Лицей

Индивидуальная выпускная работа

**Приложение для сохранения кратких записей**

**После онлайн встреч BriefTalk AI**

*Выполнил Кром Илья Васильевич*

Москва 2025

Проект представляет из себя приложение, которое делится на две части: backend и десктопное приложение. Backend представляет из себя API для регистрации пользователей и взаимодействия с моделями искусственного интеллекта. Десктопное приложение в свою очередь является удобным интерфейсом для взаимодействия с backend’ом. Также десктопное приложение позволяет вести аудио запись онлайн встреч на данном устройстве.

Основная идея, собственно – записать онлайн встречу, отправить её на backend, там она преобразуется в текст, после чего из него будут выделены основные тезисы. Таким образом после онлайн встречи у пользователя останется краткая сводка о том, что обсуждалось на встрече.

Современные деловые встречи и звонки часто занимают 2-3 часа, и при их большом количестве участникам становится трудно запомнить и структурировать полученную информацию.

На данный момент на рынке практически отсутствуют решения, которые бы эффективно записывали аудио, автоматически преобразовывали его в текст, суммировали основные темы обсуждения и сохраняли результат в удобных форматах. А то, что есть, в очень часто не поддерживает работу с русским языком, или ограниченно одним конкретным приложением для встреч (например, Zoom)

Заявляемый продукт решает эту проблему, предоставляя пользователям возможность получения как полного текста разговоров, так и краткого пересказа ключевых моментов каждой встречи на русском языке, а также возможность вести аудио запись встречи вне зависимости от «места» её проведения (будь то Google Meet или Яндекс Телемост или еще что-нибудь)

В качестве доказательства проблемного поля прикрепляю результаты проведённого мною опроса:

Целевая аудитория:

Мой продукт нацелен в первую очередь на людей, которые часто работают за компьютером, возможно из дома, и регулярно присутствуют на рабочих «созвонах».

Стек технологий:

* Python – основной язык разработки
* Qt (PySide6 в частности) – основной фреймворк для создания графического интерфейса и дизайна
* PyTorch – фреймворк для работы с ML моделями
* SQLite – СУБД для хранения настроек приложения и хранения данных на удалённом сервере
* FastAPI – фреймворк для backend’а
* SQLAlchemy – библиотека для работы с базой данных
* Docker и docker compose – для сбора backend части приложения в контейнеры

Рефлексия

1. Возникавшие в процессе работы над продуктом проблемы и способы решения: Все проблемы которые у меня возникали делятся на два типа:

А) «Баги» решение которых я нашел в интернете либо спросил у кого-то как их исправить. Таких было очень много и всяких разных, это, я считаю, особенность данной области

Б) «Баги» и проблемы для которых я не смог найти решения ни в интернете, ни у других людей. Пример такого «бага»:

Когда я запускаю моё десктопное приложение, некоторые классы объектов должны обязательно находиться в одном файле кода чтобы всё работало корректно. Это проблема, потому что этот один конкретный файл кода становится слишком большим и неудобным для дальнейшей разработки. Однако объяснения почему так происходит я так и не нашел. Попытки решить проблему путём экспериментирования также ни к чему не привели. Такого сорта проблемы встречаются сильно реже.

1. Дальнейшие перспективы развития
   1. Добавление новых «фичей» в десктопное приложение для более удобного и интуитивно понятного взаимодействия с интерфейсом
   2. Обезопасить backend приложение от всевозможных информационных атак насколько это возможно
   3. Распространить приложение среди пользователей
   4. Возможно, добавление платных функций и\или рекламы в приложение дабы оно начало приносить доход.
   5. Выступление с данным проектом на различных олимпиадах и конкурсах
2. За время выполнения проекта я научился:
   1. Работать с БД через ORM
   2. Оборачивать приложения в контейнеры с помощью Docker
   3. Настраивать подключения по SSH, а также настраивать сервера для безопасной работы с ними
   4. Разворачивать свои приложения на удалённых серверах
   5. Запускать процесс CI\CD
   6. Структурировать большое количество файлов в одну большую систему

В дальнейшем все эти навыки и знания пригодятся мне для написания собственных продуктов, как на основе этого проекта, так и с другими задумками.

1. Десктопное приложения не стоило писать на ЯП Python. У него есть плюсы, среди них относительно быстрая разработка, но возможностей для более тонкой настройки своего приложения в нём (и, в частности, во фреймворке PySide6) нет

Больше мне добавить нечего