**Техническое задание к договору №**

Настоящее Техническое задание (далее - **ТЗ**) заключено между ООО “Девиар Эй Ай”, далее именуемый Исполнителем, и \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, далее именуемое Заказчиком, по последней дате подписания, указанной в подписных разделах ниже.

За исключением случаев, предусмотренных в настоящем документе, настоящее Техническое задание регулируется условиями, содержащимися в Договоре. В случае противоречия между настоящим Техническим заданием и Договором преимущественную силу имеют условия настоящего Технического задания. Термины, написанные с заглавной буквы и неопределенные в настоящем Техническом задании, имеют значения, указанные в Договоре.

**2. Срок действия ТЗ**

Независимо от даты подписания, срок действия настоящего ТЗ начинается с даты вступления ТЗ в силу и продолжается до даты окончания действия ТЗ, если только оно не будет прекращено досрочно в соответствии с условиями Договора (далее -**Срок действия ТЗ**).

**3. Объем услуг**

|  |  |
| --- | --- |
| 3.1 Заказчик поручил ООО “Девиар Эй Ай” выполнить следующие работы: | - создание и настройка программного обеспечения с использованием искусственного интеллекта - автономного ИИ-агента с использованием технологии векторного хранения данных (далее – Агент). |
|  |
| 3.2. Место размещения Агента - в сети Интернет (далее – Сайт) и пользовательский интерфейс: | - взаимодействие через Telegram (Bot API).  - поддержка текстовых запросов на естественном языке. |
| 3.3 Список требований к функционалу Агента: | - автономная работа Агента в телеграм-боте Заказчика для поддержания диалога с пользователями телеграм-бота; |
| - возможность со стороны Заказчика добавлять и извлекать данные, загруженные в векторное хранилище Агента; |
| - возможность со стороны Заказчика видеть загруженные в векторное хранилище Агента документы; |
| - возможность предоставления пользователю ссылок на загруженные в векторное хранилище документы; |
| - возможность работы с изображениями в загружаемых файлах, а именно возможность использования изображений Агентом в диалоге с пользователем телеграм-бота; |
| - автоматическая векторизация и сохранение документа; |
| - ограничение прав: загружать и удалять документы может только клиент, конечные пользователи взаимодействуют только через чат. |
| - возможность сбора, хранения и использования Агентом данных о запросах пользователя и использование информации из них для дальнейшей работы; |
| - возможность хранения загружаемых Заказчиком для работы Агента данных в векторном формате. |
| 3.4 Форматы загружаемых данных: | pdf, txt, doc, markdown суммарно до 250 мб. |
| 3.5 Способ встраивания Агента: | - телеграм-бот, предоставляемый ООО “Девиар Эй Ай”. |

**4. Порядок выполнения работ**

4.1. Разработка проводится в 3 этапа, каждый из которых подразумевает получение обратной связи от Заказчика и внесение Заказчиком соответствующих правок, если это необходимо.

|  |  |
| --- | --- |
| 4.2 Ожидаемые результаты на дату окончания первого этапа: | - автономная работа Агента для поддержания диалога с пользователями; |
| - возможность хранения загружаемых для работы Агента данных в векторном формате. |
| - на данном этапе работа Агента осуществляется в тестовом формате |

|  |  |
| --- | --- |
| 4.3 Ожидаемые результаты на дату окончания второго этапа: | - внесенные с учетом обратной связи в результате тестирования корректировки в работу Агента; |
| - выдача Агентом ссылок на, использованные при ответе источником (загруженные документы); |
| - возможность со стороны Заказчика добавлять и извлекать данные, загруженные в векторное хранилище Агента; |
| - возможность Агента выдавать картинки относящиеся к контексту вопроса |
| - на данном этапе работа Агента осуществляется в тестовом формате |

|  |  |
| --- | --- |
| 4.4 Порядок проведения и ожидаемые результаты на дату окончания третьего этапа: | - тестирование; |
| - внесение окончательных корректировок; |
| - согласование работы Агента; |
| - на данном этапе код и прав доступа Агента передается Заказчику |

4.5 Техническая архитектура и технологический стек.

|  |  |
| --- | --- |
| 4.5.1 Компоненты | - telegram-bot интерфейс взаимодействия с пользователем. |
| - база данных и векторное хранилище PostgreSQL, elasticsearch; |
| - cистема обработки документов – модуль загрузки, векторизации и категоризации; |
| - LM API – GPT-4 (с возможностью замены на альтернативы). |

|  |  |
| --- | --- |
| 4.5.2. Технологический стек | - back-end Python (FastAPI).; |
| - RAG и эмбединг -langchain и HuggingFace/ OpenAI; |
| - база данных и векторное хранилище PostgreSQL, elasticsearch; |
| - взаимодействие с пользователем telegram bot api |
| - деплой и развертывание Docker |
| - Kubernetes оркестрация в случае разрастания сервиса |

4.6. Алгоритм обработки документов.

|  |  |
| --- | --- |
| 4.6.1 Предварительная обработка, векторизация и индексация | 1. Заказчик отправляет файл боту (форматы: .txt, .docx). Бот проверяет:  - авторизован ли пользователь (по Telegram ID).  - допустимый ли формат файла.  При соответствии критериям бот передаёт файл в бэкенд через API. |
| 2. Бэкенд сохраняет оригинальный файл в хранилище |
| 3. Документ конвертируется в текст |
| 4. Проверяется кодировка и структура текста. |
| 5. Текст разбивается на фрагменты (чанки), для каждого чанка формируются метаданные |
| 6. Каждый чанк преобразуется в эмбеддинг. полученные векторы и текст отправляются в Elasticsearch. |
| 7. Создаётся индекс с поддержкой векторного поиска. Elasticsearch автоматически индексирует как по ключевым словам, так и по вектору. |

|  |  |
| --- | --- |
| 4.6.2 Поиск и генерация ответа | 1. Пользователь задаёт вопрос боту |
| 2. Бэкенд векторизует вопрос, отправляет запрос в Elasticsearch, где происходит поиск по ключевым словам. Результаты комбинируются. |
| 3. В топ-3 чанков выбираются лучшие фрагменты. |
| 4. Формируется промпт для LLM, состоящий из вопроса пользователя и извлечённых фрагментов. |
| 5. LLM формирует ответ. |
| 6. Ответ отдается пользователю в Telegram |

|  |  |
| --- | --- |
| 4.6.3 Удаление документа | 1. Пользователь вводит /delete <имя файла>. |
| 2. Бэкенд ищет документ в Elasticsearch. |
| 3. Бэкенд удаляет все связанные чанки из индекса. |
| 4. Бот сообщает: «Документ X удалён». |

|  |  |
| --- | --- |
| 4.6.4 Просмотр списка документов | 1. Пользователь вводит /list. |
| 2. Бэкенд делает запрос в Elasticsearch. |
| 3. Бэкенд возвращает список загруженных файлов. |