

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
ИТМО»
(Университет ИТМО)**

Факультет Прикладной информатики

Направление подготовки 45.03.04 Интеллектуальные системы в гуманитарной сфере

Образовательная программа Языковые модели и искусственный интеллект

КУРСОВОЙ ПРОЕКТ

Тема: «Разработка сервиса локализации мобильного приложения»

Обучающийся: Толкачев Тимур Сергеевич, К3161

Санкт-Петербург 2025

СОДЕРЖАНИЕ

Введение	3
1 Актуальность темы.....	3
2 Цель проекта.....	4
3 Задачи проекта.....	4
Основная часть	5
1 Суть проекта	5
2 Процессы работы над проектом	5
2.1 Начало работы.....	5
2.2 Настройка базы данных	6
2.3 Подключение моделей	6
2.4 Обработка запросов.....	6
2.5 Окончание работы.....	7
3 Анализ результатов	10
4 Взаимодействие с командой	11
5 Взаимодействие с руководителем	11
Заключение	13
Список использованных источников	15
Приложение. Техническое задание к проекту	16

ВВЕДЕНИЕ

1 Актуальность темы

На протяжении всего своего существования человечество пытается решить вопрос коммуникации. Возможно ли представить информацию в такой форме, которая будет ясна каждому её воспринявшему? Одной из таких форм коммуникации является язык, который смог дать ответ на данный вопрос локально. Однако подходя к более глобальному обществу возникает такой феномен как «языковой барьер». Решение для данной проблемы не всегда получается интегрировать в мобильные приложения элегантным образом. Зачастую разработчики прибегают к «хардкоду» (когда значение установлено внутри кода программы, тем самым усложняя процесс его изменения извне) переведенных строк, не предусматривая последующие изменения в системе. Некоторые реализуют перевод через так называемые локализационные файлы. Они позволяют изменить текст внутри программы, однако это требует предусмотрительности со стороны разработчиков, дополнительных человеческих ресурсов в лице переводчиков, дополнительной обработки и парсинга сторонних файлов.

Данные решения далеко не идеальны, а потому вопрос локализации до сих пор остается актуальным.

В свете развития больших языковых моделей, которые известны в первую очередь своей способностью работать с текстом, мы начали работу над данным проектом используя возможности этих моделей для перевода текста. Проекты, направленные на перевод текста при помощи машинного обучения, разумеется, уже существуют (к примеру, переводчик DeepL). Однако интеграция данного процесса внутрь мобильного приложения для динамической локализации любой входящей информации является достаточно новым решением, потому что только в недавнее время языковые модели смогли развиваться в достаточной степени, чтобы переводить текст понятно и связно, полностью превосходя результаты более ранних интеграций.

2 Цель проекта

Целью нашего проекта - создать интерфейс, способный принимать на вход запрос на перевод текста и возвращать переведенный результат в виде JSON-конструкции, затем продемонстрировав работу этого интерфейса при помощи MVP, которое взаимодействует с данным интерфейсом.

3 Задачи проекта

Весь проект был разделен на следующие подзадачи:

- Обучить модель
- Разработать API
- Разместить облачный сервер
- Разработать MVP

Результатом данного проекта является рабочее MVP мобильного приложения с поддержкой динамической локализации контента.

ОСНОВНАЯ ЧАСТЬ

1 Суть проекта

Проект представляет собой интерфейс, к которому обращаются приложения, нуждающиеся в переводе текста. Обращения происходят посредством отправления запроса к API, который затем обращается к модели для перевода текста и валюты, затем возвращая переведенные данные в форме JSON-конструкции.

2 Процессы работы над проектом

Мы начали работу над проектом с разделения обязанностей. Всего было выделено три области работы: обучение языковых моделей (Скворцов Денис, Калинин Марк), разработка фронт-энда MVP (Данилевский Алексей), разработка бэк-энда MVP и API (Я, Ахмедов Бахадыр).

По мере работы мы проводили несколько собраний, на которых отчитывались по своей работе, получали друг от друга консультации по взаимодействию наших областей, анализировали полученные результаты.

В целом весь проект можно разделить на четыре сферы:

- 1) Создание фронт-энда MVP
- 2) Создание бэк-энда MVP
- 3) Тренировка моделей
- 4) Создание интерфейса локализации

Сферами моей работы являлись вторая и четвертая.

2.1 Начало работы

Итак, моя работа началась со знакомства с таким инструментом как Docker. Docker — это программа, сделанная для разработчиков разработчиками. Она позволяет под каждое приложение (в нашем случае веб-приложение) выделить отдельный контейнер — изолированную среду, в которой установлены только необходимые пакеты определенной версии и ничего лишнего. Запустив свой первый контейнер, я начал работать над его наполнением: база данных, взаимодействие с моделями, обработка запросов.

2.2 Настройка базы данных

Для реализации базы данных была выбрана система PostgreSQL. Эта одна из самых распространенных систем, так же имеющая документацию для языка программирования Python, однако она вызвала у меня ряд трудностей. В первую очередь у меня зачастую были проблемы с запуском самой БД. По какой-то причине иногда она просто отказывалась запускаться, говоря, что необходимый порт был занят, но это удавалось исправлять при помощи перезапуска службы программы.

Затем несколько раз возникали проблемы с синхронизацией БД и сервера. Пожалуй, это была одна из самых корневых проблем в моей работе, которую удалось решить после добавления пакета `alembic` в проект. Данный пакет позволил безболезненно синхронизировать базу данных на сервер со схемами, прописанными в коде.

2.3 Подключение моделей

К счастью, работа с моделями далась мне гораздо легче: они запускались на отдельном сервере, и все, что мне оставалось сделать – это отправлять им запросы и ждать ответа.

Благодаря слаженной работе с Марком и Денисом мне удалось быстро имплементировать функции для взаимодействия с серверами моделей.

2.4 Обработка запросов

На этом этапе нужно было подключить библиотеку FastAPI к проекту, а также делать запросы в уже настроенную БД. Вместе с Бахадыром мы смогли отправить первый запрос на сервер и получить на него ответ. Далее следовало лишь добавить 2 энд-поинта (`"/products/"`, `"/products/{id}"`) для реализации

бэк-энда нашего MVP. После чего Бахадыр реализовал функции для загрузки, чтения и удаления объектов в БД.

Localization Service API ^{v1} OAS 3.1

</swagger/v1/swagger.json>

Localization Service API

Localization Service API Documentation Docs description

ProductsApi

POST	/products/	Create Product	⌵
GET	/products/	Get All Products	⌵
GET	/products/{id}	Get Product By Id	⌵
DELETE	/products/{id}	Delete Product	⌵

Рисунок 1 – энд-поинты и методы API

2.5 Окончание работы

Закончив работать над всеми все вышеупомянутыми функциями, я приступил к завершению бэк-энда. Подтянув API для перевода валют “exchangeratesapi” я модифицировал функции обработки запросов так, чтобы они могли возвращать текст, переведенный через языковые модели и цену товара, пересчитанную на местную валюту пользователя.

После чего весь код был протестирован локально (вывод консоли локального сервера представлен на рисунке 3) и залит на облачный сервер (рисунок 2).

Затем облачный сервер успешно синхронизировался с приложением, и мы наконец увидели результат нашей работы (рисунок 4).

У нас удалось собрать цельное, рабочее мобильное приложение. Последовала еще пара доработок с нашей стороны, мы добавили несколько проверок, чтобы сервер не отключался в случае получения неожиданной или необработанной информации.

В конце мы еще пару раз протестировали функционал приложения и наконец завершили нашу работу.

EC2

>

Instances

>

i-0ab6dc9d51b9dc5b2

Instance summary for i-0ab6dc9d51b9dc5b2

(LocSer) Info

Connect

Instance state

▼

Actions

▼

Updated less than a minute ago

Instance ID

i-0ab6dc9d51b9dc5b2

Private IPv4 addresses

172.31.33.49

Instance state

Running

Hostname type

IP name: ip-172-31-33-49.eu-central-1.compute.internal

Instance type

t2.micro

Public IPv4 address

3.120.132.186 | [open address](#)

IPv6 address

–

Public IPv4 DNS

ec2-3-120-132-186.eu-central-1.compute.amazonaws.com | [open address](#)

Private IP DNS name (IPv4 only)

ip-172-31-33-49.eu-central-1.compute.internal

Answer private resource DNS name

IPv4 (A)

Elastic IP addresses

–

Рисунок 2 – Развертка локального сервера

8

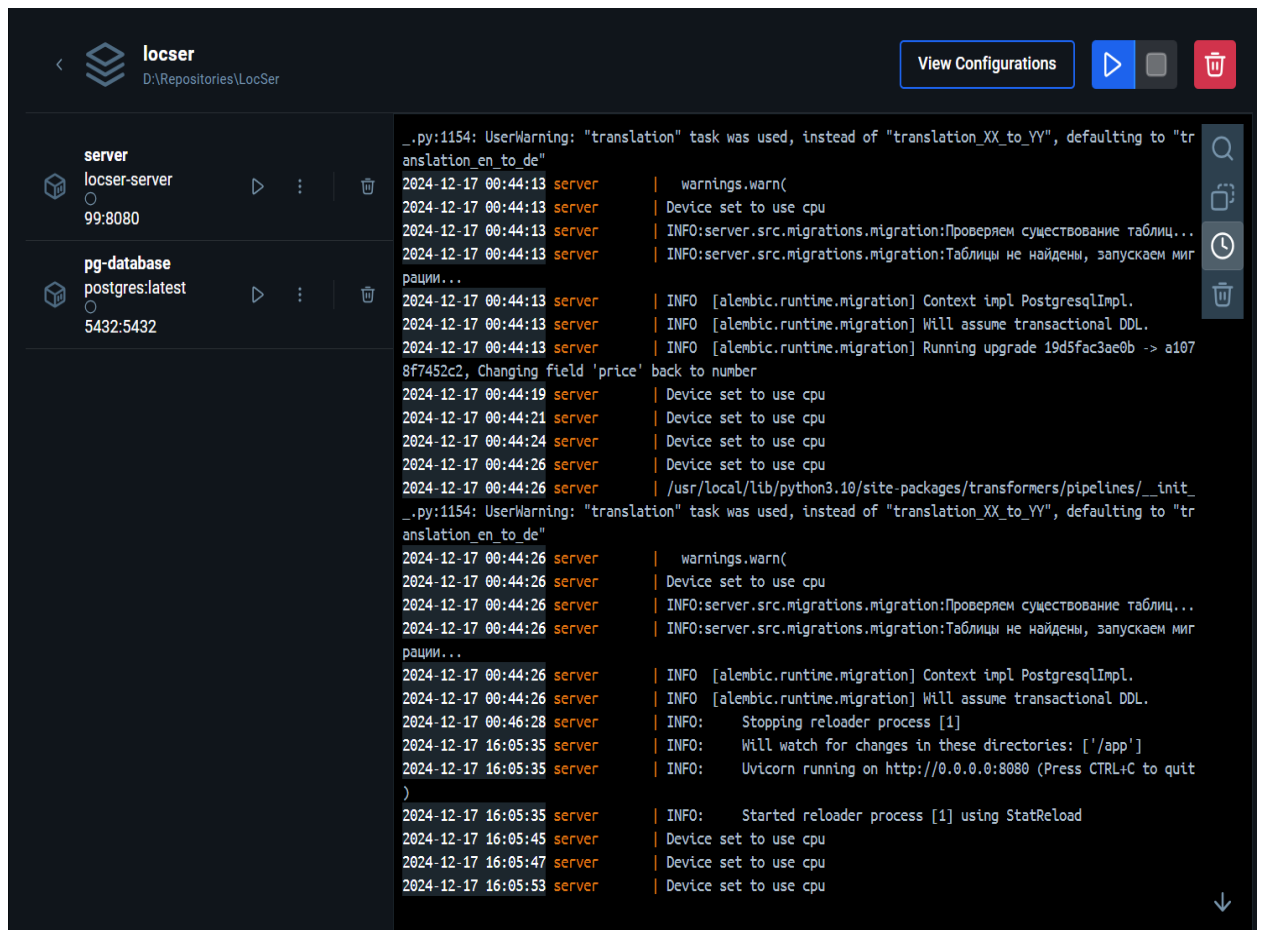


Рисунок 3 – Развертка локального сервера

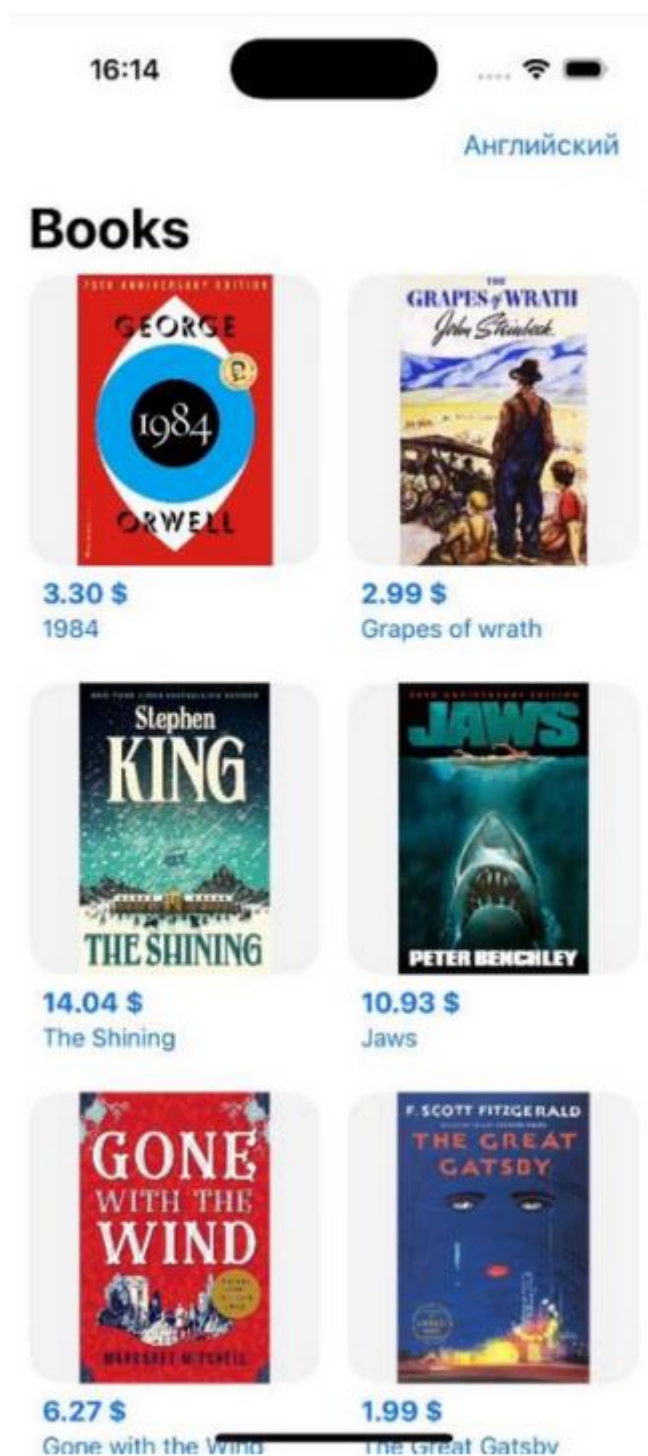


Рисунок 4 – Внешний вид приложения

3 Анализ результатов

В целом, я доволен результатами. Это был первый раз, когда я выступал в роли бэк-энд разработчика и я смог полностью выполнить свои задачи несмотря на то, что на моем пути возникло несколько трудностей. Благодаря участию в этом проекте я приобрел навыки работы с базами данных,

языковыми моделями, веб-приложениями, интерфейсами программирования и серверами.

Зачастую разработка могла уходить в тупик, но такие моменты позволяли мне разглядеть ситуацию с нескольких сторон и найти решение. У меня не было четкого расписания работы, я просто садился и решал данные мне задачи в полной мере своих возможностей.

Поскольку все задачи удалось выполнить в срок, я считаю работа вышла плодотворной, как и в разработке самого продукта, так и в получении новых знаний.

4 Взаимодействие с командой

По моему мнению работа с командой прошла гладко и даже ускорила процесс разработки в несколько раз. Вместе мы смогли корректно распределить задачи и решить их, постепенно объединяя наши решения в единую систему. Мы устраивали несколько групповых звонков, чтобы проверить соответствие и совместимость построенных нами компонентов. К концу работы мне удалось поработать со всеми членами команды включая тех, кто отвечал за тренировку моделей, их развертку, а также тех, кто разрабатывал фронт-энд.

До этого мне еще ни разу не удавалось поработать командой над одним проектом, поэтому данный опыт безусловно бесценен для меня. Теперь я знаю, как происходят взаимодействия внутри команды, насколько важны мягкие навыки в разговоре с людьми, как передать свою мысль и как воспринять чужую.

5 Взаимодействие с руководителем

Руководитель смог грамотно распределить роли, четко определить свое видение проекта, поставить достижимые задачи. При любых затруднениях мы могли обращаться к нему за советом, и он объяснял нам сложные принципы и понятия, предлагал возможные решения или помогал разбить задачу на более понятные подзадачи, если что-то было неясно.

Я считаю, что руководитель отлично справился со своими обязанностями и сумел реализовать потенциал всей команды, равномерно распределив задачи между участниками и систематически проверяя прогресс работы подкоманд, давая обратную связь, которая так необходима при работе над продуктом. Как руководитель Игорь Манаков не требует сверхурочной работы, но всегда ожидает выполнения обязанностей со стороны участников команды и всячески поддерживает их.

ЗАКЛЮЧЕНИЕ

Выполнив все наши задачи, мы достигли поставленной цели — разработали MVP сервиса и получили наглядный результат. Разумеется, наибольшую трудность представляли собой программные вычисления. Языковым моделям требуются долгие сеансы тренировок, для того чтобы превзойти уровень обычных дословных переводчиков текста. Учитывая наши ограничения по времени и финансам, я считаю, что мы смогли получить достаточный результат от тренировок этих моделей. Если этот проект понадобится вывести на более массовый уровень, то при наличии ресурсов можно будет развить качество перевода на превосходный уровень.

Аналогично с кодовой базой проекта. Изначально она давалась довольно тяжело, но с каждым днем становилось все легче и легче с ней работать. По итогу цель была достигнута, однако если потребуется внести любые изменения в проект, то их можно будет с легкостью интегрировать в уже существующий гибкий код программы.

Учитывая вышесказанное, можно сказать, что у нас получился очень достойный продукт. Я рад, что мне удалось поработать с таким проектом и узнать, как практически из воздуха можно собрать полноценное приложение, как можно поднять свой сервер и позволить ему взаимодействовать с языковыми моделями, как работает API.

Я очень рад тому, что смог внести вклад в подобный проект, так как я изначально интересовался темой локализации. Конечно, как человек, разрабатывающий бэк-энд проекта я был далеко от самого процесса перевода, но мне все равно было интересно видеть, как постепенно взаимодействуют наши компоненты. Я считаю, что этот проект был одним из самых насыщенных, которые я когда-либо делал в своей жизни. Для него я использовал более 50 различных библиотек, связывал их вместе, подводил результаты к идеалу.

Для меня было в новинку работать в команде, однако это никак не помешало разработке нашего проекта. Все с интересом исследовали свои сферы, делились результатами и помогали друг другу при взаимной работе. В целом я полностью доволен проделанной работой и ощущаю себя готовым к сотрудничеству с другими людьми в будущем.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. SQLAlchemy Документация [Электронный ресурс] – URL: <https://docs.sqlalchemy.org/en/20/> (дата обращения 01.12.2024).
2. Alembic Документация [Электронный ресурс] – URL: <https://alembic.sqlalchemy.org/en/latest/> (дата обращения 01.12.2024).
3. How to Deploy a Machine Learning Model on AWS SageMaker? [Электронный ресурс] – URL: <https://medium.com/@data.pilot/how-to-deploy-a-machine-learning-model-on-aws-sagemaker-b04b01b569ca> (дата обращения 01.12.2024).
4. Docker 101 Tutorial [Электронный ресурс] – URL: <https://www.docker.com/101-tutorial/> (дата обращения 01.12.2024).
5. SQLAlchemy – Full Stack Python [Электронный ресурс] – URL: <https://www.fullstackpython.com/sqlalchemy.html> (дата обращения 01.12.2024)

ПРИЛОЖЕНИЕ. ТЕХНИЧЕСКОЕ ЗАДАНИЕ К ПРОЕКТУ

1. **Название:** Разработка сервиса локализации мобильного приложения
2. **Цель (назначение):** Разработать MVP сервиса для локализации, который может принимать нужные строки приложения, а затем выводить их переведёнными на нужные языки.
3. **Сроки выполнения:** Начало – 31.10.2024 Окончание – 18.12.2024
4. **Исполнитель проекта (руководитель проекта):** Игорь Манаков
5. **Термины и сокращения:**

MVP (minimal viable product) – продукт, обладающий минимальными, но достаточными для удовлетворения первых потребителей функциями

Параллельный корпус – большие собрания текстов с выравниванием по предложениям с сопоставлением одного языка с другим.

Датасет – коллекция табличных данных

Loss (Функция потерь) – функция, характеризующая потери при неправильном принятии решений на основе наблюдаемых данных

6. Технические требования:

- сервис должен уметь переводить тексты асинхронно на выбранных языках
- сервис должен базироваться на удалённом сервере
- сервис должен поддерживать хостинг картинок
- сервис должен иметь API для доступа к модели

7. Содержание работы

Таблица 1 – Содержание работы

№			
1	Собрать данные	31.10.2024 – 06.11.2024	Калинин Марк Алексеевич
2	Изучить FastAPI	31.10.2024 – 10.11.2024	Ахмедов Бахадыр Бахтиёрович
3	Изучить Docker	31.10.2024 –	Толкачёв Тимур

		10.11.2024	Сергеевич
4	Познакомиться с Hugging Face	31.10.2024 – 10.11.2024	Скворцов Денис Александрович
5	Обработать данные	10.11.2024 – 17.11.2024	Скворцов Денис Александрович
6	Создать интерфейс для интеграции с backend	10.11.2024 – 17.11.2024	Калинин Марк Алексеевич
7	Добавить перевод навигационных компонентов	17.11.2024 – 20.11.2024	Калинин Марк Алексеевич
8	Разработать Endpoint'ы с конфигурацией мобильного	10.11.2024 – 24.11.2024	Ахмедов Бахадыр Бахтиёрович
9	Развернуть сервер	10.11.2024 – 24.11.2024	Толкачёв Тимур Сергеевич
10	Оптимизировать модель	17.11.2024 – 01.12.2024	Скворцов Денис Александрович
11	Обучить модель	20.11.2024 – 01.12.2024	Калинин Марк Алексеевич
12	Интегрироваться в сервис	24.11.2024 – 01.12.2024	Толкачёв Тимур Сергеевич

13	Разработать контракт API	24.11.2024 – 01.12.2024	Ахмедов Бахадыр Бахтиёрович
14	Сверстать UI мобильного приложения	31.10.2024 – 08.12.2024	Данилевский Алексей Александрович
15	Сгенерировать документацию API	01.12.2024 – 08.12.2024	Ахмедов Бахадыр Бахтиёрович
16	Доработать модель	01.12.2024 – 20.12.2024	Скворцов Денис Александрович
17	Добавить кэширование данных	01.12.2024 – 20.12.2024	Толкачёв Тимур Сергеевич
18	Интегрировать S3	08.12.2024 – 20.12.2024	Ахмедов Бахадыр Бахтиёрович
19	Создать сервисный слой приложения	08.12.2024 – 20.12.2024	Данилевский Алексей Александрович