Техническое задание

# Проект

Разработка Python-приложения для взаимодействия с локально развернутыми LLM

# Цель

Предоставить пользователю локальное решение для работы с LLM-моделями через LM Studio и Ollama, поддерживающее мультимодальность, мультидиалоги, удаление чатов, эмбеддинги и доступ к интернет-источникам при необходимости.

# 1. Установка и настройка среды

1.1. Требуется установить:

* - LM Studio — UI для локального взаимодействия с LLM
* - Ollama — CLI-интерфейс для запуска моделей на локальной машине

1.2. Требуется:

* - Разобраться в механизмах API или WebUI для взаимодействия с этими программами
* - Обеспечить доступ к локальным LLM по локальной сети

# 2. Развёртывание моделей LLM

2.1. Запустить несколько моделей, совместимых с ресурсами ПК (например, 7B или 13B моделей на базе Mistral, LLaMA, Phi, Gemma и т.д.)

2.2. Обеспечить доступность моделей через API (HTTP/REST/WebSocket или другой механизм), для интеграции с Python-приложением.

# 3. Разработка Python-приложения

Создать отдельную программу на Python со следующим функционалом:

3.1. Многодиалоговость

* - Возможность параллельного ведения нескольких независимых чатов
* - Для каждого чата должен вестись отдельный контекст

3.2. Удаление чатов

* - Возможность удалить любой диалог по ID или имени
* - После удаления весь контекст чата должен быть удалён из хранилища

3.3. Работа с файлами и изображениями

* - Поддержка отправки пользователем изображений и файлов
* - Обработка содержимого файлов (OCR, извлечение текста) и ответ LLM на их основе
* - Возможность отправки файлов в ответе LLM

3.4. Доступ к интернету

* - Пользователь может включить режим, при котором LLM будет использовать интернет для формирования ответа
* - При наличии флага (use\_web=true) реализовать:
* - обращение к поисковикам или сайтам
* - включение найденной информации в контекст LLM

3.5. Эмбеддинги (векторизация текстов)

* - Реализовать механизм векторизации пользовательских текстов
* - Сохранение векторов в базу
* - Поиск по базе и использование результатов как дополнительного контекста

# 4. Ожидаемый результат

1. Рабочая локальная система взаимодействия с несколькими LLM

2. Python-программа с вышеописанными функциями

3. README-файл с инструкциями по установке и использованию