|  |  |
| --- | --- |
| **КГЭУ** | МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«КАЗАНСКИЙ ГОСУДАРСТВЕННЫЙ ЭНЕРГЕТИЧЕСКИЙ УНИВЕРСИТЕТ»**  (ФГБОУ ВО «КГЭУ») |

Институт цифровых технологий и экономики

(полное название института)

Информационные технологии и интеллектуальные системы

(полное название кафедры)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ПРОЕКТ**

**по дисциплине** «**Проектный практикум**»

**тема «Тема из задания»**

**Выполнил:**

Хасамова Алсу Руслановна

обучающийся I курса группы ЗИВТм-1-24

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

**Руководитель проекта:**

Хамитов Ренат Минзашарифович,

доцент, к.т.н., доцент каф.ИТИС

(ФИО, должность, кафедра)

Казань, 2025 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc116025499)

[1. ОБЗОРНАЯ ЧАСТЬ 4](#_Toc116025500)

[1.1. Обзор задачи 4](#_Toc116025501)

[1.2. Обзор предварительно обученных моделей машинного обучения 4](#_Toc116025502)

[1.3. Обзор инструментов создания приложения машинного обучения 4](#_Toc116025503)

[2. РАЗРАБОТКА ПРИЛОЖЕНИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА 5](#_Toc116025504)

[2.1. Описание желаемых функций приложения 5](#_Toc116025505)

[2.2. Описание архитектуры приложения 5](#_Toc116025506)

[2.3. Описание реализации приложения 5](#_Toc116025507)

[3. ОРГАНИЗАЦИЯ КОМАНДНОЙ РАБОТЫ 6](#_Toc116025508)

[3.1. Репозиторий с кодом 6](#_Toc116025509)

[3.2. Тестирование системы 6](#_Toc116025510)

[3.3. Непрерывная интеграция 6](#_Toc116025511)

[3.4. Развертывание приложения в облаке 6](#_Toc116025512)

[3.5. Тестирование приложения 6](#_Toc116025513)

[ЗАКЛЮЧЕНИЕ 7](#_Toc116025514)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 8](#_Toc116025515)

[ПРИЛОЖЕНИ 1. КЛЮЧЕВЫЕ ФРАГМЕНТЫ КОДА ПРИЛОЖЕНИЯ 9](#_Toc116025516)

**ВВЕДЕНИЕ**

Современный мир характеризуется высокой степенью глобализации и постоянным обменом информацией между людьми, говорящими на разных языках. Быстрый и качественный перевод текста, дополненный естественной озвучкой, становится неотъемлемой частью повседневной жизни, профессиональной деятельности и образовательных процессов. С развитием технологий машинного обучения и искусственного интеллекта стало возможным автоматизировать сложные задачи перевода и синтеза речи, предоставляя пользователям удобные и эффективные инструменты для коммуникации.

В рамках данного проекта была поставлена задача создания приложения, которое способно выполнять автоматический перевод текста с английского языка на русский, а также преобразовывать полученный перевод в русскоязычную речь. Актуальность этой задачи обусловлена высоким спросом на подобные инструменты в самых разных сферах, от повседневного общения до профессиональной деятельности, связанной с международным сотрудничеством. Кроме того, сочетание технологии машинного перевода и синтеза речи позволяет сделать взаимодействие с контентом более доступным и интуитивно понятным для пользователей.

Цель работы заключается в разработке веб-приложения, которое выполняет перевод английского текста на русский язык с последующей озвучкой перевода, обеспечивая пользователю удобный и интуитивный интерфейс.

Для достижения поставленной цели были сформулированы следующие задачи:

1. Выбор и интеграция подходящих предварительно обученных моделей для перевода текста и синтеза речи.

2. Разработка веб-интерфейса с использованием библиотеки Streamlit, обеспечивающего удобное взаимодействие с пользователем.

Реализация данного проекта демонстрирует, как современные технологии машинного обучения могут применяться для решения повседневных задач, делая жизнь удобнее и продуктивнее.

**1. ОБЗОР ТЕХНОЛОГИЙ АВТОМАТИЧНСКОГО ПЕРЕВОДА И СИНТЕЗА РЕЧИ**

**1.1. Обзор задачи автоматического перевода текста и синтеза речи**

Автоматический перевод текста и синтез речи представляют собой ключевые технологии, которые существенно меняют способы взаимодействия людей с информацией. Задача автоматического перевода заключается в преобразовании текста с одного языка на другой с сохранением исходного смысла и стилистики. В свою очередь, задача синтеза речи направлена на преобразование текстовой информации в естественно звучащую речь, приближенную к человеческой. Эти технологии активно используются в голосовых помощниках, приложениях для обучения языкам, средствах доступности для людей с ограниченными возможностями и в сфере профессионального перевода.

Решение данных задач требует применения методов и моделей, основанных на машинном обучении. В последние годы наблюдается значительный прогресс благодаря развитию трансформерных архитектур, таких как BERT, GPT и, особенно, T5 и MarianMT, которые широко используются для задач перевода. Эти модели обучаются на больших наборах данных, что позволяет им достигать высокой точности и адекватности перевода. Аналогично, для синтеза речи стали стандартом модели, основанные на нейронных сетях, такие как Tacotron и WaveNet, которые способны генерировать голос высокой естественности.

Задача автоматического перевода и синтеза речи остается актуальной благодаря растущему количеству контента на различных языках и необходимости обеспечения быстрой и качественной коммуникации между людьми из разных стран. Решения в этой области направлены на улучшение доступности информации, повышение удобства взаимодействия с текстовыми и аудиоматериалами, а также оптимизацию трудозатрат, связанных с переводом и озвучиванием контента.

В рамках данного проекта задача заключается в интеграции предварительно обученных моделей для выполнения автоматического перевода текста с английского языка на русский и последующего синтеза русскоязычной речи. Основное внимание уделено простоте и эффективности использования приложения, что делает его полезным инструментом для широкого круга пользователей.

**1.2. Обзор предварительно обученных моделей машинного обучения**

В данном разделе представлен обзор предварительно обученных моделей машинного обучения, применимых к задаче перевода текста с последующей его озвучкой. В рамках проекта использованы две модели: Llama2-13b-Language-translate для перевода текста с английского на русский и Bark для озвучивания переведенного текста.

Llama2-13b-Language-translate — это модель машинного перевода, основанная на архитектуре трансформера и предназначенная для перевода с английского на 49 других языков, включая русский. Она была представлена в статье "Multilingual Translation with Extensible Multilingual Pretraining and Finetuning" и обучена на большом количестве многоязычных данных, что позволяет ей обеспечивать высокое качество перевода. Модель использует токенизатор MBart50TokenizerFast и поддерживает указание целевого языка с помощью идентификатора языка (например, "ru\_RU" для русского).

Bark — это модель, разработанная компанией Suno, предназначенная для генерации высокореалистичной, многоязычной речи, а также других аудиоэффектов, включая музыку, фоновые шумы и простые звуковые эффекты. Она способна воспроизводить невербальные коммуникации, такие как смех, вздохи и плач. Модель основана на архитектуре трансформера и предоставляет предварительно обученные контрольные точки, готовые для использования. Bark поддерживает генерацию речи на 13 языках, включая русский, что делает ее подходящей для задач озвучивания переведенного текста.

Выбор этих моделей обусловлен их способностью эффективно решать поставленные задачи перевода текста и его озвучивания, обеспечивая высокое качество и реалистичность результатов.

**1.3. Обзор инструментов создания приложения машинного обучения**

Для создания приложений машинного обучения существует множество инструментов, которые упрощают процесс разработки, предоставляя готовые решения для интеграции моделей, построения интерфейсов и взаимодействия с пользователями. Среди них выделяются такие популярные библиотеки и фреймворки, как Streamlit, Dash, Gradio, а также фреймворки для API, такие как FastAPI и Flask.

Streamlit — это мощный и удобный инструмент, предназначенный для быстрого создания интерактивных веб-приложений. Его основное преимущество заключается в том, что разработчики могут сосредоточиться на логике приложения, а создание пользовательского интерфейса происходит с использованием лаконичного и интуитивно понятного синтаксиса Python. Streamlit предоставляет готовые элементы интерфейса, такие как текстовые поля, кнопки, слайдеры, графики и таблицы, что значительно ускоряет процесс разработки.

Ключевые особенности Streamlit:

* Простота использования. Разработка приложений сводится к написанию обычного Python-кода, что делает инструмент доступным даже для начинающих разработчиков.
* Интерактивность. Мгновенное обновление интерфейса при изменении пользовательских входных данных.
* Поддержка работы с моделями. Streamlit легко интегрируется с библиотеками машинного обучения, такими как TensorFlow, PyTorch и Hugging Face.
* Развёртывание. Возможность быстрого размещения приложений в облачных сервисах, таких как Streamlit Community Cloud, Heroku и другие.

Для реализации данного проекта был выбран именно Streamlit благодаря его простоте и удобству работы. Основная задача приложения — предоставить пользователю инструмент для перевода текста с последующей озвучкой. Streamlit позволяет легко интегрировать модели машинного обучения, созданные с помощью библиотек, таких как Hugging Face Transformers, и быстро создать понятный и удобный интерфейс для взаимодействия с ними.

Реализация интерфейса включает текстовые поля для ввода текста, кнопки для выполнения перевода и генерации аудио, а также возможность прослушивания результата прямо в браузере. Благодаря функционалу Streamlit удалось минимизировать затраты на разработку интерфейса и сосредоточиться на интеграции моделей и оптимизации их работы.

**2. РАЗРАБОТКА ПРИЛОЖЕНИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА**

**2.1. Описание желаемых функций приложения**

Разрабатываемое приложение искусственного интеллекта предназначено для выполнения двух основных функций: перевода текста с английского языка на русский и последующей озвучки переведённого текста. Эти функции позволяют пользователю получить не только текстовый перевод, но и его аудиовизуальное сопровождение, что делает приложение удобным для широкого круга задач, таких как обучение, создание мультимедийного контента или помощь пользователям с ограниченными возможностями.

Приложение должно предоставлять следующий функционал:

1. Ввод текста на английском языке: пользователь должен иметь возможность ввести или вставить текст на английском языке в специальное текстовое поле.

2. Перевод текста: после нажатия кнопки «Обработать текст» приложение должно выполнить перевод введённого текста на русский язык с использованием предварительно обученной модели машинного перевода. Переведённый текст выводится на экран.

3. Генерация озвучки: сразу после этого генерируется аудио с русским озвучиванием переведённого текста. Озвучивание выполняется с помощью предварительно обученной модели синтеза речи.

4. Прослушивание аудио: приложение предоставляет встроенный плеер, который позволяет воспроизводить сгенерированный аудиофайл прямо в интерфейсе.

6. Интуитивно понятный интерфейс: приложение должно быть удобным и понятным для пользователей, даже если они не имеют опыта работы с технологиями машинного обучения. Все элементы интерфейса (кнопки, текстовые поля, плеер) должны быть расположены логично и легко доступны.

Данный функционал ориентирован на создание универсального, простого и удобного инструмента, который можно использовать для различных целей, таких как личное обучение, профессиональная деятельность или работа с контентом.

**2.2. Описание архитектуры приложения**

В качестве основных компонентов приложения и используемых технологий были выбраны следующие:

* Модели машинного обучения:
  + Для перевода текста с английского на русский используется модель T5, обученная для задач перевода. Модель предобучена на большом корпусе данных и применяется для генерации перевода.
  + Для синтеза речи использована модель VITS от Facebook, предназначенная для озвучивания текста на русском языке.
* Используемый инструмент для создания приложения:  
  Приложение разработано с использованием Streamlit — это инструмент для создания интерактивных веб-приложений на языке Python. Он позволяет быстро разрабатывать и развертывать приложения с графическим интерфейсом без необходимости заниматься детальной настройкой фронтенда.
* Код приложения: Приложение состоит из двух файлов:
  + main.py — основной файл приложения, который отвечает за взаимодействие с пользователем через веб-интерфейс.
  + utils.py — файл с реализацией логики для перевода текста и озвучивания.
* Пользовательский клиент: Приложение работает через браузер. Оно предоставляет пользователю интерфейс для ввода текста на английском языке, получения перевода на русский и генерации озвучки переведенного текста.

**Модели машинного обучения**

* Модель для перевода текста:

Для перевода текста используется модель T5 с предварительно обученными весами на задаче перевода. В приложении используется модель utrobinmv/t5\_translate\_en\_ru\_zh\_large\_1024\_v2, которая предсказывает перевод текста с английского на русский язык. Модель применяется через библиотеку transformers, и процесс перевода выполняется с использованием токенизации и генерации переведенного текста.

Пример кода для перевода:

def translate\_text\_to\_russian(text):

prefix = 'translate to ru: '

input\_text = prefix + text

input\_ids = translation\_tokenizer(input\_text, return\_tensors="pt")

with torch.no\_grad():

translated\_ids = translation\_model.generate(\*\*input\_ids)

translated\_text = translation\_tokenizer.batch\_decode(translated\_ids, skip\_special\_tokens=True)[0]

return translated\_text

* Модель для синтеза речи:

Для озвучивания текста на русском языке используется модель VITS от Facebook, которая генерирует аудиофайл из текста. Модель принимает текстовый ввод и преобразует его в аудиофайл, который сохраняется и может быть воспроизведен пользователю.

Пример кода для синтеза речи:

def synthesize\_speech\_from\_text(text, output\_audio\_path="output.wav"):

tts\_inputs = tts\_tokenizer(text, return\_tensors="pt")

with torch.no\_grad():

audio\_output = tts\_model(\*\*tts\_inputs).waveform

sf.write(output\_audio\_path, audio\_output.squeeze().numpy(), tts\_model.config.sampling\_rate)

return output\_audio\_path

**Используемые инструменты**

Streamlit был выбран как инструмент для создания веб-интерфейса приложения. Streamlit позволяет без труда интегрировать Python код в веб-приложение, обрабатывающее пользовательский ввод, и выводить результаты в виде текстов и медиафайлов.

Пример кода для отображения в Streamlit:

st.title("Переводчик и озвучка текста")

st.markdown("Введите текст на английском, чтобы перевести его на русский и получить озвучку.")

user\_input = st.text\_area("Введите текст на английском", height=150)

if st.button("Обработать текст"):

if user\_input.strip():

with st.spinner("Перевожу текст..."):

translated\_text = translate\_text\_to\_russian(user\_input)

st.success("Перевод завершён!")

st.markdown(f"\*\*Переведённый текст:\*\*\n{translated\_text}")

with st.spinner("Генерирую аудиофайл..."):

audio\_path = synthesize\_speech\_from\_text(translated\_text)

st.success("Аудиофайл готов!")

st.audio(audio\_path, format="audio/wav")

else:

st.error("Введите текст для обработки!")

**Архитектура приложения**

Приложение состоит из двух основных блоков:

1. Фронтенд: через интерфейс Streamlit пользователи могут вводить текст, инициировать перевод и озвучивание.
2. Бэкенд: весь процесс обработки текста и озвучивания осуществляется в файле utils.py, где вызываются соответствующие модели перевода и синтеза речи.

Модуль перевода:

* Вводится текст на английском.
* Текст отправляется в модель перевода, и результат возвращается на экран в виде переведенного текста.

Модуль озвучивания:

* После перевода текста, приложение генерирует аудиофайл на русском языке с помощью модели синтеза речи.
* Готовый файл может быть воспроизведен пользователем.

**Развертывание приложения**

Для развертывания приложения используется сервис Streamlit Sharing или любой другой хостинг, поддерживающий Python-скрипты. Приложение загружается на GitHub, а затем через Streamlit Sharing может быть развернуто на веб-сервисе.

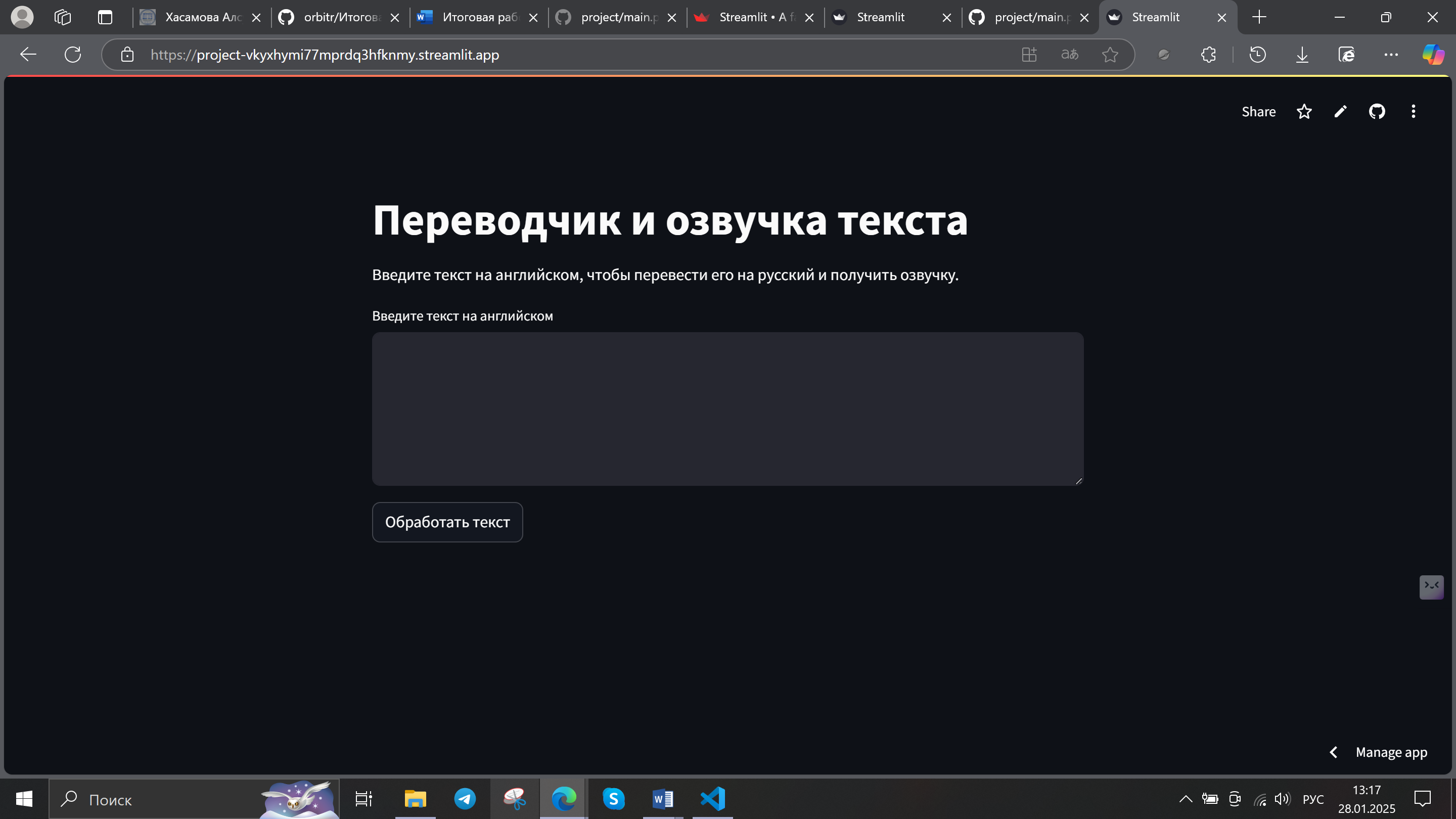
Streamlit Sharing позволяет размещать приложение без необходимости настройки серверов или инфраструктуры, что значительно упрощает развертывание. Однако для конфиденциальных данных или более сложных настроек серверов, рекомендуется использовать другие облачные решения.

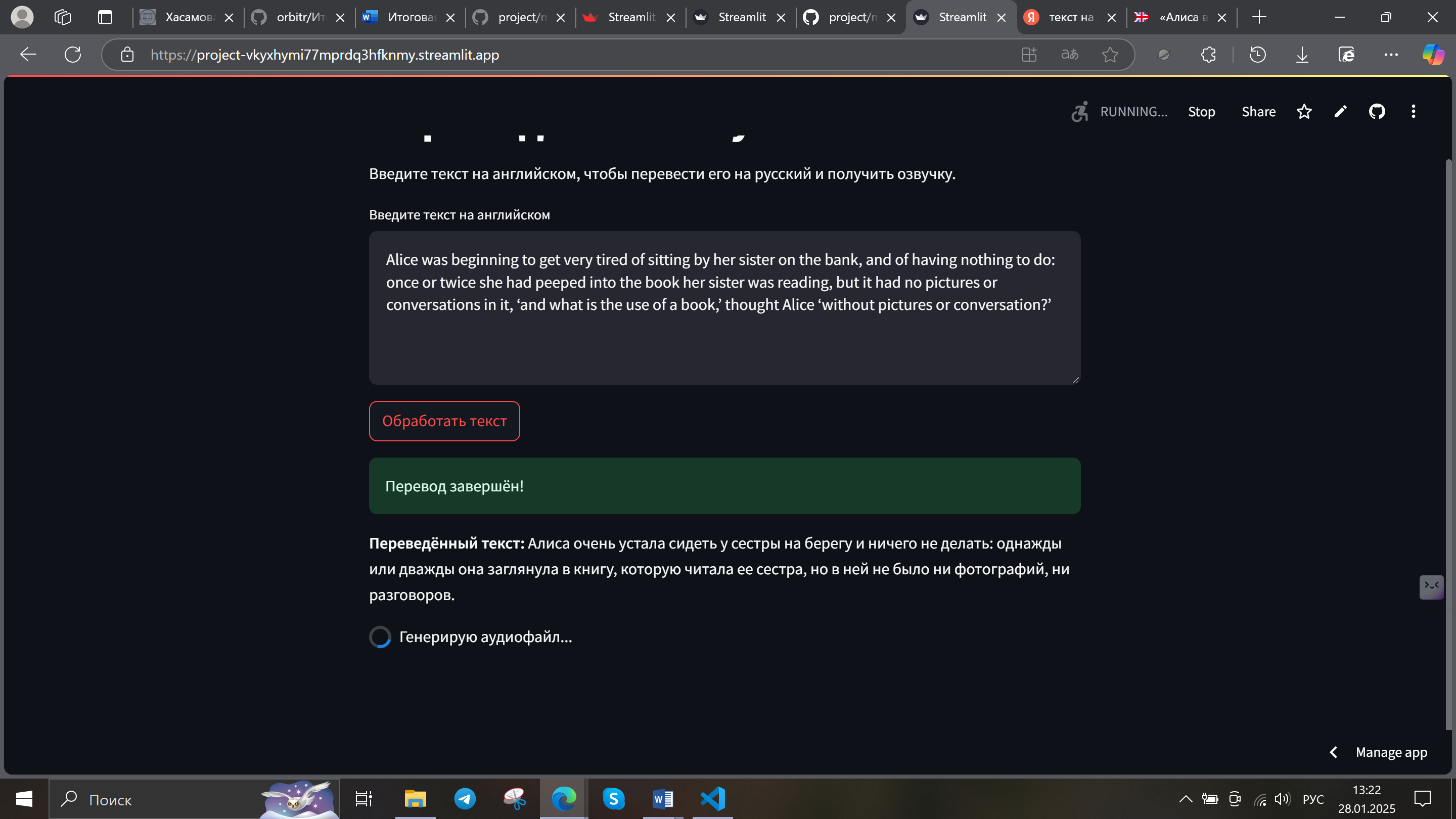
**Код приложения**

Основной файл приложения main.py реализует интерфейс для пользователя. В нем происходит взаимодействие с функциями перевода и озвучивания из модуля utils.py. Все вычисления и обработка запросов происходят на сервере, а пользователю возвращаются результаты в виде текста и аудиофайла.

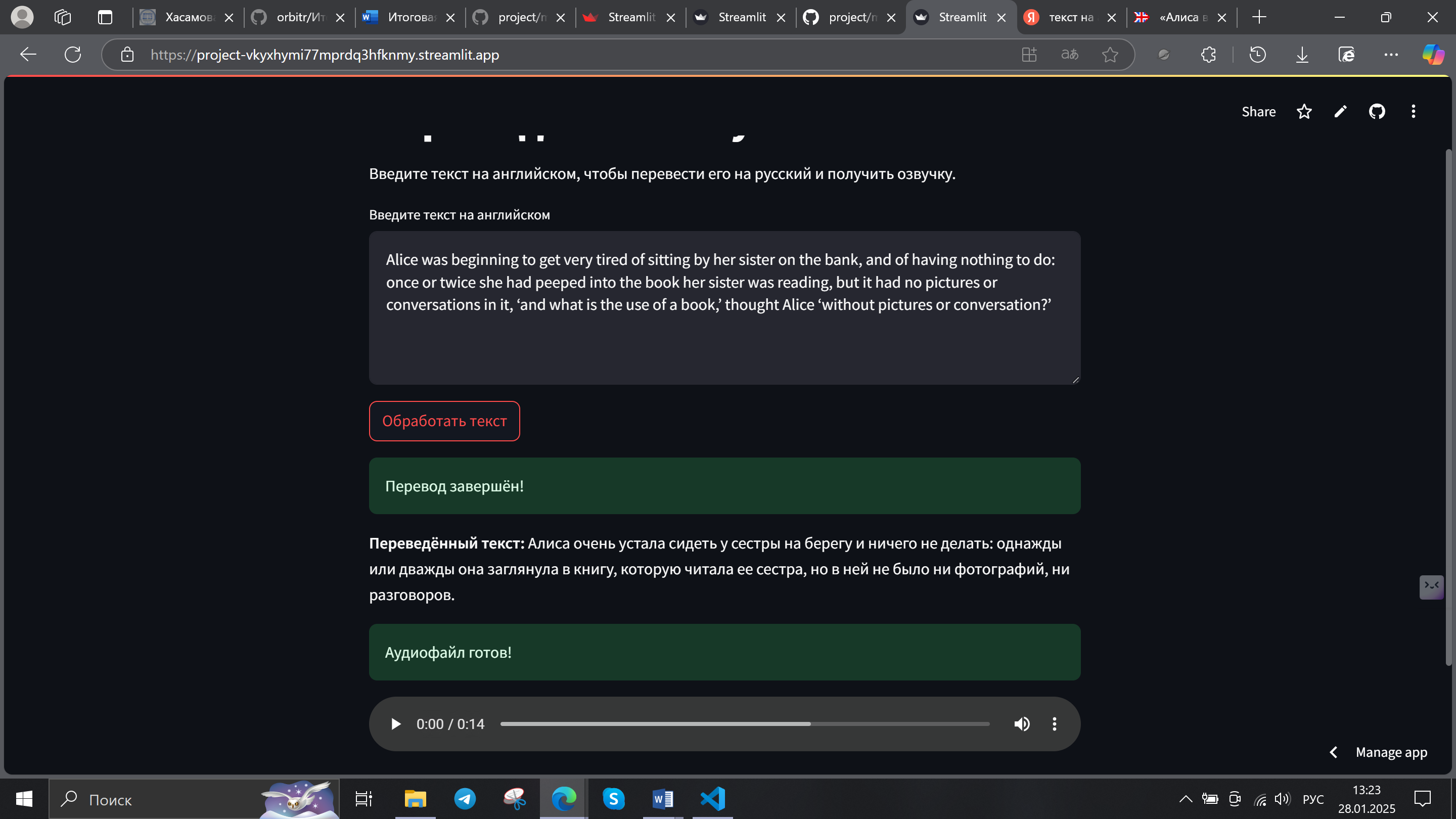
**2.3. Описание реализации приложения**

Приложение размещено по адресу <https://project-vkyxhymi77mprdq3hfknmy.streamlit.app/>. При обращении к приложению открывается основное окно с заголовком, полем загрузки текста и кнопкой обработки текста.





После введения текста и нажатия на кнопку происходит обработка текста, и по окончании этого процесса мы увидим сначала переведенный текст, а после генерации аудиофайла и его.



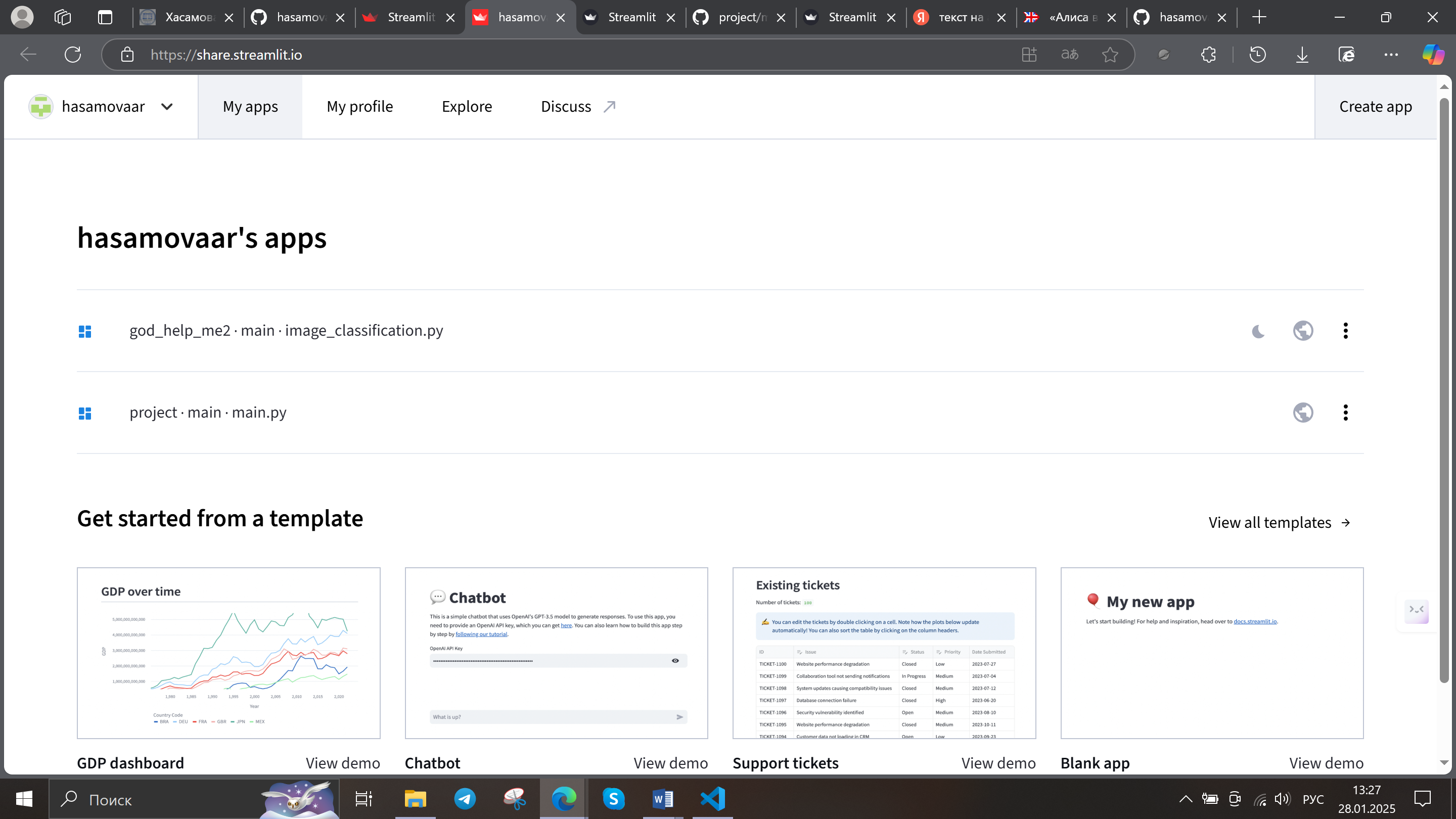
**3. ОРГАНИЗАЦИЯ КОМАНДНОЙ РАБОТЫ**

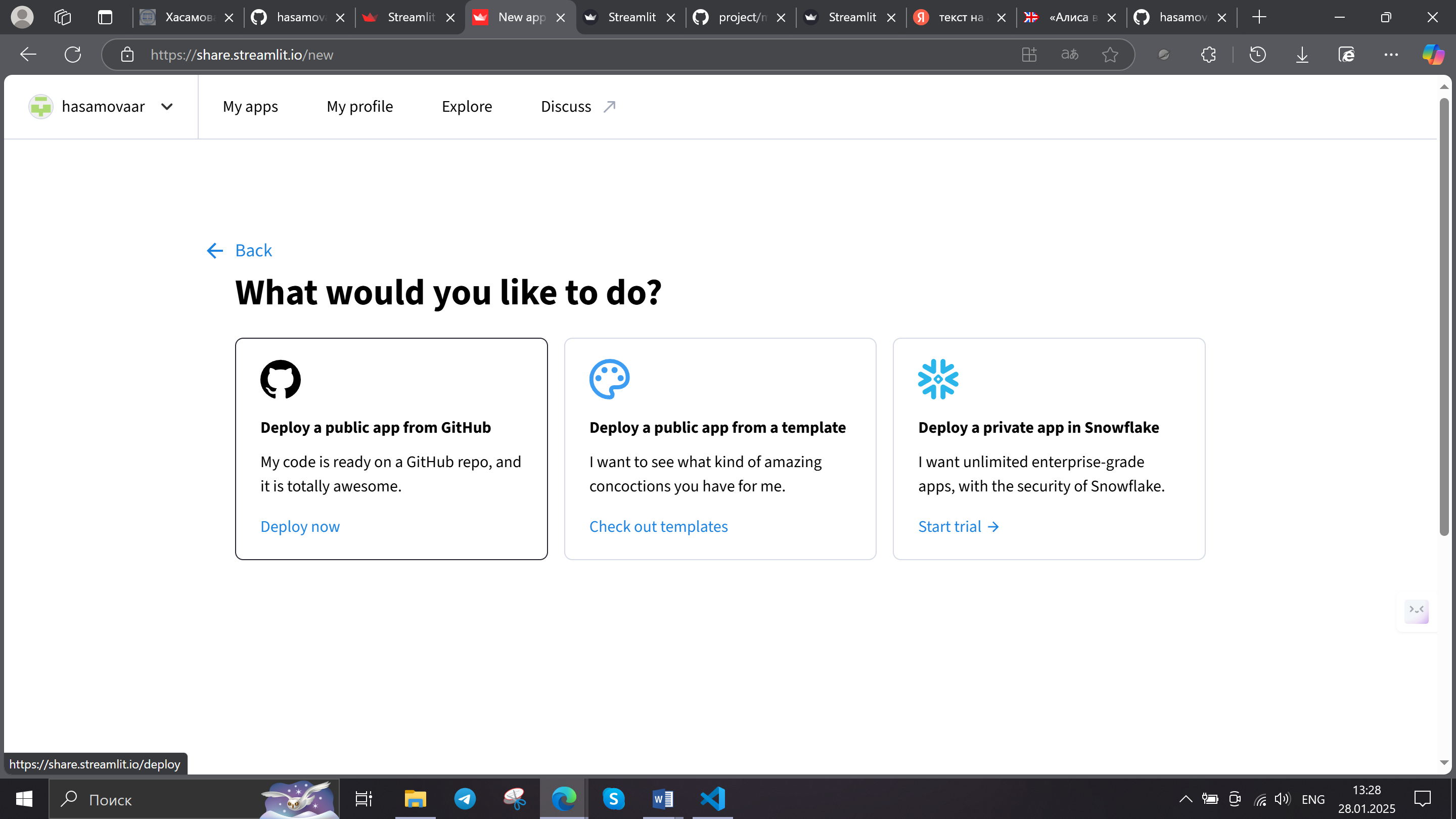
**3.1. Репозиторий с кодом**

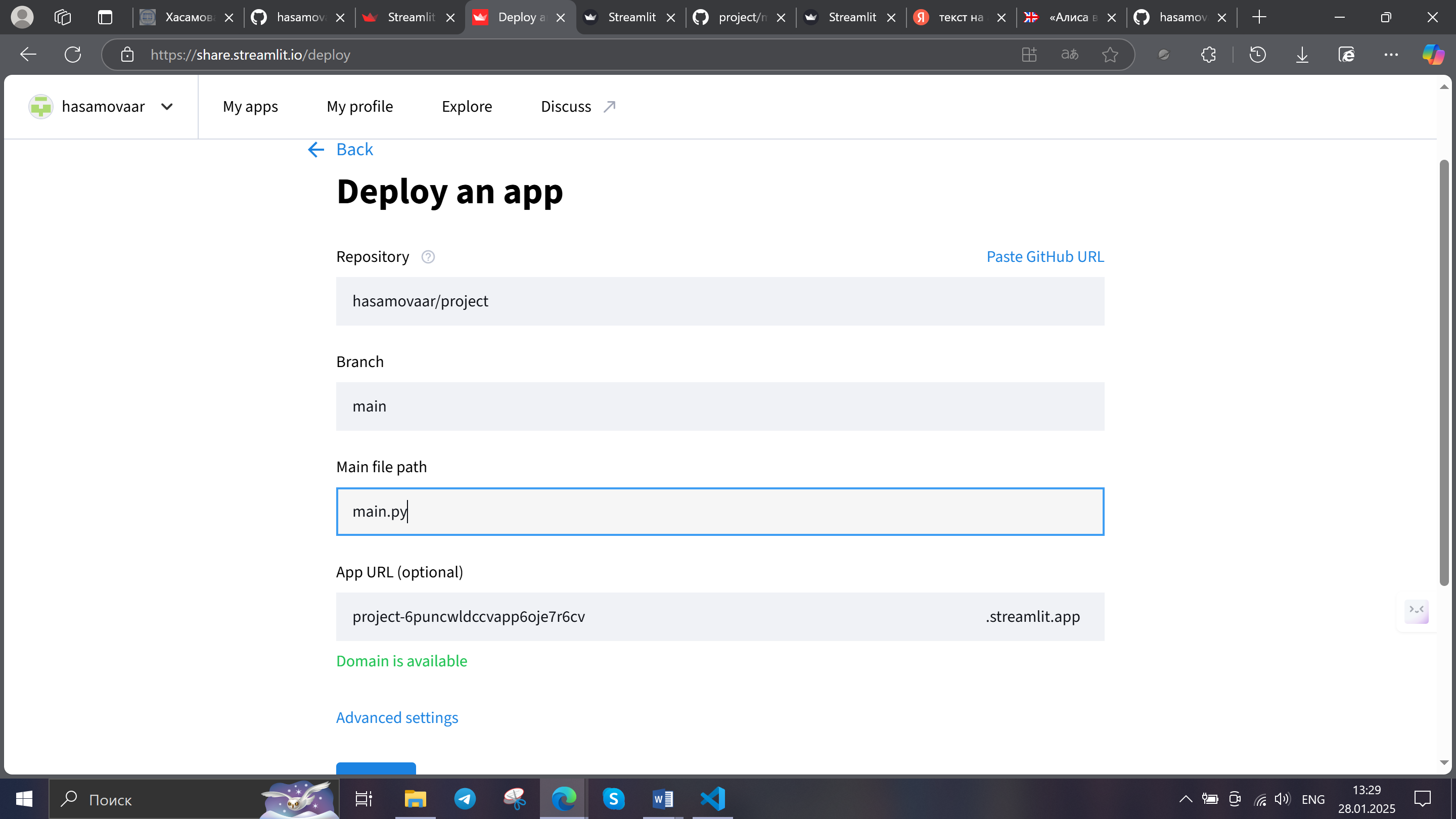
Исходный код приложения размещен в репозитории GitHub по адресу https://github.com/hasamovaar/project.

**3.2. Развертывание приложения в облаке**

Приложение было развернуто в облаке streamlit.io с помощью функции Create app >> Deploy a public app from GitHub.







**ЗАКЛЮЧЕНИЕ**

В ходе выполнения проекта была реализована функциональная система для автоматизации перевода текста с английского языка на русский и его последующей озвучки.

Проект решает задачу предоставления удобного инструмента для перевода текста и его озвучивания, что может быть полезно в образовательных, профессиональных и повседневных целях. Пользователь вводит текст на английском языке, после чего получает переведённый текст на русском и возможность прослушать его в аудиоформате.

Для реализации перевода текста была использована предварительно обученная модель T5 (`utrobinmv/t5\_translate\_en\_ru\_zh\_large\_1024\_v2`), которая обеспечивает высокую точность перевода. Для озвучивания переведённого текста применялась модель `facebook/mms-tts-rus`, специально предназначенная для генерации речи на русском языке. Обе модели позволяют эффективно обрабатывать входные данные и выдавать качественные результаты.

Для создания приложения использовалась библиотека \*\*Streamlit\*\*, которая позволяет легко создавать веб-интерфейсы для работы с машинным обучением. Этот инструмент упростил процесс разработки пользовательского интерфейса и сделал взаимодействие с приложением интуитивно понятным.

Приложение предоставляет пользователю простой интерфейс для ввода текста на английском языке. После обработки текст переводится на русский язык, а затем озвучивается. Пользователь может скачать аудиофайл в формате WAV. Приложение работает локально и доступно для запуска через любой современный браузер.

Использование GitHub позволило эффективно управлять версиями кода и поддерживать прозрачность разработки.

В будущем планируется добавить модуль автоматизированного тестирования, чтобы повысить стабильность приложения и упростить его дальнейшее развитие. Также рассматривается возможность расширения функционала, включая поддержку дополнительных языков перевода и улучшение качества синтеза речи.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Франсуа Шолле. Глубокое обучение на Python. – Питер, 2018. – 400 с. – ISBN 978-5-4461-0770-4.
2. Сквозная платформа машинного обучения с открытым исходным кодом TensorFlow [Электронный ресурс]. URL: https://www.tensorflow.org/ (дата обращения: 07.10.2022).