Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

Работа выполнена в СКБ «Промышленная робототехника и электроника»

СОГЛАСОВАНО

Руководитель ОНиПКРС

_____ Е.М. Димитриади
(подпись)
«19» 06 2026 г.

Декан ФЭУ

_____ А.С. Гудим
(подпись)
«19» 06 2026 г.

УТВЕРЖДАЮ

И.о. проректора по научной работе, д-р техн. наук, профессор

_____ А.В. Космынин
(подпись)
«19» 06 2026 г.

«Интеллектуальный справочно-тренировочный комплекс для школьников»

Комплект конструкторской документации

Руководитель СКБ	18.06.2026 *(подпись, дата)*	С.И. Сухоруков
Руководитель проекта	18.06.2026 *(подпись, дата)*	Д.М. Грабарь
Наставник проекта	18.06.2026 *(подпись, дата)*	Е.Ю. Щербакова

Комсомольск-на-Амуре 2026

Карточка проекта

Название	Интеллектуальный справочно-тренировочный комплекс для школьников
Тип проекта	Тип проекта: техническое творчество (инициативный), в рамках научно-исследовательского и инновационного конкурса.
Исполнители	Школьник _____ А.М. Грабарь – МОУ СОШ № 23 Школьник _____ И.А. Черкашин – СОО-Лицей ФГБОУ ВО "КнАГУ"
Срок реализации	2 семестра

Использованные материалы и компоненты

Наименование	Количество, шт.
Rasberi Pi 5	1
5-дюймовый сенсорный экран Raspberry Pi HDMI	1
Кабели USB	1
Кабели HDMI	1

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

ЗАДАНИЕ
на разработку

Название проекта: Интеллектуальный справочно-тренировочный комплекс для школьников

Назначение: Предназначен для помощи школьникам в изучении и повторении учебного материала

Область использования: Основное общее образование, среднее общее образование

Функциональное описание проекта: Система, обеспечивающая быстрый поиск информации на основе загруженных материалов, отвечает на заданные вопросы и указывает на разделы, где информацию можно найти. Основана на технологии RAG, генерация ответа пользователю с учетом дополнительной информации, что позволяет языковой модели формировать ответ по достоверным источникам.

Техническое описание устройства: Устройство выполнено на микрокомпьютер Rasberi Pi 5 с подключенным дюймовым сенсорным экраном через HDMI и USB. На встроенном накопителе Rasberi Pi 5 размещена операционная система, векторная база данных, Qrant, подключенные языковые модели, а также программное приложение на Python с фреймворком Streamlit, реализующее пользовательский интерфейс и RAG-конвейер.

Требования: Питание от сети постоянного тока напряжением 5 В через порт USB-C (блок питания мощностью не менее 25 Вт). Внутренние шины питания: 5 В для микрокомпьютера и сенсорного экрана. Наличие визуальной индикации режимов работы с помощью встроенного светодиода Raspberry Pi 5 (постоянное свечение — питание включено, мигание — выполнение обра-

ботки запроса или индексации файлов). Взаимодействие между модулями, интерфейс Streamlit, Embedding model, Qdrant, Reranker, LLM, осуществляется через внутренние вызовы Python-приложения. Передача данных между экраном и микрокомпьютером по HDMI и USB.

План работ:

Наименование работ	Срок
Анализ предметных областей школьной программы, сбор учебных материалов	октябрь 2025
Формирование требований к платформе, постановка технического задания	октябрь 2025
Выбор стека технологий, предварительная оценка подходящих LLM	октябрь – ноября 2025
Проектирование архитектуры базы знаний (структура, метаданные,онтология и т.д.)	ноябрь – декабрь 2025
Проектирование пользовательского интерфейса, создание прототипа	ноябрь – декабрь 2025
Проектирование RAG-конвейера и механизма поиска	январь 2026
Реализация базы знаний, индексирование данных	январь – февраль 2026
Интеграция LLM и настройка RAG	февраль 2026
Разработка модуля подбора тренировочных заданий	март 2026
Тестирование функциональных модулей, отладка	апрель 2026
Финальная интеграция модулей	апрель – май 2026
Подготовка итоговой документации и презентации проекта	май 2026
Проведение демонстрации и защита проекта	май 2026

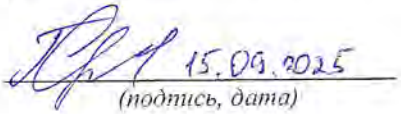
Комментарии:

Перечень графического материала:

1. Принципиальная схема;

2. Внешний вид изделия;

4. Блок-схема алгоритмов (при наличии управляющих программ);

Руководитель проекта  Д.М. Грабарь

(подпись, дата)

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

ПАСПОРТ
«Интеллектуальный справочно-тренировочный комплек для школьников»

Руководитель проекта 18.06.2026 Д.М. Грабарь
(подпись, дата)

Наставник проекта 18.06.2026 Е.Ю. Щербакова
(подпись, дата)

Комсомольск-на-Амуре 2026

Содержание

1 Общие положения 7
1.1 Наименование изделия 7
1.2 Наименования документов, на основании которых ведется проектирование изделия 7
1.3 Перечень организаций, участвующих в разработке изделия 7
1.4 Сведения об использованных при проектировании нормативно-технических документах 8
2 Назначение и принцип действия 9
2.1 Назначение изделия 9
2.2 Области использования изделия 9
2.3 Принцип действия изделия 9
3 Состав изделия и комплектность 10
4 Технические характеристики 11
4.1 Основные технические характеристики блока микрокомпьютера 11
4.2 Основные технические характеристики блока сенсорного экрана 11
4.3 Основные технические характеристики блока программного обеспечения 11
5 Устройство и описание работы изделия 13
5.1 Устройство изделия 13
5.2 Описание работы изделия 14
6 Условия эксплуатации 16
6.1 Правила и особенности размещения изделия 16
6.2 Меры безопасности 16
6.3 Правила хранения и транспортирования 17
ПРИЛОЖЕНИЕ А 18
ПРИЛОЖЕНИЕ Б 26

1 Общие положения

Настоящий паспорт является документом, предназначенным для ознакомления с основными техническими характеристиками, устройством, правилами установки и эксплуатации устройства «Интеллектуальный справочно-тренировочный комплек для школьников» (далее «изделие»).

Паспорт входит в комплект поставки изделия. Прежде, чем пользоваться изделием, внимательно изучите правила обращения и порядок работы с ним. В связи с постоянной работой по усовершенствованию изделия, повышающей его надежность и улучшающей условия эксплуатации, в конструкцию могут быть внесены изменения, не отраженные в данном издании.

1.1 Наименование изделия

Полное наименование изделия – «Интеллектуальный справочно-тренировочный комплек для школьников» (аббревиатура / сокращение).

1.2 Наименования документов, на основании которых ведется проектирование изделия

Проектирвание «Интеллектуальный справочно-тренировочный комплек для школьников» осуществляется на основании требований и положений следующих документов:

- задание на разработку.

1.3 Перечень организаций, участвующих в разработке изделия

Заказчиком проекта «Интеллектуальный справочно-тренировочный комплек для школьников» является Федеральное государственное бюджетное образовательное учреждение высшего образования «Комсомольский-на-Амуре государственный университет» (далее заказчик), находящийся по адресу: 681013, Хабаровский край, г. Комсомольск-на-Амуре, Ленина пр-кт., д. 17.

Исполнителями проекта «Интеллектуальный справочно-тренировочный комплек для школьников» являются Конструкторы студенческого конструкторского бюро название скб (далее СКБ), школьники школ МОУ СОШ № 23, А.М. Грабарь, ФГБОУ ВО «КнАГУ», И.А. Черкашин.

1.4 Сведения об использованных при проектировании нормативно-технических документах

При проектировании использованы следующие нормативно-технические документы:

ГОСТ 2.001-2013. Единая система конструкторской документации. Общие положения.

ГОСТ 2.102-2013. Единая система конструкторской документации. Виды и комплектность конструкторских документов.

ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам.

ГОСТ 2.610-2006. Единая система конструкторской документации. Правила выполнения эксплуатационных документов.

ГОСТ 2.004-88. Единая система конструкторской документации. Общие требования к выполнению конструкторских технологических документов на печатающих и графических устройствах вывода ЭВМ.

ГОСТ 2.051-2006. Единая система конструкторской документации. Электронные документы. Общие положения.

ГОСТ 2.052-2006. Единая система конструкторской документации. Электронная модель изделия. Общие положения.

ГОСТ 2.601-2013. Единая система конструкторской документации. Эксплуатационные документы.

2 Назначение и принцип действия

2.1 Назначение изделия

Платформа предназначена для помощи в изучении и повторении учебного материала. Система, обеспечивающая быстрый поиск информации на основе загру-женных материалов, отвечает на заданные вопросы и указывает на разделы, где информацию можно найти. Основана на технологии RAG, генерация ответа пользователю с учетом дополнительной информации, что позволяет языковой модели формировать ответ по достоверным источникам.

В состав изделия входят:

2.2 Области использования изделия

Изделие может применяться в урочной и внеурочной деятельности обучающихся основного общего и среднего общего образования, в том числе в рамках предпрофильной подготовки.

2.3 Принцип действия изделия

Платформа основана на технологии RAG. Emdedding model преобразует запрос пользователя в эмбеддинг. Получившие вектора используется для поиска релевантных фрагментов в векторной базе данных Qdrant. Найденные данные проходят дополнительную проверку реранкером, после чего передаются языковой модели для формирования ответа, исключительно по загруженным данным в базе знаний.

3 Состав изделия и комплектность

В комплект поставки входит:

- Микрокомпьютер Rasberi Pi 5
- 5-дюймовый сенсорный экран Raspberry Pi HDMI
- Кабели HDMI и USB для подключения экрана
- Паспорт.

4 Технические характеристики

4.1 Основные технические характеристики блока микрокомпьютера

Основные технические характеристики блока приведены в таблице 1.

Таблица 1 – Технические характеристики блока микрокомпьютера

Наименование параметра	Значение
Процессор	ARM Cortex-A76, 4 ядра, 2,4 ГГц
Оперативная память	4 ГБ (LPDDR4X)
Интерфейсы	Ethernet, Wi-Fi, Bluetooth, USB, HDMI,
Питание, В	5 В (USB-C)
Габариты, мм	85 × 56 × 17
Масса нетто, кг	0,048

4.2 Основные технические характеристики блока сенсорного экрана

Основные технические характеристики блока приведены в таблице 2.

Таблица 2 – Технические характеристики блока сенсорного экрана

Наименование параметра	Значение
Тип дисплея	TFT LCD
Разрешение	800 × 480 пикселей
Сенсорная панель	Ёмкостная, 5 точек касания
Интерфейсы	HDMI, USB
Питание, В	5 В
Габариты, мм	121 × 76 × 15
Масса нетто, кг	0.2

4.3 Основные технические характеристики блока программного обеспечения

Основные технические характеристики блока приведены в таблице 2.

Таблица 3 – Технические характеристики блока программного обеспечения

Наименование параметра	Значение
Векторная база данных	Qdrant
Эмбеддинг-модель	На выбор пользователя
Языковая модель (LLM)	На выбор пользователя
Реранкер модель	На выбор пользователя
Top-K фрагментов	Настраивается
Размер чанка	Настраивается
Перекрытие чанков	Настраивается
Поддерживаемые форматы файлов	.md

5 Устройство и описание работы изделия

5.1 Устройство изделия

Изделие выполнено на базе микрокомпьютера Raspberry Pi 5 с подключённым 5-дюймовым сенсорным экраном. Электронная часть сосредоточена в корпусе. Экран подключается к Raspberry Pi 5 через HDMI-порт и USB. Питание на микрокомпьютер подаётся через порт USB-C от блока питания 5 В / 5 А. Питание сенсорного экрана осуществляется напряжением 5 В.

Вычислительный модуль выполнен на базе с четырёхъядерном процессоре ARM Cortex-A76 частотой 2,4 ГГц и 4 ГБ оперативной памяти. Модуль включает интерфейсы: USB, HDMI, Ethernet, Wi-Fi, Bluetooth.

Сенсорный экран представляет собой 5-дюймовый TFT LCD-дисплей с разрешением 800×480 пикселей и ёмкостной сенсорной панелью, поддерживающей до 5 касаний. Экран отображает пользовательский интерфейс и обеспечивает ввод данных через касания.

Программное обеспечение включает векторную базу данных Qdrant, эмбеддинг-модель, LLM и реранкер, подключить можно любую модель на выбор. Поддерживаются файлы в формате .md. Параметры поиска, Top-k фрагменты, размер чанка и перекрытие чанков настраиваются. При загрузке файлов, они разбиваются на чанки, эмбеддинговая модель преобразует в вектора и сохраненяет в Qdrant. Обработка запросов происходит по RAG-конвейерц: эмбеддинг запроса, поиск в Qdrant, реранкинг и генерацию ответа LLM. Интерфейс реализован на Streamlit и выводится на сенсорный экран.

Структурная схема изделия представлена на рисунке 1.

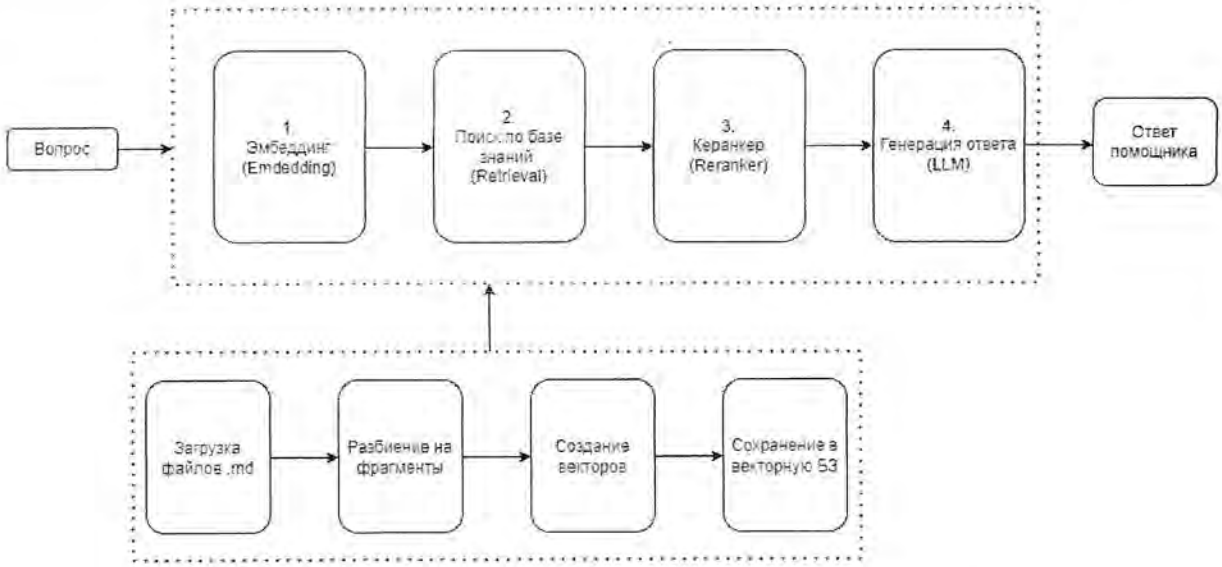


Рисунок 1 – Структурная схема изделия

5.2 Описание работы изделия

Перед началом эксплуатации необходимо подключите все компоненты устройства. Сенсорный экран подключите к микрокомпьютеру Raspberry Pi 5 через HDMI-порт для вывода изображения и через USB-порт для передачи данных сенсорной панели.

Далее идет установка программного обесчения. Для этого есть два спосособа установки: с использованием Docker или без него.

При установке с Docker необходимо клонировать репозиторий с программным кодом, приложение А. После запускаем контейнеры командой «docker compose up -d» Эта команда автоматически поднимет векторную базу данных Qdrant на порту 6333 и веб-приложение на порту 8501. После успешного запуска интерфейс становится доступен по адресу http://localhost:8501.

Без установки Docker нужно иметь Python версии 3.11 или выше и менеджер пакетов uy. Для установки зависимости выполняется команда «uv sync», после чего приложение запускается командой «uv run streamlit run main.py.»

Код программы предоставлен в приложении А.

После запуска приложения на сенсорном экране отображается пользовательский интерфейс. Для работы открываем боковую панель, в разделе настройки подключаем модели: Языковая модель, Эмбеббинг, Реранкер. Мож-

но подключить облачную модель через API или локально. Настраиваем параметры поиска: Top-k фрагменты (финальных), размер чанка (символов) и перекрытие чанков. После выбора всех параметров настройки сохраняются кнопкой «Сохранить настройки».

Последний этапом будет загрузка учебных материалов в формате .md через раздел «база знаний». Загружаем файлы и переходим в чат, задаем вопросы.

Внешний вид пользовательского интерфейса предоставлен в приложении Б.

6 Условия эксплуатации

Изделие выпускается в климатическом исполнении УХЛ 4.2 по ГОСТ 15150-69 и предназначен для использования в стационарных условиях в закрытых помещениях при соответствующих климатических условиях:

- интервал температур от +10 до +35 °С;
- относительная влажность воздуха до 80 % при температуре +25 °С;
- высота над уровнем моря не более 2000 м;
- атмосферное давление от 86,6 до 106 кПа (от 650 до 800 мм рт. ст.).

В помещении, где используется изделие не должно возникать условий для конденсации влаги (выпадения росы). Изделие является электронным прибором, требующим бережного обращения.

Для обеспечения безотказной работы, сохранения точности и его сбережения необходимо соблюдать следующие правила:

- изучить паспорт, прежде чем приступить к работе с изделием;
- предохранять изделие от ударов и повреждений;
- не допускать самостоятельную разборку изделия.

6.1 Правила и особенности размещения изделия

Изделие должно быть расположено на расстоянии не менее 1 м от нагревательных приборов.

ВНИМАНИЕ! При эксплуатации изделия запрещается проводить самостоятельно какие-то либо работы по извлечению и установке внутренних компонентов изделия.

6.2 Меры безопасности

Необходимо соблюдать требования техники безопасности и следующие меры предосторожности:

- не оставлять изделие включенным без наблюдения;

- после транспортировки в холодное время года изделие необходимо выдержать при комнатной температуре не менее двух часов;

- внутренние осмотры и ремонт изделия должны производиться только квалифицированными специалистами;
- не устанавливайте изделие на неустойчивой подставке, стойке или ненадежном кронштейне.

6.3 Правила хранения и транспортирования

Транспортирование изделия в упакованном виде может производиться железнодорожным, автомобильным (в закрытых транспортных средствах), воздушным, речным и морским видами транспорта в соответствии с правилами перевозок грузов, действующих на транспорт данного вида. Условия транспортирования изделия по части воздействия климатических факторов должны соответствовать группе 5 по ГОСТ 15150.

После транспортирования изделие должно быть выдержано не менее 2 часов в транспортной таре при температуре 20±5 °С и относительной влажности воздуха не более 80 %.

Распакованное изделие должно храниться в отапливаемом и вентилируемом чистом помещении при температуре от +5 до +40 °С и относительной влажности воздуха не более 60 %. При температуре ниже 25 °С допускается увеличение относительной влажности до 80 %. Воздух в помещении не должен содержать примесей, вызывающих коррозию металлов, налеты на поверхностях оптических деталей.

ПРИЛОЖЕНИЕ А

(обязательное)

Листинг А.1 – Клонирование репозитория

```
git clone https://github.com/your-username/student-helper.git
cd student-helper
```

Листинг А.2 – код приложения

```
from typing import Any, Dict
import streamlit as st
from loguru import logger
from src.chat_bot import ChatBot
from src.config import ConfigManager
from src.file_manager import FileManager
from src.llm_manager import LLMManager
from src.qdrant_manager import QdrantManager
class StreamlitUI:
    """Главный класс интерфейса Streamlit"""
    def __init__(self) -> None:
        self._init_components()
    def _init_components(self) -> None:
        """Ленивая инициализация компонентов в session_state"""
        if "config_manager" not in st.session_state:
            st.session_state.config_manager = ConfigManager()
        if "qdrant_manager" not in st.session_state:
            st.session_state.qdrant_manager = QdrantManager(st.session_state.config_manager)
        if "llm_manager" not in st.session_state:
            st.session_state.llm_manager = LLMManager(st.session_state.config_manager)
        if "file_manager" not in st.session_state:
            st.session_state.file_manager = FileManager(st.session_state.config_manager)
        if "chat_bot" not in st.session_state:
            st.session_state.chat_bot = ChatBot(
                st.session_state.config_manager,
```

```
            st.session_state.qdrant_manager,
            st.session_state.llm_manager,
        )
    if "chat_history" not in st.session_state:
        st.session_state.chat_history = []
@property
def _cfg(self) -> ConfigManager:
    return st.session_state.config_manager
@property
def _qdrant(self) -> QdrantManager:
    return st.session_state.qdrant_manager
@property
def _llm(self) -> LLMManager:
    return st.session_state.llm_manager
@property
def _files(self) -> FileManager:
    return st.session_state.file_manager
@property
def _bot(self) -> ChatBot:
    return st.session_state.chat_bot
def _render_sidebar(self) -> None:
    s = self._cfg.settings
    with st.sidebar:
        st.title("⚙ Настройки")
        with st.expander("🤖 Языковая модель (LLM)", expanded=False):
            llm_provider = st.selectbox(
                "Провайдер",
                ["cloud", "local"],
                index=0 if s.llm.provider == "cloud" else 1,
                key="sb_llm_provider",
            )
            llm_model = st.text_input("Модель", value=s.llm.model_name, key="sb_llm_model")
            llm_key = st.text_input("API Key", value=s.llm.api_key, type="password",
key="sb_llm_key")
```

```
        llm_url = st.text_input("Base URL", value=s.llm.base_url, key="sb_llm_url")
        llm_temp = st.slider("Temperature", 0.0, 1.0, s.llm.temperature, 0.05)
        llm_tokens = st.number_input(
          "Max tokens", min_value=128, max_value=8192, value=s.llm.max_tokens,
step=128
        )
      with st.expander("[12/34] Эмбеддинги", expanded=False):
        emb_provider = st.selectbox(
          "Провайдер",
          ["cloud", "local"],
          index=0 if s.embeddings.provider == "cloud" else 1,
          key="sb_emb_provider",
        )
        emb_model = st.text_input("Модель", value=s.embeddings.model_name,
key="sb_emb_model")
        emb_key = st.text_input("API Key", value=s.embeddings.api_key, type="password",
key="sb_emb_key")
        emb_url = st.text_input("Base URL", value=s.embeddings.base_url,
key="sb_emb_url")
      with st.expander("🗄 □ Qdrant", expanded=False):
        q_mode = st.selectbox(
          "Режим",
          ["local", "remote"],
          index=0 if s.qdrant.mode == "local" else 1,
          key="sb_q_mode",
        )
        q_host = st.text_input("Host", value=s.qdrant.host, key="sb_q_host")
        q_port = st.number_input(
          "Port", min_value=1, max_value=65535, value=s.qdrant.port, step=1,
key="sb_q_port"
        )
        q_key = st.text_input("API Key (для remote)", value=s.qdrant.api_key,
type="password", key="sb_q_key")
        q_https = st.checkbox("HTTPS", value=s.qdrant.https, key="sb_q_https")
```

```
        q_coll = st.text_input("Коллекция", value=s.qdrant.collection_name, key="sb_q_coll")
      with st.expander("🔍 Параметры поиска", expanded=False):
        ret_k = st.number_input(
          "Top-K фрагментов", min_value=1, max_value=20, value=s.app.retrieval_k, step=1
        )
        chunk_size = st.number_input(
          "Размер чанка (символов)", min_value=100, max_value=2000, value=s.app.chunk_size, step=50
        )
        chunk_overlap = st.number_input(
          "Перекрытие чанков", min_value=0, max_value=500, value=s.app.chunk_overlap, step=10
        )
      if st.button("💾 Сохранить настройки", use_container_width=True):
        self._cfg.update_llm_config(
          provider=llm_provider,
          model_name=llm_model,
          api_key=llm_key,
          base_url=llm_url,
          temperature=llm_temp,
          max_tokens=int(llm_tokens),
        )
        self._cfg.update_embeddings_config(
          provider=emb_provider,
          model_name=emb_model,
          api_key=emb_key,
          base_url=emb_url,
        )
        self._cfg.update_qdrant_config(
          mode=q_mode,
          host=q_host,
          port=int(q_port),
```

```
            api_key=q_key,
            https=q_https,
            collection_name=q_coll,
        )
        self._cfg.update_app_config(
            retrieval_k=int(ret_k),
            chunk_size=int(chunk_size),
            chunk_overlap=int(chunk_overlap),
        )
        self._qdrant.disconnect()
        self._llm.reset()
        st.success("✅ Настройки сохранены!")
    st.divider()
    st.subheader("🗁 База знаний")
    self._render_file_management()
def _render_file_management(self) -> None:
  if "processed_hashes" not in st.session_state:
    st.session_state.processed_hashes = set()
  uploaded = st.file_uploader(
    "Загрузить .md файл",
    type=["md"],
    help="Поддерживаются только Markdown (.md) файлы",
  )
  if uploaded is not None:
    content = uploaded.read()
    file_hash = self._files.compute_hash(content)
    # Защита от повторного вызова после st.rerun()
    if file_hash not in st.session_state.processed_hashes:
      st.session_state.processed_hashes.add(file_hash)
      self._handle_upload(uploaded.name, content, file_hash)  # ← единый метод
  st.markdown("**Загруженные файлы:**")
  try:
    file_list = self._qdrant.list_files()
  except Exception as e:
```

```
        st.error(f"Не удалось получить список файлов: {e}")
        return
    if not file_list:
        st.caption("Файлов пока нет. Загрузите .md файл выше.")
        return
    for info in file_list:
        col_name, col_btn = st.columns([4, 1])
        with col_name:
            st.markdown(f"📄 `{info['file_name']}`")
        with col_btn:
            if st.button("🗑", key=f"del_{info['file_hash']}", help="Удалить из базы"):
                self._handle_delete(info)
def _handle_upload(self, name: str, content: bytes, file_hash: str) -> None:
    # ← принимает уже прочитанные данные, не UploadedFile
    try:
        if self._qdrant.file_hash_exists(file_hash):
            st.sidebar.warning(f"⚠ Файл «{name}» уже есть в базе (дубликат пропущен).")
            return
        with st.sidebar.status(f"Обрабатываю «{name}»…", expanded=True) as status:
            st.write("🔪 Разбиваю на фрагменты…")
            _, chunks = self._files.process_uploaded_file(name, content)
            st.write(f"🔢 Векторизую {len(chunks)} фрагментов…")
            texts = [c["text"] for c in chunks]
            payloads = [c["metadata"] for c in chunks]
            vectors = self._llm.embed_texts(texts)
            st.write("💾 Сохраняю в Qdrant…")
            self._qdrant.ensure_collection(len(vectors[0]))
            self._qdrant.upsert_vectors(vectors, payloads)
            status.update(
                label=f"✅ «{name}» добавлен ({len(chunks)} фрагментов)",
                state="complete",
            )
        st.rerun()
```

```
        except Exception as e:
            logger.error(f"Ошибка при загрузке файла: {e}")
            st.sidebar.error(f"✗ Ошибка загрузки: {e}")
    def _handle_delete(self, file_info: Dict[str, Any]) -> None:
        try:
            self._qdrant.delete_by_file_hash(file_info["file_hash"])
            # После удаления сбрасываем хеш из processed_hashes,
            # чтобы файл можно было загрузить повторно в этой же сессии
            st.session_state.processed_hashes.discard(file_info["file_hash"])
            st.sidebar.success(f"🗑 «{file_info['file_name']}» удалён из базы.")
            st.rerun()
        except Exception as e:
            st.sidebar.error(f"✗ Ошибка удаления: {e}")
    def _render_chat(self) -> None:
        title = self._cfg.settings.app.title
        st.title(f"🏛 {title}")
        st.caption(
            "Задавай вопросы по загруженным учебным материалам. "
            "Я отвечу только по информации из базы знаний."
        )
        for msg in st.session_state.chat_history:
            with st.chat_message(msg["role"]):
                st.markdown(msg["content"])
        user_input = st.chat_input("Введи свой вопрос…")
        if user_input:
            with st.chat_message("user"):
                st.markdown(user_input)
            st.session_state.chat_history.append({"role": "user", "content": user_input})
            with st.chat_message("assistant"):
                with st.spinner("🔍 Ищу ответ в базе знаний…"):
                    history = st.session_state.chat_history[:-1]
                    answer = self._bot.answer(user_input, history)
                st.markdown(answer)
```

```
            st.session_state.chat_history.append({"role": "assistant", "content": answer})
        if st.session_state.chat_history:
            if st.button("🗑 Очистить историю чата"):
                st.session_state.chat_history = []
                st.rerun()
    def run(self) -> None:
        """Запускает Streamlit-приложение"""
        st.set_page_config(
            page_title="Школьный Помощник",
            page_icon="🏫",
            layout="wide",
            initial_sidebar_state="expanded",
        )
        st.markdown(
            """
            <style>
                /* Кнопка Deploy */
                [data-testid="stBaseButton-header"] { display: none !important; }
                /* Hamburger-меню */
                #MainMenu { visibility: hidden !important; }
            </style>
            """,
            unsafe_allow_html=True,
        )
        self._render_sidebar()
        self._render_chat()
if __name__ == "__main__":
    ui = StreamlitUI()
    ui.run()
```

ПРИЛОЖЕНИЕ Б

(обязательное)

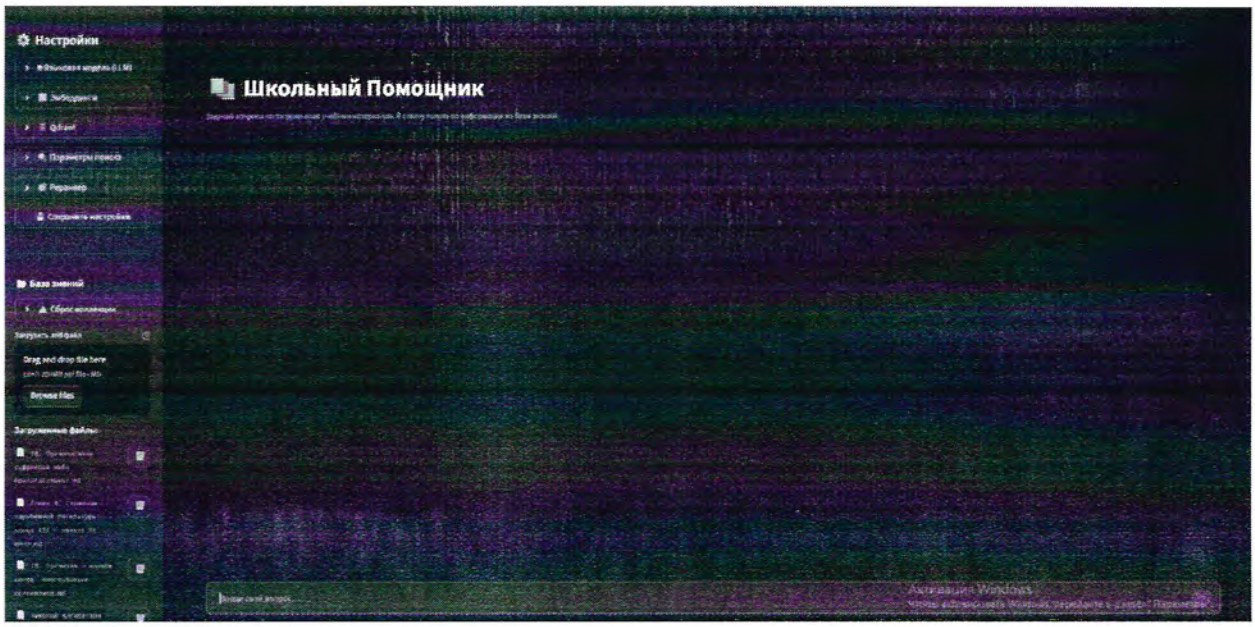


Рисунок Б.1 – Интерфейс программы

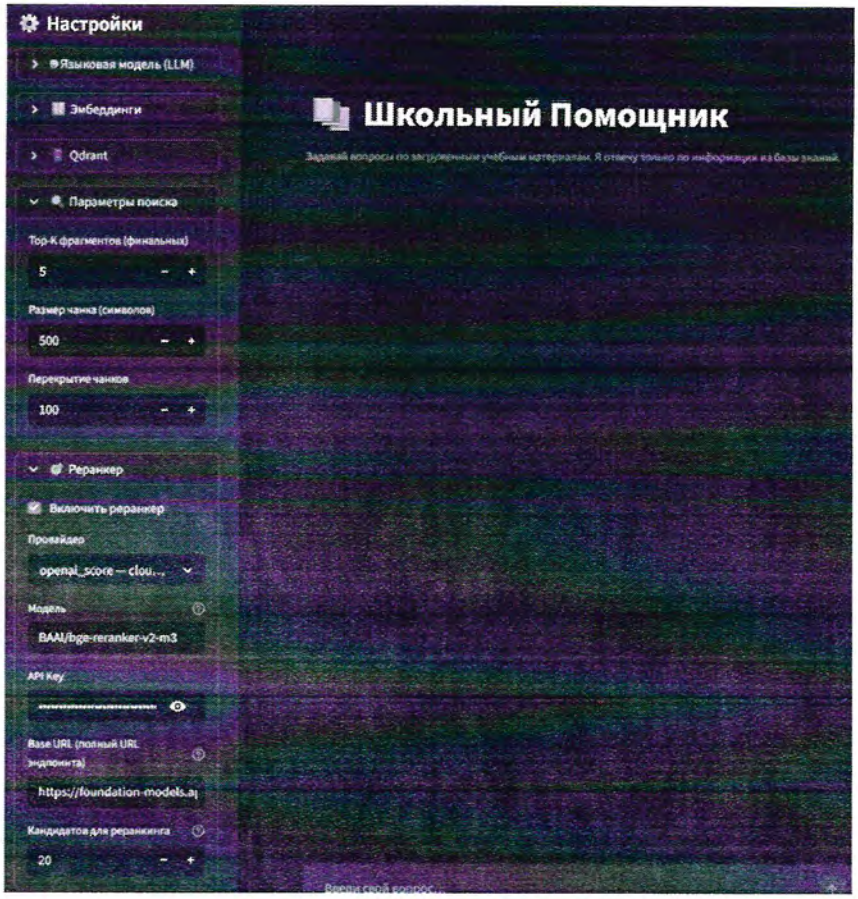


Рисунок Б.2 – Настройки поиска, модели

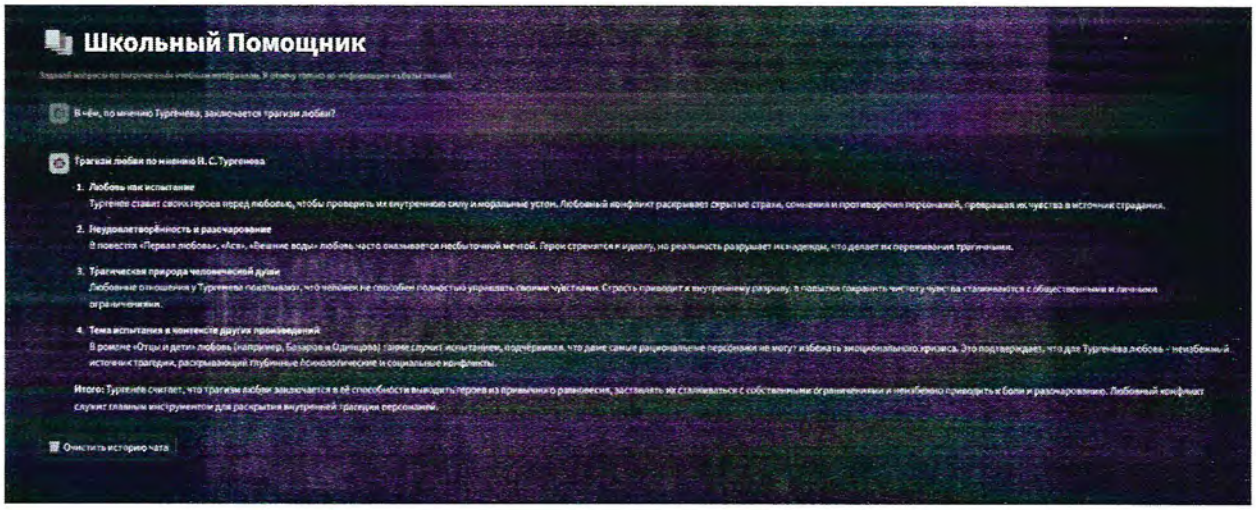


Рисунок Б.3 – Пример ответа

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

СОГЛАСОВАНО

Руководитель ОНиПКРС

_____ Е.М. Димитриади
(подпись)
«19» 06 2026 г.

Декан ФЭУ

_____ А.С. Гудим
(подпись)
«19» 06 2026 г.

УТВЕРЖДАЮ

И.о. проректора по научной работе, д-р техн. наук, профессор

_____ А.В. Космынин
(подпись)
«19» 06 2026 г.

АКТ
о приемке проекта
«Интеллектуальный справочно-тренировный комплек для школьников»

г. Комсомольск-на-Амуре «19» 06 2026 г.

Комиссия в составе представителей:

со стороны заказчика

- С.И. Сухоруков – руководитель СКБ,
- А.С. Гудим – декана ФЭУ

со стороны исполнителя

- Д.М. Грабарь – руководителя проекта,
- Е.Ю. Щербакова – наставник проекта,
- И.А. Черкашин – СОО-Лицей ФГБОУ ВО "КнАГУ",
- А.М. Грабарь – МОУ СОШ № 23

составила акт о нижеследующем:

«Исполнитель» передает проект «Интеллектуальный справочно-тренировный комплек для школьников», в составе:

1. Микрокомпьютер Rasberi Pi 5
2. - 5-дюймовый сенсорный экран Raspberry Pi HDMI
3. Кабели HDMI и USB для подключения экрана

Руководитель проекта	18.06.2026 *(подпись, дата)*	Д.М. Грабарь
Наставник проекта	18.06.2026 *(подпись, дата)*	Е.Ю. Щербакова
Исполнители проекта	18.06.2026 *(подпись, дата)*	И.А. Черкашин
	18.06.2026 *(подпись, дата)*	А.М. Грабарь