**Grok Insights Bot**

**Overview**

**Problem**

Marketers, startups, and researchers struggle to quickly analyze social media trends on platforms like X, requiring hours of manual data collection and analysis.

**Solution**

A Telegram bot that integrates multiple AI-powered APIs (xAI, OpenAI, Hugging Face, uClassify, TextRazor, MonkeyLearn, Watson) to analyze X posts by topic, delivering concise reports on sentiment, keywords, and entities.

**Impact**

Provides real-time, actionable insights for marketing campaigns, market research, and trend analysis, saving significant time and effort.

**About the Project**

Grok Insights Bot is a Telegram bot that leverages seven AI APIs to analyze X posts and generate detailed reports. Users can select a service, analyze topics, view history, and share results, making it a powerful tool for marketers, bloggers, and data analysts. This project demonstrates my expertise in AI-powered bot development, API integration, database management, and modular code architecture.

**Features**

* **Multi-API Text Analysis**: Analyze X posts using seven services (xAI, OpenAI, Hugging Face, uClassify, TextRazor, MonkeyLearn, Watson) for sentiment, keywords, and entities.
* **Interactive UI**: Inline buttons for repeating analysis, sharing results, and viewing history.
* **Database**: SQLite stores analysis history for each user.
* **Localization**: Supports Russian and English languages.
* **Modular Design**: Separate service modules for easy extensibility.

**How to Work with the Bot**

**Start the Bot in Telegram**

Find the bot by its name (e.g., @grok\_insights\_bot) or start a chat with it.

**Main Commands**

* /start — Launch the bot and receive a welcome message.
* /select\_service — Choose an analysis service (xAI, OpenAI, Hugging Face, uClassify, TextRazor, MonkeyLearn, Watson).
* /analyze <topic> — Analyze posts by topic (e.g., /analyze AI).
* /history — View analysis history.
* /help — Display the list of commands.

**Example Usage**

* Send /start.
* Send /select\_service and choose TextRazor.
* Send /analyze AI to receive a report:

Report for topic 'AI' (TextRazor):

Sentiment: Positive

Entities: Artificial intelligence, Machine learning

* Use inline buttons ("Repeat", "Share", "History") for further actions.

**Notes**

* **xAI and OpenAI**: Require active credits for full functionality.
* **Hugging Face, uClassify, TextRazor**: Operate in free tiers (~500-1000 requests/day).
* **MonkeyLearn and Watson**: Included as examples but currently inactive (require corporate accounts or credits).
* **Error Handling**: APIs may return "insufficient credits" or similar errors, confirming successful integration.
* Analysis results are stored in insights.db for history tracking.

**API Keys**

API keys can be obtained from:

* xAI: [x.ai/api](https://x.ai/api)
* OpenAI: [platform.openai.com](https://platform.openai.com/)
* Hugging Face: [huggingface.co](https://huggingface.co/)
* uClassify: [uclassify.com](https://uclassify.com/)
* TextRazor: [textrazor.com](https://textrazor.com/)
* MonkeyLearn: [monkeylearn.com](https://monkeylearn.com/) (via Medallia)
* Watson: [cloud.ibm.com](https://cloud.ibm.com/)

**Project Structure**

* main.py — Main bot executable.
* config.py — Configuration for API keys and services.
* utils.py — Utilities for API interactions and SQLite database management.
* handlers.py — Command and button handlers.
* create\_db.py — SQLite database initialization.
* services/ — API-specific modules:
  + xai.py, openai.py, huggingface.py, uclassify.py, textrazor.py, monkeylearn.py, watson.py — Active services.

**Technologies**

* **Python 3.8**: Core language.
* **aiogram 2.21**: Telegram Bot API framework.
* **xAI, OpenAI, Hugging Face, uClassify, TextRazor, MonkeyLearn, Watson APIs**: Text analysis.
* **SQLite**: Data storage.
* **aiohttp**: Asynchronous API requests.
* **Git**: Version control.

**Author**

Rostislav — AI agent and full-stack developer.  
This project is part of my portfolio, showcasing expertise in AI-powered bot development, multi-API integration, and modular software design.

**License**

MIT License