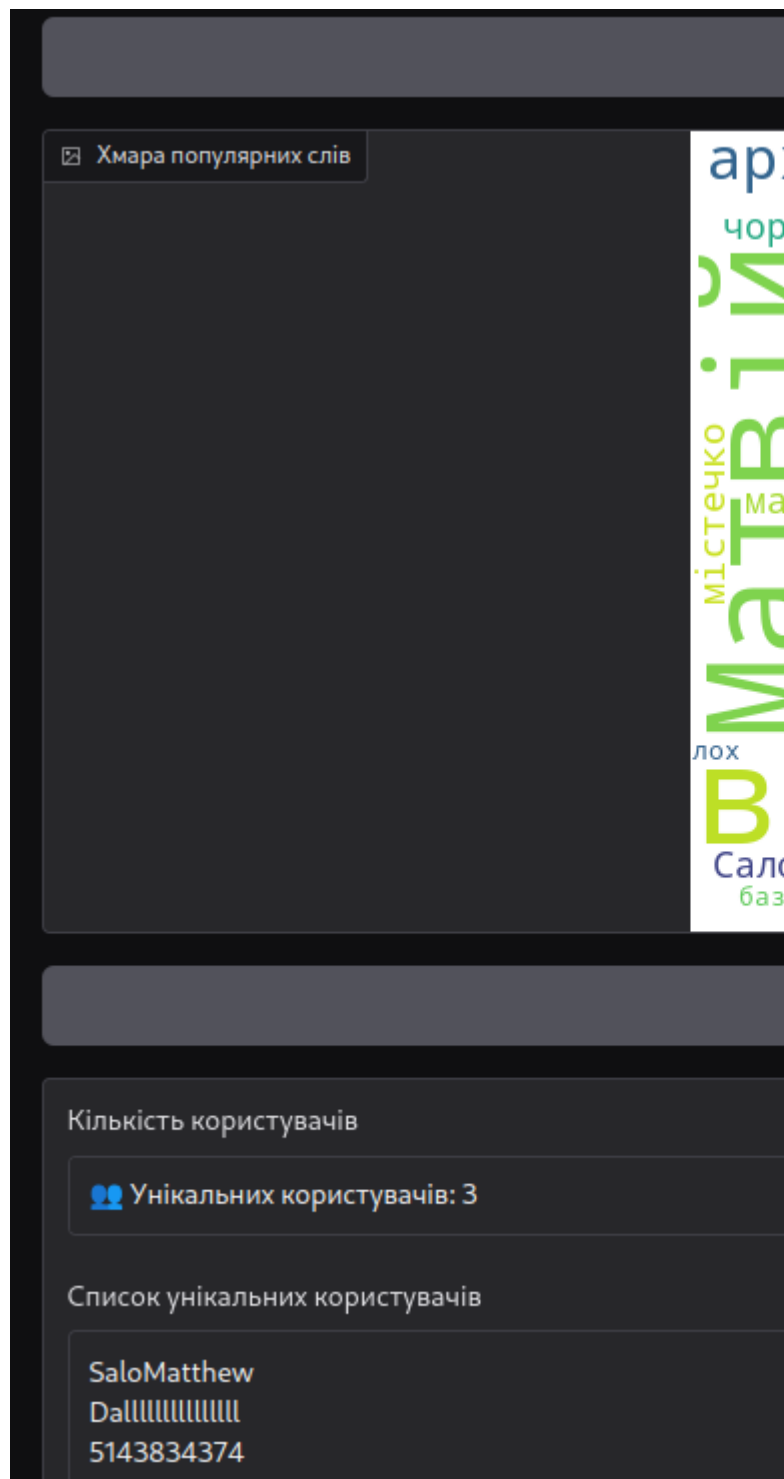
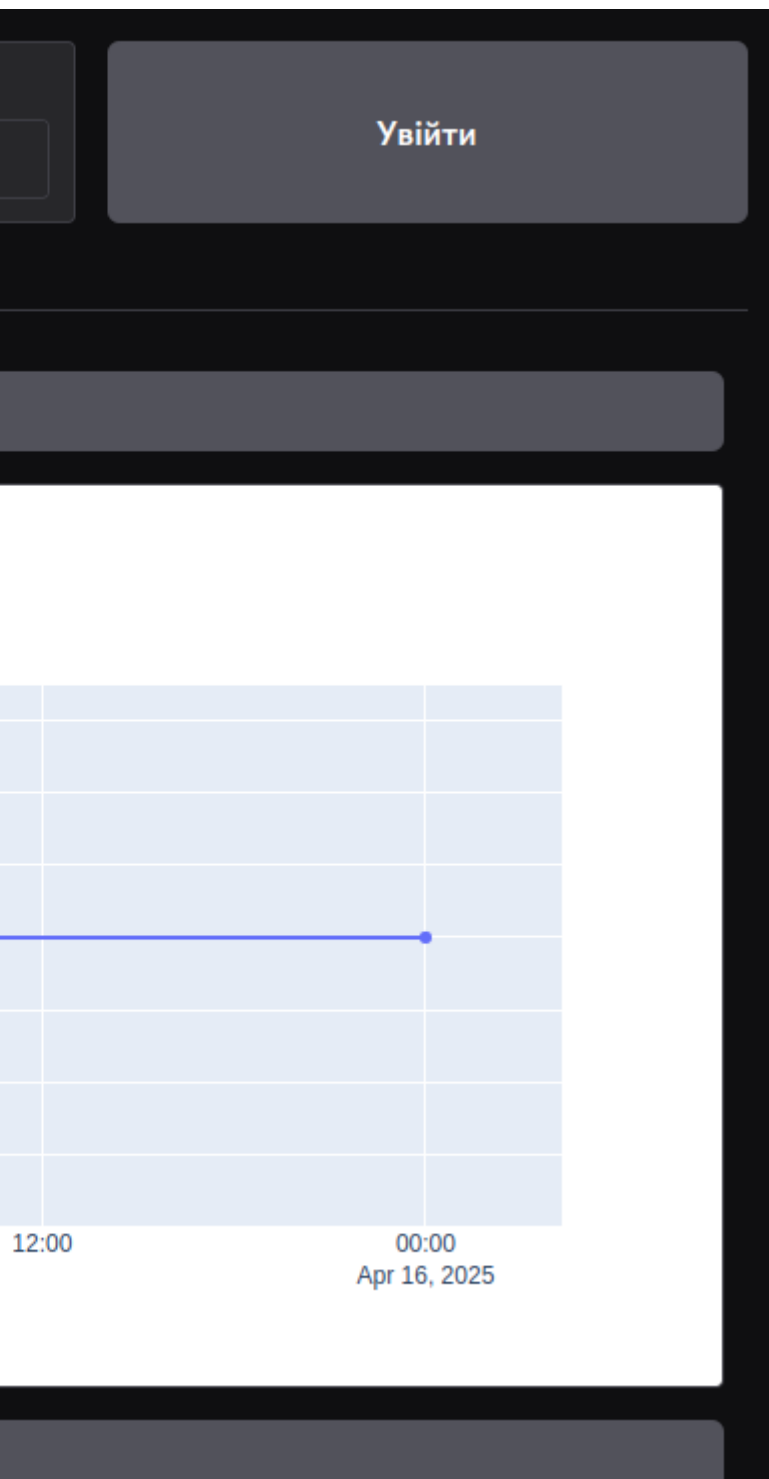
- Ті ж .md файли індексуються через Chroma
- Пошук по embedding здійснюється LangChain chain-ом

◆ Gradio інтерфейс

- app.py генерує Gradio UI з паролем
- Вкладки:
 -  **User Activity**
 -  **Message Charts**
 -  **Referrals**
 -  **Referral Lookup**


Приклади інтерфейсу Gradle






Онлайн версії


Deploy (Hugging Face)

-  **Основна логіка:** [insurellm-gpt](#) — основний модуль, який реалізує пошук у базі знань за допомогою LangChain, зберігає повідомлення,

підтримує реферальну аналітику та забезпечує Gradio-інтерфейс для адміністрування роботи проекту.

-  **Аналітика та огляд історії:** [chat-log-inspector](#) — Gradio-додаток для перегляду збережених повідомлень, аналізу активності, частоти запитів і користувачів.

Dev версія (GitHub)

-  **Розробка та експерименти:** [EduBot repository](#) — репозиторій з embedding-системою, індексацією знань, візуалізацією векторів, прикладами коду та структуруванням бази знань.



База даних

SQLite файл `chat_logs.db` містить:



Таблиці:

- `messages`: `user_id`, `message`, `response`, `timestamp`
- `referrals`: `user_id`, `code`, `timestamp`



Ця база синхронізується з GitHub через `GITHUB_TOKEN`



Архітектура (діаграма)

Нижче представлено загальну схему архітектури системи EduBot, яка показує як Dev-частина (embedding + візуалізація) та Deploy-частина (Telegram-бот та Gradio UI) взаємодіють через спільну векторну базу знань (Chroma):

Markdown Files ↓ LangChain Loaders ↓ Embedding → Chroma ↓ ↘
Gradio UI Telegram Bot

EduBot Architecture



Розгортання Hugging Face

1. Створи Hugging Face Space з типом “Gradio”
2. Завантаж:
 - `app.py`, `database.py`
 - `chat_logs.db`, `.env`, `knowledge-base/`
 - `requirements.txt`

3. Налаштуй Secret variables (API ключі)

4. Запусти Space



Запуск Dev локально

```
bash git clone https://github.com/Matviy-commands/EduBot.git cd  
EduBot pip install -r requirements.txt python main/  
visualize_vectors.py
```



Контакти

Проект створено командою Forwards. Зв'язатися: +380 68 821 20 61