**Содержание**

[Введение 3](#_Toc165826347)

[1 Постановка задачи 4](#_Toc165826348)

[1.1 Обзор аналогичных решений 4](#_Toc165826349)

[1.2 Спецификация требований 6](#_Toc165826350)

[2 Проектирование web-приложения 7](#_Toc165826351)

[2.1 Проектирование вариантов использования 7](#_Toc165826352)

[2.2 Структура web-приложения 8](#_Toc165826353)

[2.3 Проектирование базы данных 8](#_Toc165826354)

[2.4 Проектирование сервера web-приложения 13](#_Toc165826355)

[3 Разработка web-приложения 19](#_Toc165826356)

[3.1 Разработка бэкэнда 19](#_Toc165826357)

[3.2 Разработка фронтэнда 22](#_Toc165826358)

[4 Тестирование web-приложения 25](#_Toc165826359)

[5 Руководство пользователя 28](#_Toc165826360)

[Заключение 34](#_Toc165826361)

[Список используемых источников 35](#_Toc165826362)

[Приложение А 36](#_Toc165826363)

Введение

В современном мире глобализации и цифровых технологий трудно переоценить значение точного и быстрого перевода текстов. Перевод становится важным инструментом для преодоления языковых барьеров и эффективной коммуникации в различных сферах жизни — от корпоративных блогов до локализации web-сайтов. Это особенно актуально в условиях постоянного увеличения объема информации и необходимости оперативного доступа к контенту на разных языках.

Наша жизнь сегодня пронизана международным взаимодействием и многообразием информации, что делает переводы не просто полезными, но и необходимыми. Профессиональные и частные пользователи, работающие с большими объемами текста, сталкиваются с трудностью выбора подходящих решений для качественного перевода. Часто этот процесс может быть длительным и трудоемким.

Именно поэтому возникла необходимость в инновационных решениях, способных упростить и ускорить процесс перевода текстов. Web-приложение «GPTranslate» удовлетворяет эту потребность, предлагая пользователям мощный инструмент для перевода текстов при помощи искусственного интеллекта. Оно предоставляет возможность выбрать модель перевода и настроить стиль в зависимости от конкретных требований и целей, будь то технический перевод, маркетинговый контент или любой другой тип текста.

Основная цель проекта – создать универсальное и удобное в использовании web-приложение, которое позволит пользователям эффективно справляться с переводами текстов различной сложности и объема. Важным аспектом является предоставление настроек для выбора подходящей модели и стиля перевода, что обеспечивает высокое качество и точность перевода.

Актуальность данного проекта обусловлена растущим спросом на качественные решения для перевода текстов в условиях глобализации и необходимости оперативной локализации контента. Web-приложение «GPTranslate» сокращает время на перевод и предоставляет пользователям возможность гибкой настройки параметров перевода, что делает ее ценным инструментом в сфере текстового контента.

Целевая аудитория приложения включает широкий спектр пользователей: от профессиональных переводчиков и сотрудников международных компаний до владельцев web-сайтов и блогеров, нуждающихся в качественном и быстром переводе своих материалов.

1 Постановка задачи и обзор аналогичных решений

1.1 Постановка задачи

Web-приложение предназначено для перевода статей и управления этим процессом. Функционал приложения разделен на роли: пользователь, модератор и администратор.

Пользователь начинает с регистрации (или создания администратором) и аутентификации в системе. После успешной аутентификации он может выполнять следующие действия:

– создать статью: пользователь может загрузить статью из документа или набрать текст самостоятельно. По желанию пользователь может выбрать язык исходной статьи;

– создавать и изменять свои конфигурации переводчика: пользователь может создавать, изменять и удалять конфигурации, включающие в себя модель для перевода, стиль перевода и конечные языки, на которые необходимо выполнить перевод;

– выполнять перевод статьи: пользователь выбирает статью для перевода, выбирает модель, стиль и языки, на которые необходимо выполнить перевод, и запускает процесс перевода;

– оценивать перевод статьи: пользователь может оценивать переводы своих статей отметками «нравится» и «не нравится»;

– создавать жалобу на перевод статьи: если пользователь считает, что перевод некачественный, он может подать жалобу, указав причины. Также пользователь может создавать комментарии к своим открытым жалобам;

– редактировать личную информацию: пользователь может изменять своё имя и сбрасывать пароль на новый.

Модератор может быть создан администратором. После аутентификации он может выполнять следующие действия:

– просматривать и изменять список жалоб: модератор может получать список открытых жалоб, удовлетворять или отклонять их;

–оставлять комментарии к открытым жалобам.

Администратор может быть создан специальной консольной командой или другим администратором. После аутентификации ему доступны следующие действия:

– управление списком пользователей: администратор может получать список всех пользователей, создавать, изменять и удалять любых пользователей;

– управление списком запросов перевода: администратор может получать список запросов к моделям перевода, создавать новые объекты запросов и изменять и удалять существующие;

– управление списком моделей перевода: администратор может создавать записи о моделях перевода и изменять и удалять существующие записи;

– получать статистику жалоб: администратору доступна статистика, показывающая, сколько жалоб было подано и удовлетворено по разным моделям и запросам перевода, что позволит ему удалять неудовлетворительно работающие модели и запросы.

1.2 Обзор аналогичных решений

В качестве первого аналогичного решения был выбран сервис DeepL. Он известен своей высокой точностью и естественностью перевода по сравнению с другими сервисами. Внешний вид страницы сервиса представлен на рисунке 1.1.

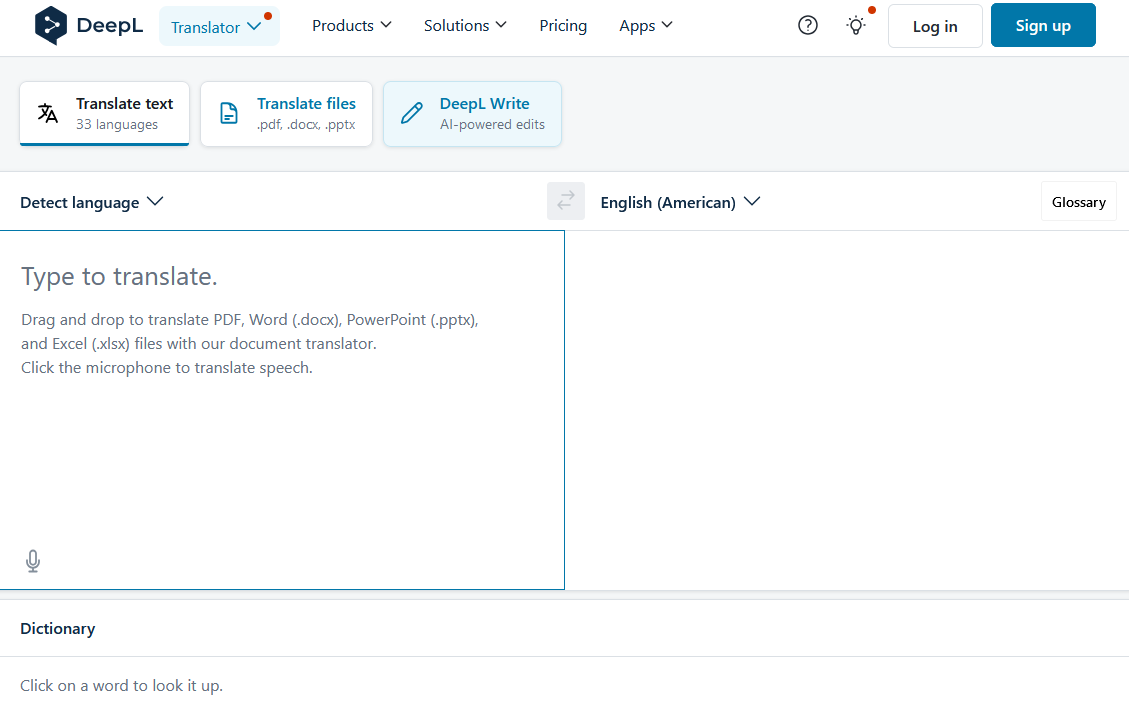


Рисунок 1.1 – Страница сервиса DeepL

Он использует нейронные сети для обработки текста и предлагает перевод на множество языков.

В качестве второго аналогичного решения был рассмотрен сервис Google Translate. Внешний вид страницы данного сервиса представлен на странице 1.2.

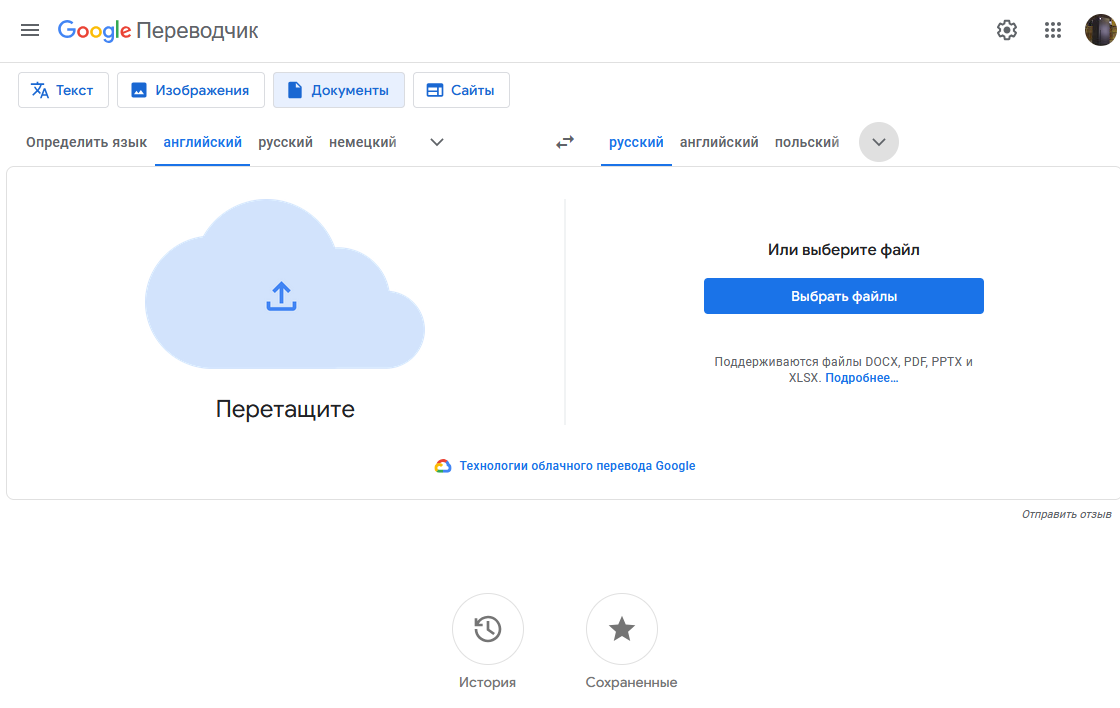


Рисунок 1.2 – Страница сервиса Google Translate

Это один из самых известных и широко используемых сервисов машинного перевода. Google Translate использует мощные нейронные сети и постоянно улучшается.

В качестве третьего аналогичного решения был рассмотрен сервис Wordvice. Внешний вид его страницы представлен на рисунке 1.3.

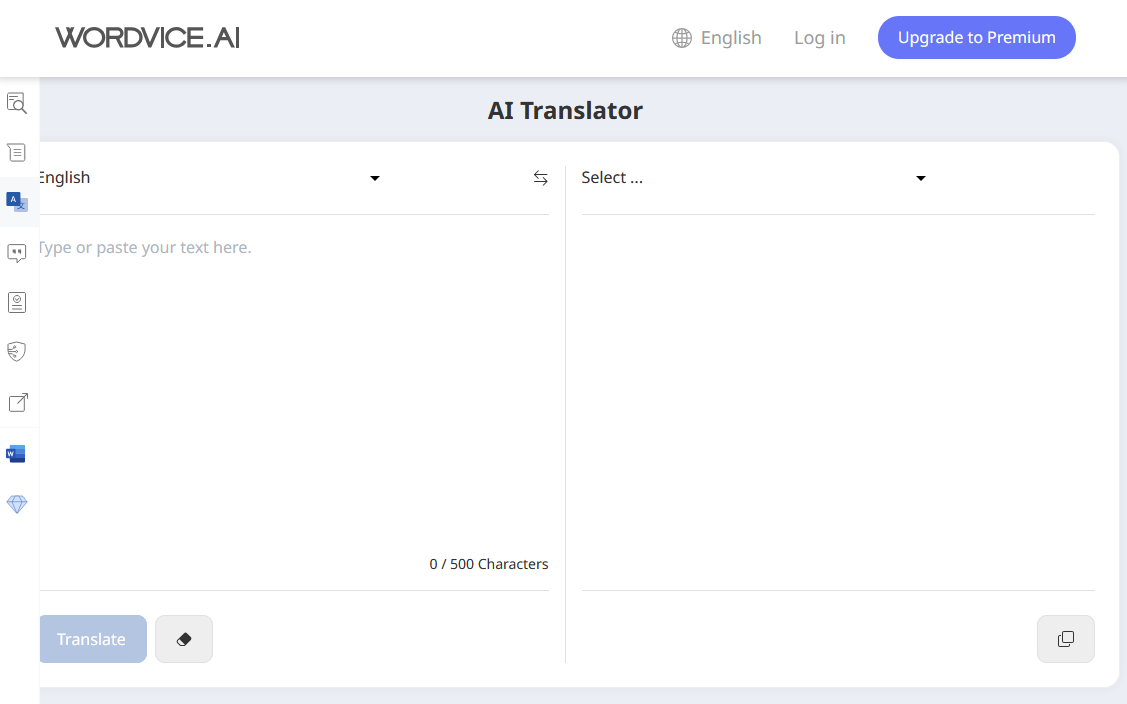


Рисунок 1.3 – Страница сервиса Wordvice

Он использует нейронные сети для перевода текста, предоставляет интеграцию с Microsoft Word и услуги обобщения и перефразирования текста при помощи искусственного интеллекта, а также поддерживает множество языков.

1.3 Выводы

При рассмотрении аналогичных решений были выделены их особенности.

Сервис DeepL обладает следующими преимуществами:

– высокое качество перевода: DeepL часто хвалят за более естественный и точный перевод по сравнению с конкурентами, особенно в контексте отдельных предложений и коротких текстов;

– поддержка многих языков: сервис предлагает широкий диапазон языков для перевода;

– удобный интерфейс: простой и интуитивно понятный интерфейс;

– быстрая обработка: перевод выполняется относительно быстро.

Сервис DeepL имеет следующие недостатки:

– ограничения по объему текста: для очень больших текстов использование данного сервиса может быть менее предпочтительным, чем Google Translate;

– стоимость: бесплатная версия имеет серьёзные ограничения на объём переводимого текста, платная подписка может быть невыгодной для недостаточно больших объемов перевода;

– меньшее количество дополнительных функций: по сравнению с Google Translate, DeepL предлагает меньше дополнительных функций, таких как транслитерация или распознавание речи.

Сервис Google Translate обладает следующими преимуществами:

– широкая доступность и поддержка языков: данный сервис поддерживает огромное количество языков и доступен практически везде;

– обработка больших объемов текста: данный сервис хорошо справляется с переводом длинных документов и объёмных web-страниц;

– множество дополнительных функций: данный сервис предлагает множество дополнительных функций, таких как транслитерация, распознавание речи, перевод изображений и т.д.;

– интеграция с другими сервисами Google: данный сервис удобно интегрируется с другими продуктами Google.

Сервис Google Translate имеет следующие недостатки:

– качество перевода может варьироваться: качество перевода может быть неравномерным, особенно для сложных текстов, технических терминов и идиом. Может быть более «дословным», чем DeepL;

– неестественность перевода: результаты перевода могут звучать неестественно или механически

Сервис Wordvice обладает следующими преимуществами:

– поддержка многих языков: как и остальные рассмотренные сервисы, Wordvice предоставляет возможность перевода на множество различных языков;

– интеграция с Microsoft Word: данный сервис предоставляет возможность перевода текста непосредственно внутри документа Word без необходимости переключения окон и копирования текста;

– минималистичный интерфейс: интерфейс данного сервиса не перегружен лишними деталями, пользователь может легко выполнять свои задачи, не отвлекаясь на лишние элементы интерфейса.

Сервис Wordvice имеет следующие недостатки:

– отсутствие возможности загружать свои документы: при необходимости перевода больших объёмов текста исходный текст неободимо копировать в текстовое поле, что снижает удобство пользования сервисом; (ПРОВЕРИТЬ)

– меньшее количество функций: как и DeepL, Wordvice имеет меньшее по сравнению с Google Translate количество функций.

Выбор платформы (ОПЦИОНАЛЬНО)

* Высокая производительность: FastAPI основан на ASGI (Asynchronous Server Gateway Interface), что позволяет обрабатывать запросы асинхронно. Это обеспечивает высокую скорость и эффективность, особенно при большом количестве одновременных пользователей. Это критично для приложения с потенциально множеством запросов на перевод, оценки, и модерацию.
* Простота использования и разработки: FastAPI имеет простой и интуитивно понятный синтаксис, основанный на Python. Это ускоряет разработку и снижает порог входа для разработчиков, знакомых с Python. Быстрая разработка особенно важна для MVP (минимально жизнеспособного продукта) или для быстрого прототипирования.
* Автоматическая генерация документации: FastAPI автоматически генерирует интерактивную документацию на основе кода с помощью OpenAPI и Swagger UI. Это значительно упрощает работу с API и позволяет разработчикам и другим участникам проекта легко понимать, как работает приложение.
* Валидация данных: FastAPI предоставляет мощные инструменты для валидации входных данных, что помогает предотвратить ошибки и повысить надежность приложения. Это особенно важно для защиты от некорректных данных, которые могут повредить системе.
* Совместимость с другими технологиями: FastAPI хорошо интегрируется с другими популярными библиотеками Python, такими как SQLAlchemy (для работы с базами данных), databases (для работы с различными базами данных через один API), и Vue.js (как указано в ТЗ для клиентской части). Эта совместимость упрощает разработку и развертывание.
* Отличное сообщество и поддержка: FastAPI имеет активное и дружелюбное сообщество, что обеспечивает доступ к помощи и ресурсам при возникновении проблем. Это важно для долгосрочного развития и поддержки проекта.

2 Проектирование web-приложения

2.1 Функциональность web-приложения

Функциональные возможности web-приложения представлены в диаграмме вариантов использования, представленной на рисунке 2.1.

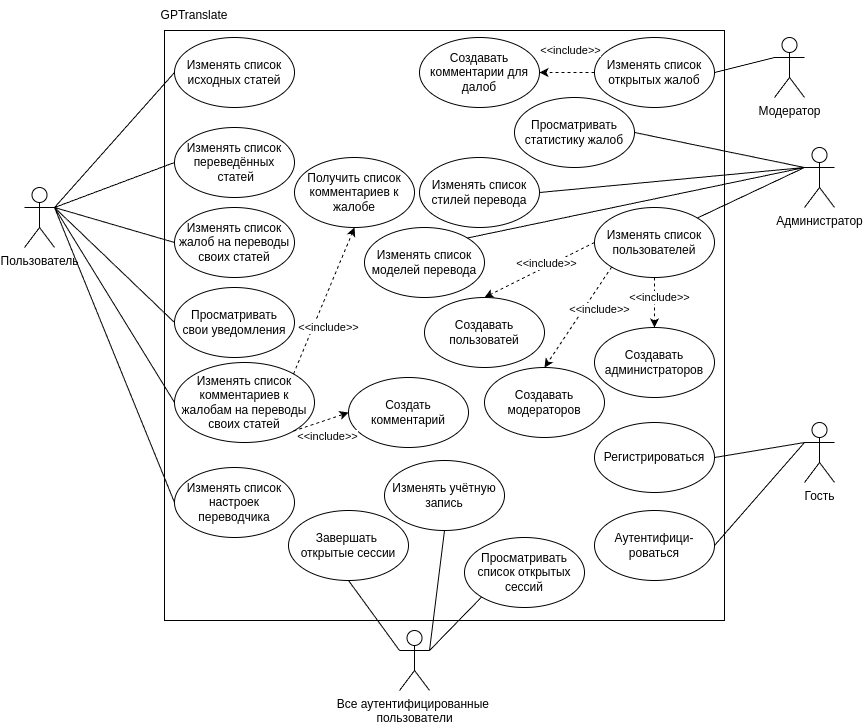


Рисунок 2.1 – Диаграмма вариантов использования

Перечень ролей и их назначение приведены в таблице 2.1.

Таблица 2.1 – Назначение ролей пользователей в web-приложении

|  |  |
| --- | --- |
| Роль | Назначение |
| Гость | Регистрация и аутентификация |
| Пользователь | Загрузка и запуск перевода статей, получение переводов |
| Модератор | Рассмотрение жалоб на переводы |
| Администратор | Управление пользователями, запросами перевода, моделями перевода |

Функционал пользователя представлен в таблице 2.2.

Таблица 2.2 – Функционал пользователя

|  |  |
| --- | --- |
| Вариант использования | Пояснение |
| Изменять список исходных статей | Загружать или вводить с клавиатуры исходные статьи, получать их список, изменять содержимое исходных статей, удалять их |
| Изменять список переведённых статей | Запускать перевод исходных статей, получать их список, оставлять оценку переводам статей, удалять переводы статей |
| Изменять список жалоб на переводы своих статей | Создавать жалобы на переводы своих статей, получать их список, закрывать открытые жалобы на переводы своих статей |
| Просматривать свои уведомления | Получать список непрочитанных уведомлений |
| Изменять список комментариев к жалобам на переводы своих статей | Получать список комментариев, создавать комментарии к открытым жалобам на переводы своих статей |
| Изменять список настроек переводчика | Получать список своих конфигураций, создавать новые, обновлять и удалять существующие |

Функционал модератора представлен в таблице 2.3.

Таблица 2.3 – Функционал модератора

|  |  |
| --- | --- |
| Вариант использования | Пояснение |
| Изменять список открытых жалоб | Получать список открытых жалоб на статьи пользователей, получать списки комментариев и создавать новые комментарии к ним, принимать или отклонять жалобы |

Функционал администратора представлен в таблице 2.4.

Таблица 2.4 – Функционал администратора

|  |  |
| --- | --- |
| Вариант использования | Пояснение |
| Просматривать статистику жалоб | Получать данные о том, какая часть переводов при помощи каждой модели получает жалобы и какая их доля удовлетворяется модераторами |
| Изменять список стилей перевода | Создавать новые стили, обновлять и удалять существующие |
| Изменять список моделей перевода | Добавлять информацию о новых моделях, изменять и удалять существующие записи |
| Изменять список пользователей | Получать список пользователей, создавать новых, изменять и кдалять существующих |

Таким образом, пользователю доступно

2.2 Проектирование базы данных

Согласно схеме вариантов использования была создана база данных. Её структура представлена на рисунке 2.2.

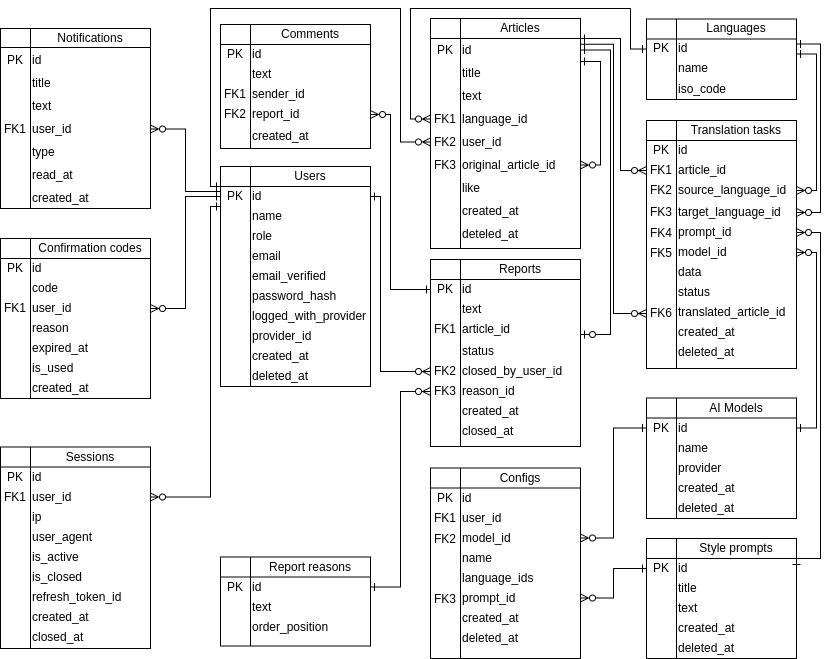


Рисунок 2.2 – Логическая схема базы данных

База данных содержит тринадцать таблиц, хранящих информацию о пользователях, сессиях, статьях и прочих данных. Типы данных были выбраны согласно документации [1]. Назначение таблиц базы данных представлено в таблице 2.5.

Таблица 2.5 – Назначение таблиц базы данных

|  |  |
| --- | --- |
| Таблица | Назначение |
| Users | Хранит информацию о пользователях (имя, адрес электронной почты и хеш пароля для аутентификации и так далее) |
| Sessions | Хранит информацию о сессиях пользователей (идентификатор пользователя, флаг активности, время создания и так далее) |
| Confirmation\_codes | Хранит информацию о кодах подтверждения адреса электронной почты и сброса пароля |
| Languages | Хранит информацию о доступных для перевода языках (название, ISO код) |
| Articles | Хранит информацию о статьях (заголовок, текст, идентификатор пользователя и так далее) |
| Report\_reasons | Хранит информацию о доступных причинах для жалобы на перевод статьи (текст, позиция в списке для сортировки) |
| Reports | Хранит информацию о жалобах на переводы статей (идентификатор статьи, текст, идентификатор, причина и так далее) |
| Report\_comments | Хранит информацию о комментариях к жалобам на переводы статей (текст, идентификатор пользователя, идентификатор жалобы, дата и время создания) |
| Style\_prompts | Хранит информацию о запросах перевода с разными стилями (название, текст и так далее) |
| AI\_Models | Хранит информацию о моделях искусственного интеллекта, использующихся для перевода (название, поставщик и так далее) |
| Configs | Хранит информацию о конфигурациях переводчика, которые могут использоваться пользователями для упрощения запуска перевода своих статей (идентификаторы запроса перевода, модели, языков и так далее) |
| Translation\_tasks | Хранит информацию о задачах перевода, которые считываются отдельным процессом и выполняются им (идентификаторы статьи, модели, исходного и конечного языков, статус и так далее) |
| Notifications | Хранит информацию об уведомлениях пользователей (идентификатор пользователя, текст и так далее) |

Описание столбцов таблицы Users представлено в таблице 2.6.

Таблица 2.6 – Описание таблицы Users

|  |  |  |
| --- | --- | --- |
| Название столбца | Тип данных | Описание |
| id | uuid | Идентификатор пользователя, первичный ключ |
| name | varchar(20) | Имя пользователя |
| email | varchar(50) | Адрес электронной почты пользователя |
| email\_verified | boolean | Флаг, указывающий, был ли подтверждён адрес электронной почты пользователя |
| password\_hash | varchar(60) | Хеш пароля соискателя |
| role | enum user\_role | Роль пользователя (пользователь, модератор, администратор) |
| logged\_with\_provider | varchar | Название провайдера OAuth 2.0, использовавшегося для регистрации |
| provider\_id | varchar | Идентификатор пользователя, полученный от провайдера OAuth при регистрации |
| created\_at | timestamp without timezone | Дата и время создания пользователя |
| deleted\_at | timestamp without timezone | Дата и время удаления пользователя |

Описание столбцов таблицы Sessions представлено в таблице 2.7.

Таблица 2.7 – Описание таблицы Sessions

|  |  |  |
| --- | --- | --- |
| Название столбца | Тип данных | Описание |
| id | uuid | Идентификатор сессии, первичный ключ |
| user\_id | uuid | Идентификатор пользователя, которому принадлежит сессия, внешний ключ |
| ip | varchar(15) | IPv4 адрес узла, из которого была открыта сессия |
| user\_agent | varchar(100) | User agent клиента (например, браузера) |
| is\_closed | boolean | Флаг, указывающий, была ли сессия закрыта |
| refresh\_token\_id | uuid | Идентификатор refresh токена, связанного с данной сессией |
| created\_at | timestamp without timezone | Дата и время создания сессии |
| closed\_at | timestamp with out timezone | Дата и время закрытия сессии |

Описание столбцов таблицы Confirmation\_codes представлено в таблице 2.8.

Таблица 2.8 – Описание таблицы Confirmation\_codes

|  |  |  |
| --- | --- | --- |
| Название столбца | Тип данных | Описание |
| id | integer | Идентификатор кода, первичный ключ |
| code | varchar | Строковое значение кода |
| reason | enum confirmationtype | Тип кода (подтверждение адреса электронной почты, сброс пароля) |
| user\_id | uuid | Идентификатор пользователя, для которого предназначен данный код подтверждения, внешний ключ |
| expired\_at | timestamp without timezone | Временная отметка, после которой код будет считаться истёкшим |
| is\_used | boolean | Флаг, указывающий, был ли код использован |
| created\_at | timestamp without timezone | Дата и время создания кода |

Описание столбцов таблицы Languages представлено в таблице 2.9.

Таблица 2.9 – Описание таблицы Languages

|  |  |  |
| --- | --- | --- |
| Название столбца | Тип данных | Описание |
| id | integer | Идентификатор языка, первичный ключ |
| name | varchar | Отображаемое название языка |
| iso\_code | varchar | ISO код языка |

Описание столбцов таблицы Articles представлено в таблице 2.10.

Таблица 2.10 – Описание таблицы Articles

|  |  |  |
| --- | --- | --- |
| Название столбца | Тип данных | Описание |
| id | uuid | Идентификатор статьи, первичный ключ |
| title | varchar(50) | Название статьи |
| text | varchar(10240) | Текст статьи |
| user\_id | uuid | Идентификатор пользователя, которому принадлежит статья, внешний ключ |
| language\_id | integer | Идентификатор языка статьи, внешний ключ |
| original\_article\_id | uuid | Идентификатор статьи, переводом которой является данная статья, внешний ключ |
| like | boolean | Флаг, указывающий, какую оценку пользователь поставил переводу (положительную, отрицательную, не поставил оценку) |
| created\_at | timestamp without timezone | Дата и время создания статьи |
| deleted\_at | timestamp without timezone | Дата и время удаления статьи |

Описание столбцов таблицы Report\_reasons представлено в таблице 2.11.

Таблица 2.11 – Описание таблицы Report reasons

|  |  |  |
| --- | --- | --- |
| Название столбца | Тип данных | Описание |
| id | integer | Идентификатор причины, первичный ключ |
| text | varchar | Текст причины |
| order\_position | integer | Положение причины в списке при сортировке |

Описание столбцов таблицы Reports представлено в таблице 2.12.

Таблица 2.12 – Описание таблицы Reports

|  |  |  |
| --- | --- | --- |
| Название столбца | Тип данных | Описание |
| id | uuid | Идентификатор жалобы, первичный ключ |
| text | varchar(1024) | Текст жалобы |
| article\_id | uuid | Идентификатор статьи, на которую была оставлена жалоба, внешний ключ |
| status | enum reportstatus | Статус жалобы (открыта, закрыта пользователем, отклонена, удовлетворена) |
| closed\_by\_user\_id | uuid | Идентификатор пользователя, закрывшего жалобу (пользователь, которому принадлежит статья или модератор), внешний ключ |
| reason\_id | int | Идентификатор причины, по которой была оставлена жалоба, внешний ключ |
| created\_at | timestamp without timezone | Дата и время создания жалобы |
| closed\_at | timestamp without timezone | Дата и время закрытия жалобы |

Описание столбцов таблицы Report\_comments представлено в таблице 2.13.

Таблица 2.13 – Описание таблицы Report\_comments

|  |  |  |
| --- | --- | --- |
| Название столбца | Тип данных | Описание |
| id | uuid | Идентификатор комментария, первичный ключ |
| text | varchar(100) | Текст комментария |
| sender\_id | uuid | Идентификатор пользователя, оставившего комментарий, внешний ключ |
| report\_id | uuid | Идентификатор жалобы, к которой был оставлен комментарий, внешний ключ |
| created\_at | timestamp without timezone | Дата и время создания комментария |

Описание столбцов таблицы Style\_prompts представлено в таблице 2.14.

Таблица 2.14 – Описание таблицы Style\_prompts

|  |  |  |
| --- | --- | --- |
| Название столбца | Тип данных | Описание |
| id | integer | Идентификатор запроса, первичный ключ |
| title | varchar(20) | Название запроса |
| text | varchar(200) | Текст запроса |
| created\_at | timestamp without timezone | Дата и время создания запроса |
| deleted\_at | timestamp without timezone | Дата и время удаления запроса |

Описание столбцов таблицы AI\_Models представлено в таблице 2.15.

Таблица 2.15 – Описание таблицы AI\_Models

|  |  |  |
| --- | --- | --- |
| Название столбца | Тип данных | Описание |
| id | integer | Идентификатор модели, первичный ключ |
| title | varchar(20) | Название модели |
| text | varchar(20) | Поставщик модели |
| created\_at | timestamp without timezone | Дата и время создания записи о модели |
| deleted\_at | timestamp without timezone | Дата и время удаления записи о модели |

Описание столбцов таблицы Configs представлено в таблице 2.16.

Таблица 2.16 – Описание таблицы Configs

|  |  |  |
| --- | --- | --- |
| Название столбца | Тип данных | Описание |
| id | integer | Идентификатор конфигурации, первичный ключ |
| name | varchar(20) | Название конфигурации |
| user\_id | uuid | Идентификатор пользователя, создавшего конфигурацию, внешний ключ |
| prompt\_id | integer | Идентификатор запроса перевода, внешний ключ |
| language\_ids | integer[] | Идентификаторы языков перевода |
| model\_id | integer | Идентификатор модели перевода, внешний ключ |
| created\_at | timestamp without timezone | Дата и время создания конфигурации |
| deleted\_at | timestamp without timezone | Дата и время удаления конфигурации |

Описание столбцов таблицы Translation\_tasks представлено в таблице 2.17.

Таблица 2.17 – Описание таблицы Translation\_tasks

|  |  |  |
| --- | --- | --- |
| Название столбца | Тип данных | Описание |
| id | uuid | Идентификатор задачи, первичный ключ |
| article\_id | uuid | Идентификатор исходной статьи, внешний ключ |
| source\_language\_id | integer | Идентификатор исходного языка, внешний ключ |
| target\_language\_id | integer | Идентификатор конечного языка, внешний ключ |
| prompt\_id | integer | Идентификатор запроса перевода, внешний ключ |
| model\_id | integer | Идентификатор модели перевода, внешний ключ |
| status | enum translationtaskstatus | Статус задачи (создана, в процессе выполнения, завершена успешно, завершена с ошибкой) |
| data | jsonb | Дополнительная информация о задаче (текст ошибки) |
| translated\_article\_id | uuid | Идентификатор переведённой статьи, внешний ключ |
| created\_at | timestamp without timezone | Дата и время создания задачи |
| deleted\_at | timestamp without timezone | Дата и время удаления задачи |

Описание столбцов таблицы Notifications представлено в таблице 2.18.

Таблица 2.18 – Описание таблицы Notifications

|  |  |  |
| --- | --- | --- |
| Название столбца | Тип данных | Описание |
| id | uuid | Идентификатор уведомления, первичный ключ |
| title | varchar | Заголовок уведомления |
| text | varchar | Текст уведомления |
| user\_id | uuid | Идентификатор пользователя, которому предназначено уведомление, внешний ключ |
| type | enum notificationtype | Тип уведомления (информационное, предупреждение, ошибка) |
| created\_at | timestamp without timezone | Дата и время создания записи о модели |
| read\_at | timestamp without timezone | Дата и время удаления записи о модели |

Назначение связей приведено в таблице 2.19.

Таблица 2.18 – Описание таблицы Notifications

|  |  |
| --- | --- |
| Связь | Назначение |
| Users.id-Notifications.user\_id | Идентификатор пользователя, которому адресовано уведомление |
| Users.id-Confirmation\_codes.user\_id | Идентификатор пользователя, которому предназначен код подтверждения |
| Users.id-Sessions.user\_id | Идентификатор пользователя, который создал сессию |
| Users.id-Articles.user\_id | Идентификатор пользователя, который загрузил статью или запустил перевод исходной статьи |
| Users.id-Reports.closed\_by\_user\_id | Идентификатор пользователя, закрывшего жалобу (создавшего её пользователя или любого модератора) |
| Report\_reasons.id-Reports.reason\_id | Идентификатор причины, по которой была создана жалоба на перевод статьи |
| Articles.id-Articles.original\_article\_id | Идентификатор исходной статьи, из которой был создан перевод |
| Articles.id-Translation\_tasks.article\_id | Идентификатор статьи, которую необходимо перевести |
| Articles.id-Translation\_tasks.translated\_article\_id | Идентификатор перевода статьи |
| Articles.id-Reports.article\_id | Идентификатор перевода, на который была создана жалоба |
| Languages.id-Articles.language\_id | Идентификатор языка статьи |
| Languages.id-Translation\_tasks .source\_language\_id | Идентификатор исходного языка, с которого необходимо перевести статью |
| Languages.id-Translation\_tasks .target\_language\_id | Идентификатор конечного языка, на который необходимо перевести статью |
| Reports.id-Comments.report\_id | Идентификатор жалобы, под которой был оставлен комментарий |
| AI\_Models.id- Translation\_tasks.model\_id | Идентификатор записи о модели искусственного интеллекта, которая используется для перевода статьи |
| AI\_Models.id- Configs.model\_id | Идентификатор записи о модели искусственного интеллекта |
| Style\_prompts.id- Translation\_tasks.prompt\_id | Идентификатор запроса перевода, который используется для перевода статьи |
| Style\_prompts.id- Configs.prompt\_id | Идентификатор запроса перевода |

Сценарий создания базы данных на языке SQL представлен в Приложении А.

2.3 Архитектура web-приложения

Архитектура web-приложения представлена на рисунке 2.3.

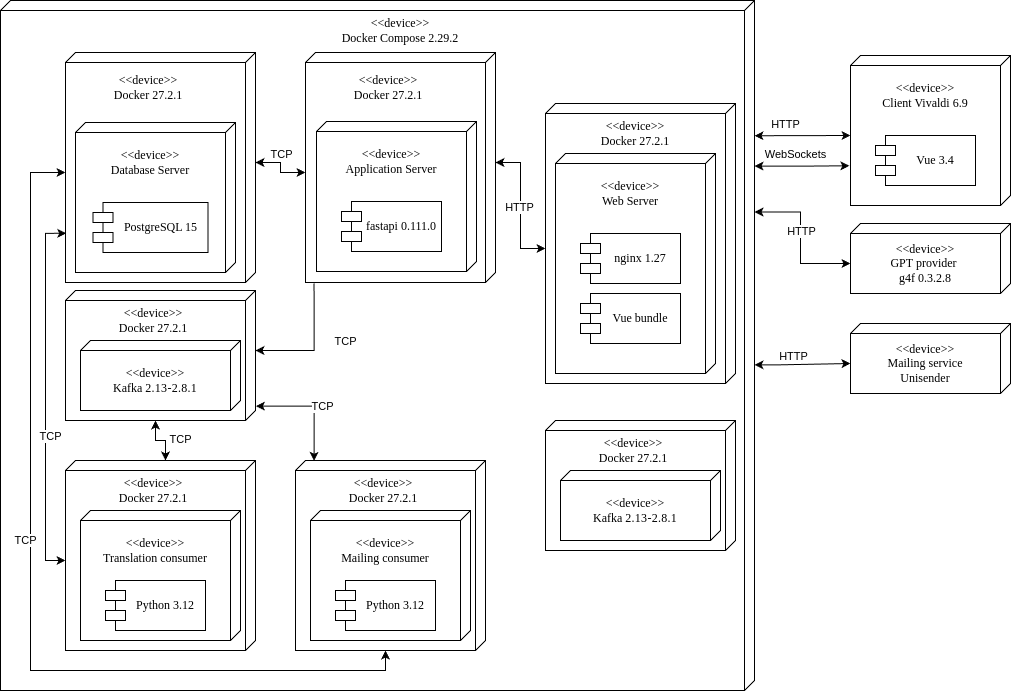


Рисунок 2.3 – Архитектура web-приложения

Пояснение назначения каждого элемента web-приложения представлено в таблице 2.19.

Таблица 2.19 – Назначение элементов архитектурной схемы web-приложения

|  |  |
| --- | --- |
| Элемент | Назначение |
| Web Server (nginx) | Принимать запросы клиента, предоставлять соединение по протоколу HTTPS, предоставлять доступ к статическим файлам фронтэнд-части web-приложения |
| Database Server (PostgreSQL) | Хранить данные, которые должны храниться длительное время |
| Kafka | Принимать и направлять сообщения нужным потребителям сообщений |
| Application Server | Обрабатывать запросы пользователя, запрашивать данные из базы данных, отправлять сообщения в очереди сообщений |
| Translation consumer | Переводить статьи при помощи внешнего сервиса |
| Mailing consumer | Отправлять электронные письма при помощи внешнего сервиса |
| Redis | Хранить данные с маленьким сроком жизни |
| GPT provider | Переводить тексты по запросу |
| Mailing service | Отправлять электронные письма по запросу |
| Client (Vivaldi) | Отображать фронтэнд-часть web-приложения, отправлять запросы пользователя, отображать ответы сервера |

Таким образом, web-приложение состоит из различных компонентов, каждый из которых выполняет собственные функции.

3 Реализация web-приложения

3.1 Реализация бэкэнда

Для разработки бэкэнда был использован фреймворк h3. Согласно [2], в нём для обработки запросов могут применяться простые обработчики запросов. Обработчики запросов, добавляющие к ответу заголовки CORS и возвращающие html-страницу на любой запрос, представлены в листинге 3.1.

|  |
| --- |
| app.use(defineEventHandler(event => {  appendHeaders(event, {  'Access-Control-Allow-Origin': '\*',  'Access-Control-Expose-Headers': '\*'  })  }));  app.use(defineEventHandler(async event => {  setResponseHeader(event, 'Content-Type', 'text/html');  return fs.readFileSync('./views/react-front/dist/index.html');  })); |

Листинг 3.1 – Простые обработчики запросов

Также для обработки запросов могут использоваться более продвинутые роутеры. Фрагмент

3.2 Реализация фронтэнда

Для разработ

4 Тестирование web-приложения

Для

5 Руководство пользователя

При

Заключение

При выполнении курсового проекта было создано приложение поиска и предложения работы. Сервер был создан при помощи платформы Node.js, языка программирования JavaScript и фреймворка h3. Web-сайт был реализован при помощи библиотеки React. База данных была реализована в СУБД PostgreSQL. Были реализованы все функциональные требования, а именно:

– обеспечение возможности регистрации и авторизации;

– поддержка ролей гостя, соискателя, работодателя и администратора;

– обеспечение возможности изменять образование, опыт работы и список ключевых навыков соискателя;

– предоставление возможности работодателю принимать и отклонять отклики;

− обеспечение возможности оставлять отзывы об исполнителе и работодателе;

− предоставление возможности отслеживать статус предложения о работе в реальном времени;

− обеспечение возможности соискателю откликаться на предложения о работе;

− предоставление возможности создавать и удалять учётные записи;

− обеспечение возможности фильтровать предложения по критериям (тип, необходимый опыт, оплата и так далее);

− предоставление возможности размещать и удалять вакансии и заказы на выполнение работ.

Также были реализованы WebSocket-сервер и поддержка протокола HTTPS. Приложение было протестировано на наличие ошибок с использованием ручного и автоматического тестирования. Для ручного тестирования использовался браузер и Postman. Для автоматического тестирования использовалось вспомогательное приложение, использующее функцию fetch.

По итогам тестирования были исправлены следующие ошибки в приложении: неверный формат отправляемых данных, отсутствие необходимых данных в ответе, ошибки в проверке авторизации.

Также было создано иллюстрированное руководство пользователя, в котором были описаны способы взаимодействия с приложением и наглядно продемонстрированы web-страницы приложения.

Список используемых источников

1. PostgreSQL Documentation [Электронный ресурс]. – Электронные данные. – Режим доступа: <https://www.postgresql.org/docs/>.
2. h3 – The Web Framework for Modern JavaScript Era [Электронный ресурс]. – Электронные данные. – Режим доступа: https://h3.unjs.io.
3. Enabling HTTPS on express.js [Электронный ресурс]. – Электронные данные. – Режим доступа: https://stackoverflow.com/questions/11744975/enabling-https-on-express-js.
4. Material UI components [Электронный ресурс]. – Электронные данные. – Режим доступа: https://mui.com/material-ui/all-components/.

Приложение А

Ли