

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий
Кафедра Информационных систем и технологий
Специальность 1-40 01 01 «Программное обеспечение информационных технологий»
Специализация 1-40 01 01 10 «Программное обеспечение информационных технологий (программирование интернет-приложений)»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту на тему:**

Web-приложение «Изучение английского языка»

Выполнил студент Белашков Василий Александрович
(Ф.И.О.)

Руководитель проекта ст. преп. Дубовик М. В.
(учен. степень, звание, должность, подпись, Ф.И.О.)

Заведующий кафедрой к.т.н., доц. Смелов В.В.
(учен. степень, звание, должность, подпись, Ф.И.О.)

Консультанты ст. преп. Дубовик М. В.
(учен. степень, звание, должность, подпись, Ф.И.О.)

Норм контролёр ст. преп. Дубовик М. В.
(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой _____

Содержание

Введение	3
1. Постановка задачи	4
1.1 Аналитический обзор литературы	4
1.2 Анализ прототипов	6
1.3 Функциональные требования	7
2. Проектирование приложения	9
2.1 Проектирование базы данных	9
2.2 Идентификация и авторизация	15
2.3 Диаграмма развёртывания	15
3. Разработка приложения.....	16
3.1 Реализация базы данных	16
3.2 Реализация серверной части.	16
3.3 Серверная и клиентская часть веб-сокетов.....	18
3.4 Реализация документации.....	19
3.5 Проектирование клиентской части приложения	20
3.6 Конечные точки HTTP API.....	21
4. Тестирование	23
4.1 Страница словари.....	23
4.2 Страница уроки	24
5. Руководство пользователя	26
5.1 Регистрация и авторизация	26
5.2 Статистика	27
5.3 Страница изменения словарей и слов.....	28
5.4 Страница просмотра словарей и слов.....	29
5.5 Страница изучения слов.....	29
5.6 Страница уроков	30
5.7 Страница изменения уроков и заданий	31
6. Заключение	33
7. Список использованной литературы	34
Приложение А	35
Приложение Б.....	39
Приложение В	41

Введение

В современном мире знание английского языка является важным условием для успешной карьеры, обучения за границей, путешествий и общения с иностранными друзьями и коллегами. Однако, не всегда у нас есть возможность посещать курсы и занятия по изучению английского языка в оффлайн формате. В таком случае, web-приложения для изучения английского языка становятся наиболее удобным и доступным решением.

Целью данного курсового проекта является разработка web-приложения для изучения английского языка. Оно будет позволять пользователям улучшать свои навыки и знания в любое удобное время и в любом месте. В рамках проекта будет рассмотрено несколько аспектов, необходимых для создания и успешного функционирования web-приложения.

Первым этапом будет разработка концепции приложения и его функциональных требований. Здесь мы определим основные цели и задачи, которые должно решать приложение, а также его целевую аудиторию. Кроме того, необходимо обозначить функции и возможности, которые будут включены в приложение.

Вторым этапом является проектирование приложения, включающее в себя разработку структуры базы данных, взаимодействия пользователя с интерфейсом приложения и создание системы обработки и хранения данных. Также здесь определяется дизайн приложения и разработка интерфейсов, которые будут максимально удобны и понятны для пользователей.

Третий этап – программирование и тестирование приложения. Здесь будет создан код, который обеспечит функциональность и работу приложения, а также протестирован на соответствие функциональным требованиям.

Кроме того, приложение будет содержать возможности для ведения статистики прогресса пользователя. Это позволит пользователю отслеживать свой прогресс.

Таким образом, данное web-приложение будет полезным и удобным инструментом для изучения английского языка, доступным в любое время и в любом месте. Разработка и успешное функционирование приложения потребует учета многих аспектов и внимательной работы над каждым этапом процесса разработки. Однако, благодаря правильно выбранной концепции, дизайну и функциональности, приложение сможет помочь пользователям в достижении своих целей в области изучения английского языка.

1. Постановка задачи

По заданию курсового проекта необходимо разработать web-приложение, позволяющее администратору и пользователю авторизоваться; администратору добавлять, изменять и удалять словари, слова в словарях, уроки и задания в уроках; пользователю добавлять, изменять и удалять словари и слова в них, изучать слова в словарях администратора, изучать уроки и задания добавленные администратором, изучать слова в соревновательном режиме, а именно в виде квиза.

1.1 Аналитический обзор литературы

Одним из ключевых моментов в разработке программного обеспечения является изучение аналогов, выявление достоинств и недостатков в них. С помощью анализа аналогов можно выделить функционал, который обязательно должен присутствовать в web-приложении. Это необходимо для построения каркаса будущего приложения. Также анализ помогает выделить недостатки и избежать их в собственной реализации.

DUOLINGO.COM

Данное web-приложение имеет следующие возможности:

- изучение грамматических правил и слов по определенным темам в виде игрового процесса;
- тематические диалоги, которые позволяют приблизиться к жизненным ситуациям;
- личная статистика прогресса и статистика друзей;
- система достижений и бонусов за получение этих достижений.

Интерфейс web-приложения представлен на рисунке 1.1.

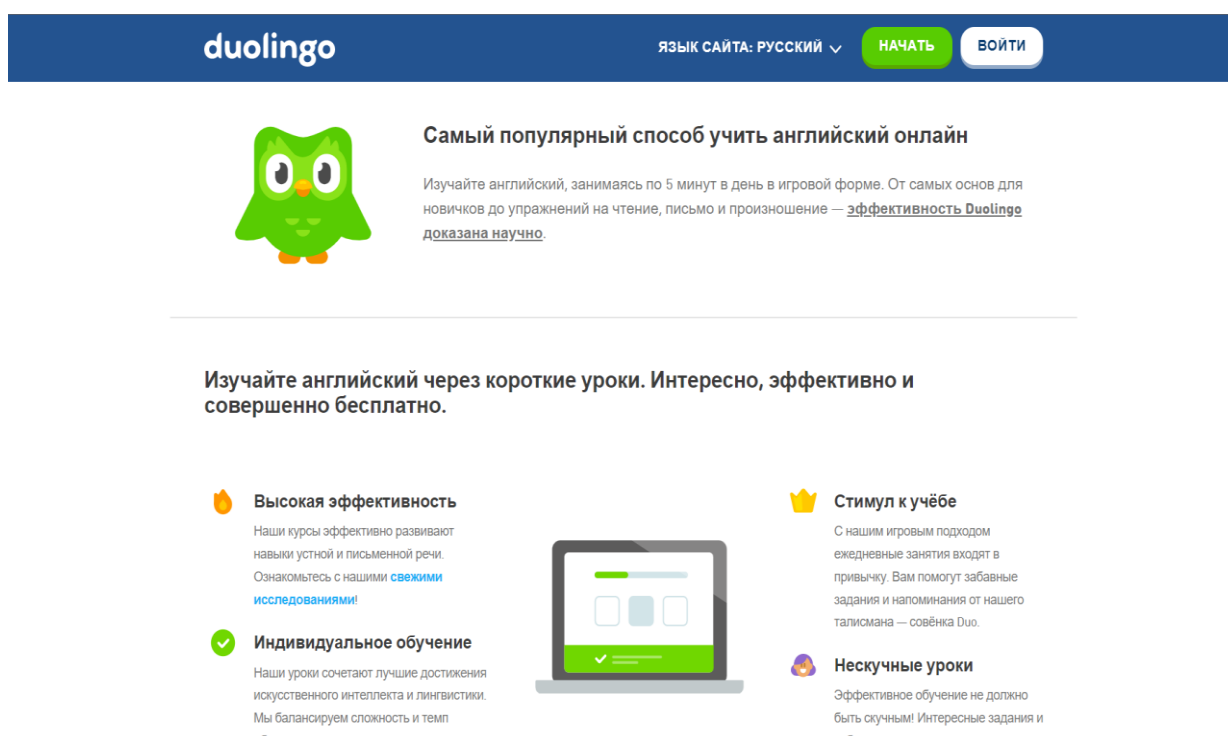


Рисунок 1.1 – Web-приложение duolingo.com

Из недостатков необходимо отметить то, что имеющийся контент позволяет получить только начальные навыки овладения языком. Также невозможно настроить приложение под свои навыки. У приложения есть и плюсы, например, за различные достижения пользователю вручаются награды (за количество выученных слов, за ежедневные посещения). Отслеживая статистику других пользователей, повышается мотивация изучения и прохождения уровней дальше.

POLIGLOT16.RU

Приложение имеет следующие возможности:

- практические и тестовые задания по грамматике;
- изучение слов по тематическим словарям;
- добавление собственных словарей;
- настройка режимов обучения.

Интерфейс web-приложения представлен на рисунке 1.2.

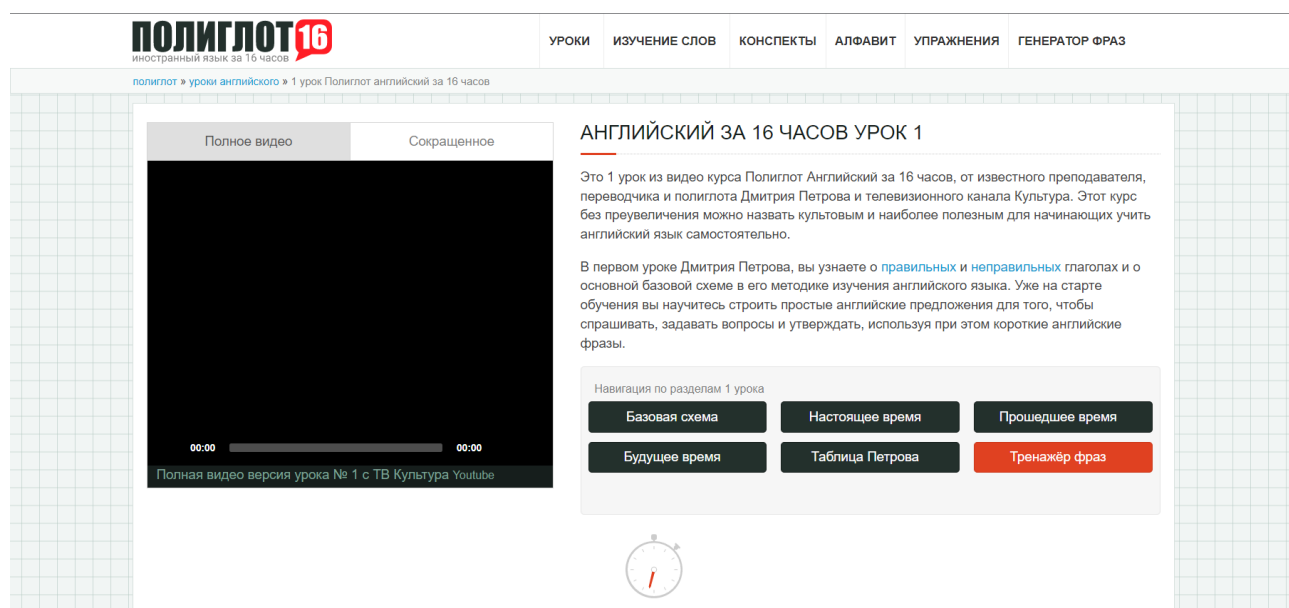


Рисунок 1.2 – Web-приложение poliglot16.ru

Довольно устаревший дизайн и не обновляемая учебная база, что делает web-приложение быстро выходящим из актуальности.

PUZZLE-ENGLISH.COM

Данное приложение так же было разработано под мобильные операционные системы IOS и Android. Имеет следующие возможности:

- уроки по грамматическим правилам и расширение словарного запаса;
- конспект с грамматическими правилами;
- тематические истории в процессе прохождения, которых хорошо запоминаются слова за счёт образования ассоциаций;
- видео-уроки с текстовым сопровождением;
- возможность установить напоминание о занятии на любое время.

Интерфейс web-приложения представлен на рисунке 1.3.

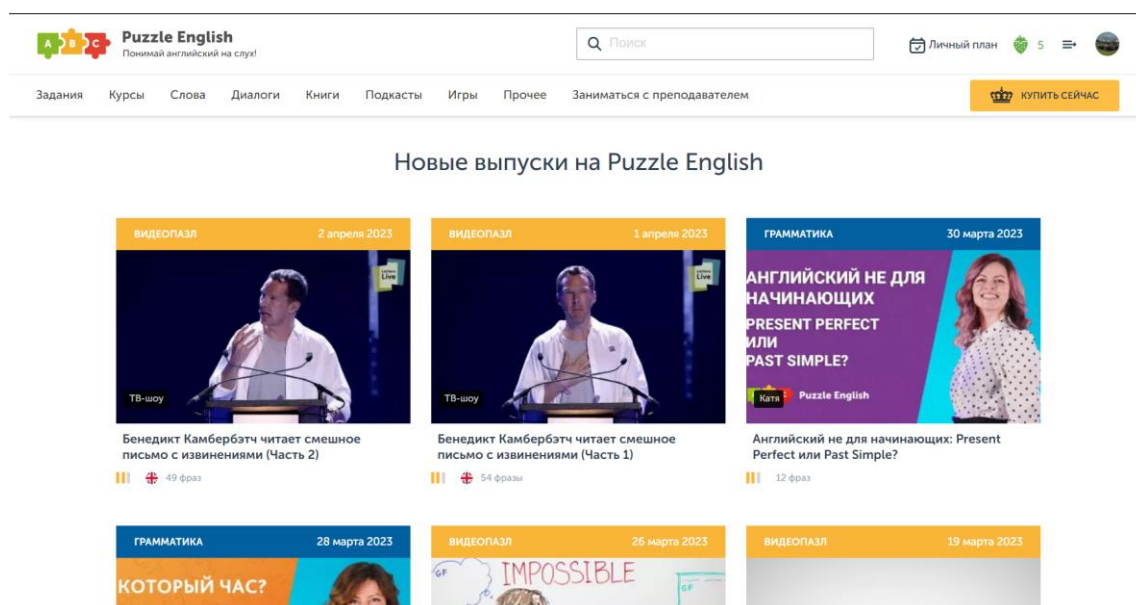


Рисунок 1.3 – Web-приложение puzzle-english.com

К недостаткам данного web-приложения можно отнести:

- основная часть контента является платной;
- невозможность изучать грамматику и лексику отдельно;
- довольно короткие уроки, за которые сложно усвоить тему.

1.2 Анализ прототипов

Схожесть рассмотренных программных средств заключается в том, что все они имеют как грамматические задания, так и задания на расширение лексикона. Также эти приложения содержат контент, который позволяет получить лишь начальные знания языка.

Главные отличия программ кроются в их функциональности. Так, например, в приложениях Duolingo и Puzzle-English изучение грамматики и словарного запаса проходит в модульной форме, в отличие от приложения Poliglot16. Только в приложении Duolingo имеется статистика прогресса, в приложении Poliglot16 возможность добавлять собственные словари, а в приложении Puzzle-English есть грамматические конспекты. Также приложение Puzzle-English основную часть своего контента предлагает на платной основе.

Таким образом, можно сделать вывод о том, что в каждом приложении есть свои плюсы и минусы. Но, разумеется, каждому разработчику хочется, чтобы его приложение не имело отрицательных качеств, так как это сильно влияет рейтинг приложения, от чего напрямую зависит прибыль и статус самого разработчика.

Анализируя ранее приведённые примеры, можно составить основные функциональные особенности для подобного типа приложений:

- регистрация и авторизация пользователя;
- тренировка грамматики и расширение словарного запаса;
- база грамматических заданий и словарей;
- настройки обучения;
- статистика пользователя.

Как сказано выше, программа должна быть предназначена для различной аудитории пользователей. Это значит, что приложение должно быть простое и иметь доступный дизайн.

Все эти пункты и были учтены при выполнении данного курсового проектирования.

1.3 Функциональные требования

Приложение должно иметь разделение ролей. Для каждой роли были разработаны UML-диаграммы вариантов использования. Роль каждой учётной записи определена в базе данных. Создать администратора на сайте невозможно. Для каждой из ролей предопределены доступные страницы, поэтому при попытке получить доступ к ресурсу, которого нет или недостаточно прав, пользователь будет перенаправлен на страницу с оповещением об отсутствии искомого ресурса.

Войдя в аккаунт, пользователь получает роль авторизованного пользователя. Диаграмма вариантов использования для этой роли отображена на рисунке 1.4.



Рисунок 1.4 – Диаграмма использования авторизованного пользователя

В приложении есть роль администратора. Для этой роли диаграмма вариантов использования отображена на рисунке 1.5.

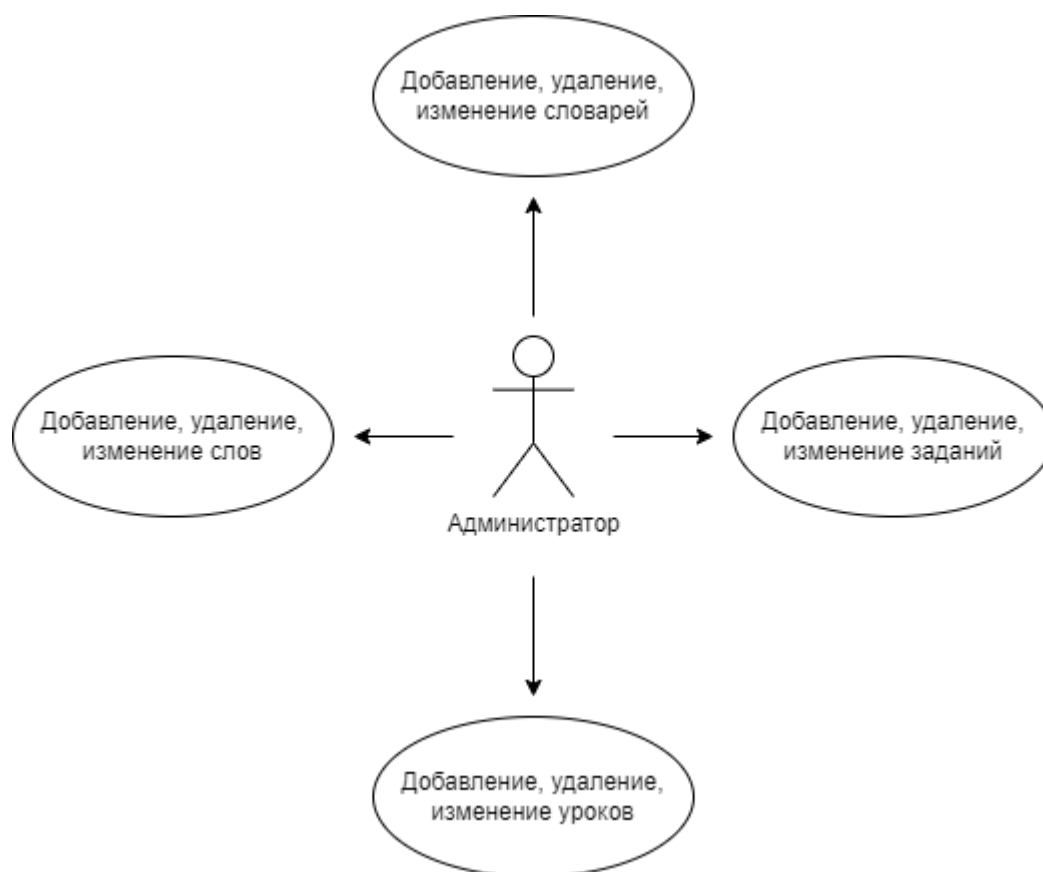


Рисунок 1.5 – Диаграмма использования для работника университета

На основе данных UML-диаграмм и словесного описания функциональных требований строится вся бизнес логика программы.

Таблица dictionaries содержит словари. Перечень полей таблицы dictionaries приведен в таблице 2.1.

Таблица 2.1 – Описание полей таблицы dictionaries

Поле	Типы данных, ограничения целостности
id	Int, identity, primary key
name	varchar(30), not null
description	varchar(255)
creator_id	Int, not null

Описание каждого поля таблицы university_data:

- id – отвечает за идентификацию словаря;
- name – название словаря;
- description – краткое описание словаря;
- creator_id – идентификатор создателя словаря.

Таблица dictionary_to_word служит для связи таблиц dictionaries и words отношением многие ко многим.

Перечень полей таблицы dictionary_to_word приведен в таблице 2.2.

Таблица 2.2 – Описание полей таблицы dictionary_to_word

Поле	Типы данных, ограничения целостности
dictionary_id	Int, not null, foreign key
word_id	Int, not null, foreign key

Описание каждого поля таблицы dictionary_to_word:

- dictionary_id – внешний ключ, ссылающийся на идентификатор (id) таблицы dictionaries;
- word_id – внешний ключ, ссылающийся на идентификатор (id) таблицы words.

Таблица grammar_progresses служит для связи таблиц users и tasks отношением многие ко многим и каждая запись в таблице информирует о том, что определённый пользователь успешно выполнил определённое задание.

Перечень полей таблицы grammar_progresses приведен в таблице 2.3.

Таблица 2.3 – Описание полей таблицы grammar_progresses

Поле	Типы данных, ограничения целостности
id	Int, identity, primary key
task_id	Int, not null, foreign key
user_id	Int, not null, foreign key

Описание каждого поля таблицы grammar_progresses:

- id – идентификатор строки, является первичным ключом таблицы;
- task_id – внешний ключ, ссылающийся на идентификатор (id) таблицы tasks;
- user_id – внешний ключ, ссылающийся на идентификатор (id) таблицы users.

Таблица lessons хранит информацию о каждом уроке и имеет связь с таблицей tasks один ко многим, так как в одном уроке может быть несколько заданий.

Перечень полей таблицы lessons приведен в таблице 2.4.

Таблица 2.4 – Описание полей таблицы lessons

Поле	Типы данных, ограничения целостности
id	Int, identity, primary key
name	varchar(30), not null
theory	varchar(max), not null
passing_percent	Int, not null

Описание каждого поля таблицы lessons:

- id – идентификатор строки, является первичным ключом таблицы;
- name – название урока;
- theory – теоретические сведения по теме урока;
- passing_percent – минимальный процент правильно выполненных заданий урока, для того, чтобы тема урока считалась достаточно изученной.

Таблица lexicon_progresses служит для связи таблиц users и words отношением многие ко многим и каждая запись в таблице хранит информацию о прогрессе пользователя в изучении слов.

Перечень полей таблицы lexicon_progresses приведен в таблице 2.5.

Таблица 2.5 – Описание полей таблицы lexicon_progresses

Поле	Типы данных, ограничения целостности
id	Int, identity, primary key
progress_count	Int, not null
is_learned	Boolean, not null
user_id	Int, not null, foreign key
word_id	Int, not null, foreign key

Описание каждого поля таблицы lexicon_progresses:

- id – идентификатор строки, является первичным ключом таблицы;
- progress_count – количество правильных переводов слова пользователем, сделанных подряд;
- is_learned – указывает, выучено слово (true) или нет (false);
- user_id – внешний ключ, ссылающийся на идентификатор (id) таблицы users;
- word_id – внешний ключ, ссылающийся на идентификатор (id) таблицы words.

Таблица persmissions хранит разрешения ролей на выполнения определённых запросов.

Перечень полей таблицы persmissions приведен в таблице 2.6.

Таблица 2.6 – Описание полей таблицы persmissions

Поле	Типы данных, ограничения целостности
id	Int, identity, primary key

Окончание таблицы 2.6

Поле	Типы данных, ограничения целостности
descriptor	Int, not null
context	varchar(255)
method	varchar(255), not null
role_id	Int, not null, foreign key

Описание каждого поля таблицы permissions:

- id – идентификатор строки, является первичным ключом таблицы;
- descriptor – наименование контроллера;
- context – контекст выполняемого роута;
- method – метод http запроса;
- role_id – внешний ключ, ссылающийся на идентификатор (id) таблицы roles.

Таблица roles хранит данные о каждой роли.

Перечень полей таблицы roles приведен в таблице 2.7.

Таблица 2.7 – Описание полей таблицы roles

Поле	Типы данных, ограничения целостности
id	Int, identity, primary key
title	varchar(255), not null

Описание каждого поля таблицы roles:

- id – идентификатор строки, является первичным ключом таблицы;
- title – название роли.

Таблица sessions хранит данные о сессиях пользователей и связан с таблицей users связью один к одному.

Перечень полей таблицы sessions приведен в таблице 2.8.

Таблица 2.8 – Описание полей таблицы sessions

Поле	Типы данных, ограничения целостности
user_id	Int, unique, foreign key
refresh_token	varchar(max), not null, foreign key

Описание каждого поля таблицы sessions:

- user_id – внешний ключ, ссылающийся на идентификатор (id) таблицы users;
- refresh_token – рефреш токен пользователя.

Таблица statistics хранит статистические данные об обучении пользователей за каждый день, а именно количество правильных переводов слов, сделанных заданий и квиз очков.

Перечень полей таблицы statistics приведен в таблице 2.9.

Таблица 2.9 – Описание полей таблицы statistics

Поле	Типы данных, ограничения целостности
id	Int, identity, primary key
date	Date, not null
words	int

Окончание таблицы 2.9

Поле	Типы данных, ограничения целостности
tasks	int
quiz_points	int
user_id	Int, not null, foreign key

Описание каждого поля таблицы statistics:

- id – идентификатор строки, является первичным ключом таблицы;
 - date – дата создания записи;
 - words – количество правильных переводов слова пользователем;
 - tasks – количество правильно сделанных заданий пользователем;
 - quiz_points – количество квиз очков, полученных пользователем за правильные переводы слов в квизе;
 - user_id – внешний ключ, ссылающийся на идентификатор (id) таблицы roles.
- Таблица tasks хранит задания уроков.

Перечень полей таблицы tasks приведен в таблице 2.10.

Таблица 2.10 – Описание полей таблицы tasks

Поле	Типы данных, ограничения целостности
id	int unique foreign key
english_sentence	varchar(300) not null
russian_sentence	varchar(300) not null
lesson_id	int not null foreign key

Описание каждого поля таблицы tasks:

- id – идентификатор строки, является первичным ключом таблицы;
- english_sentence – задание, написанное на английском языке;
- russian_sentence – перевод задания из поля english_sentence на русский язык;
- lesson_id – внешний ключ, ссылающийся на идентификатор (id) таблицы lesson.

Таблица users хранит данные пользователей.

Перечень полей таблицы users приведен в таблице 2.11.

Таблица 2.11 – Описание полей таблицы users

Поле	Типы данных, ограничения целостности
id	Int, unique, primary key
name	varchar(255), not null
email	varchar(255), not null
password	varchar(255), not null
role_id	Int, not null, foreign key
id_enable_lesson	Int, not null

Описание каждого поля таблицы users:

- id – идентификатор строки, является первичным ключом таблицы;
- name – имя пользователя;
- email – электронная почта пользователя;

- password – захешированный пароль пользователя;
- role_id – внешний ключ, ссылающийся на идентификатор (id) таблицы roles;
- id_enable_lesson – идентификатор (id) урока, доступного пользователю.

Таблица user_settings хранит настройки обучения пользователей.

Перечень полей таблицы user_settings приведен в таблице 2.12.

Таблица 2.12 – Описание полей таблицы user_settings

Поле	Типы данных, ограничения целостности
user_id	int unique foreign key
count_repeat_word_for_learned	int not null
count_learning_words_simultaneously	int not null
mode_learning_words	varchar(255) not null
mode_learning_tasks	varchar(255) not null

Описание каждого поля таблицы user_settings:

- user_id – внешний ключ, ссылающийся на идентификатор (id) таблицы users;
- count_repeat_word_for_learned – количество правильных ответов подряд, чтобы слово считалось выученным;
- count_learning_words_simultaneously – количество слов изучаемы одновременно;
- mode_learning_words – способ изучения лексикона;
- mode_learning_tasks – способ изучения грамматики.

Таблица words хранит слова, добавленные администратором и пользователями.

Перечень полей таблицы words приведен в таблице 2.13

Таблица 2.13 – Описание полей таблицы words

Поле	Типы данных, ограничения целостности
id	int unique foreign key
english_spelling	int not null
transcription	int not null
russian_spelling	varchar(255) not null
description	varchar(255) not null

Описание каждого поля таблицы words:

- id – идентификатор строки, является первичным ключом таблицы;
- english_spelling – английское написание слова;
- transcription – транскрипция слова на английском;
- russian_spelling – русский перевод слова из поля english_spelling;
- description – описание, необходимое для достижения однозначного понимания слова.

2.2 Идентификация и авторизация

Для выполнения задач при разработке приложения, существует необходимость регистрации, авторизации и идентификации, для этой задачи используется JWT.

JSON Web Token (JWT) — это открытый стандарт для создания токенов доступа, основанный на формате JSON. Как правило, используется для передачи данных для аутентификации в клиент-серверных приложениях. Токены создаются сервером, подписываются секретным ключом и передаются клиенту, который в дальнейшем использует данный токен для подтверждения своей личности.

2.3 Диаграмма развёртывания

Диаграмма развёртывания — это структурная диаграмма в языке моделирования UML, которая показывает физическое размещение компонентов программной системы на различных узлах (например, компьютеры, серверы, мобильные устройства) и связи между ними. Она используется для визуализации аппаратного и программного окружения, в котором функционирует система. На рисунке 2.2 представлена диаграмма развёртывания данного web-приложения.

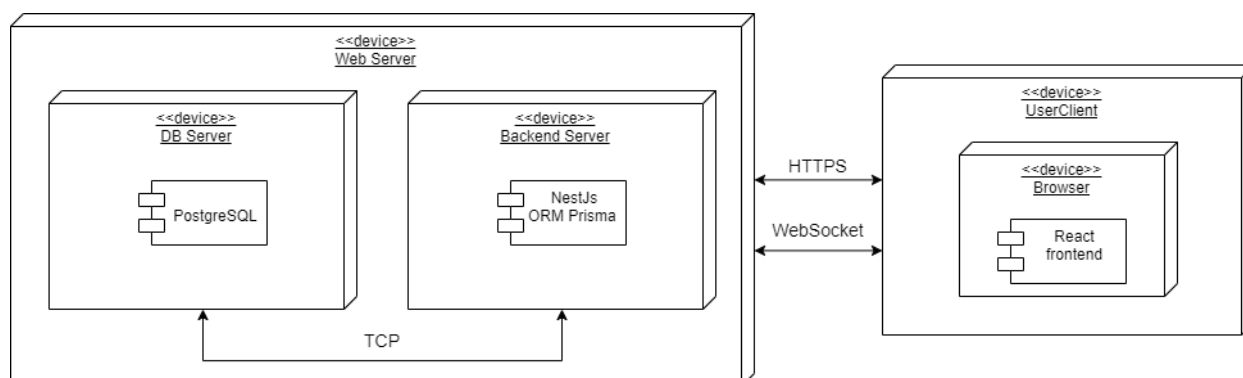


Рисунок 2.2 – Диаграмма развёртывания

На диаграмме развёртывания представлены следующие элементы:

- узлы, представляющие физические или виртуальные устройства, на которых размещаются компоненты системы (сервер, рабочая станция, мобильное устройство и т.д.);
- компоненты – это модули или части программной системы, которые могут быть развернуты на узлах (база данных, веб-сервер, клиентское приложение и т.д.);
- артефакты представляют файлы или ресурсы, необходимые для работы системы (исполняемые файлы, конфигурационные файлы, библиотеки и т.д.);
- связи показывают взаимодействия между узлами и компонентами (сетевые соединения, протоколы обмена данными и т.д.).

3. Разработка приложения

При проектировании программного средства в качестве программной платформы для сервера была использована программная платформа Node.js. На стороне клиента используется библиотека React.js.

3.1 Реализация базы данных

В программном средстве существует возможность добавления, изменения и удаления данных в базе данных. Взаимодействие приложения и базы данных происходит с помощью моделей ORM Prisma. Скрипт схемы базы данных представлен в приложении А.

3.2 Реализация серверной части.

При написании сервера на NodeJs используется фреймворк NestJs.

NestJs — это фреймворк для создания масштабируемых и модульных приложений на Node.js. Он предоставляет разработчикам инструменты для создания серверных приложений с использованием современных технологий, таких как TypeScript, инверсия управления (IoC) и внедрение зависимостей (DI).

NestJs строится на основе экосистемы Node.js и обеспечивает совместимость со многими известными инструментами и библиотеками, такими как Express, Fastify и Socket.io. Фреймворк также предоставляет разработчикам много возможностей для создания собственных модулей и компонентов.

NestJs имеет ряд инструментов и библиотек для решения общих задач, таких как аутентификация, авторизация, валидация данных и обработка ошибок. Это позволяет разработчикам создавать приложения быстрее и более эффективно, а также повышать качество их работы.

Проект разделен на несколько основных частей, которые расположены по разным директориям. На рисунке 3.1 представлена структура проекта.

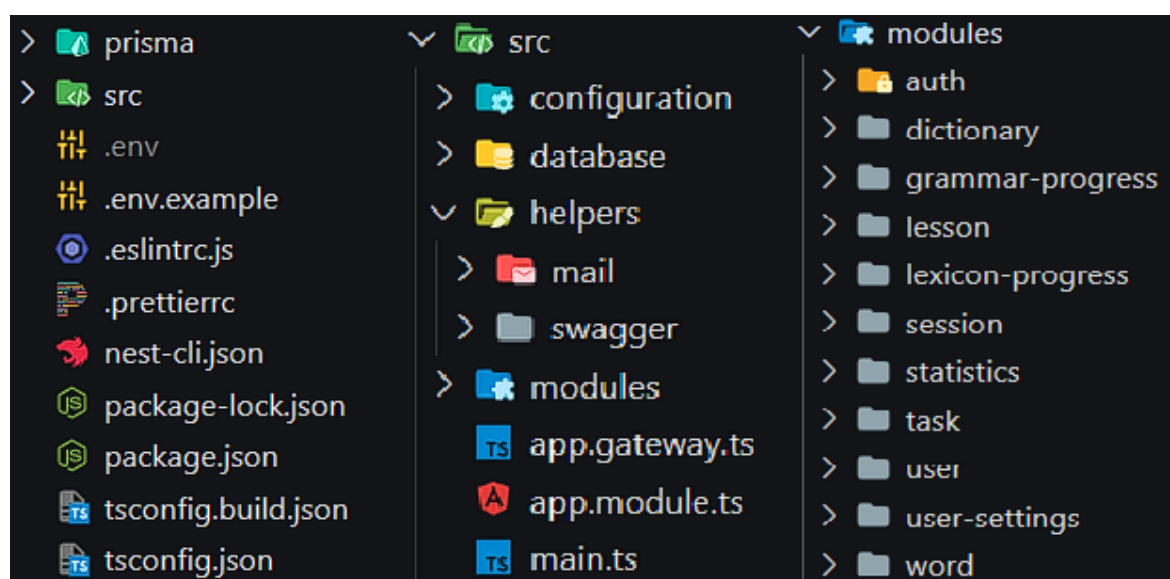


Рисунок 3.1 – Структура сервера

Основная логика сервера содержится в директории `modules`, так как в ней находятся модули для необходимых сущностей приложения. Структура всех модулей примерно одинакова и для наглядности представлена на рисунке 3.2.

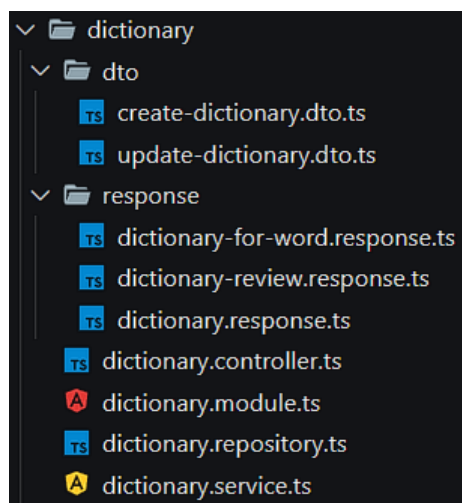


Рисунок 3.2 – Структура модуля `dictionary`

Контроллеры в NestJS — это компоненты, которые обрабатывают HTTP-запросы и отправляют HTTP-ответы клиентам. Они отвечают за управление взаимодействием между клиентской частью приложения и серверной логикой. Контроллеры в NestJS играют важную роль в разработке веб-приложений, позволяя разделить логику приложения на отдельные компоненты и обеспечивая высокую гибкость и модульность кода.

Модуль в NestJS — это класс, который использует декоратор `@Module`. В этом классе определяются провайдеры (providers), контроллеры (controllers), мидлвары (middlewares) и другие компоненты, необходимые для функционирования модуля.

Каждый ресурс в NestJS объявляется внутри модуля, используя соответствующие декораторы. Например, для создания контроллера необходимо использовать декоратор `@Controller`, а для создания провайдера - декоратор `@Injectable`.

В NestJS, сервисы — это компоненты, которые выполняют бизнес-логику и обеспечивают доступ к данным при помощи паттерна проектирования `repository`. Сервисы используются для инкапсуляции и повторного использования кода, связанного с определенной функциональностью.

Директория `helpers` содержит модули `mail` и `swagger`. Модуль `mail` служит для отправки электронных писем пользователям при регистрации. Модуль `swagger` содержит некоторую вспомогательную логику для ведения документации.

Директории `configuration` и `database`, находящиеся в директории `src`, необходимы для конфигурации всего приложения и подключения к базе данных соответственно.

Файл `package.json` — это файл конфигурации проекта. Он содержит метаданные о проекте, такие как имя проекта, версия, список зависимостей, скрипты для запуска проекта и т.д. Листинг файла представлен в приложении Б.

3.3 Серверная и клиентская часть веб-сокетов

При разработке приложения использованы веб-сокеты для изучения слов в режиме квиз.

WebSocket — протокол связи поверх TCP-соединения, предназначенный для обмена сообщениями между браузером и веб-сервером в режиме реального времени с помощью постоянного двустороннего. Он обеспечивает более эффективную и низколатентную передачу данных по сравнению с традиционными методами, такими как HTTP-запросы. В общем виде схема установки WebSocket соединения представлена на рисунке 3.3.

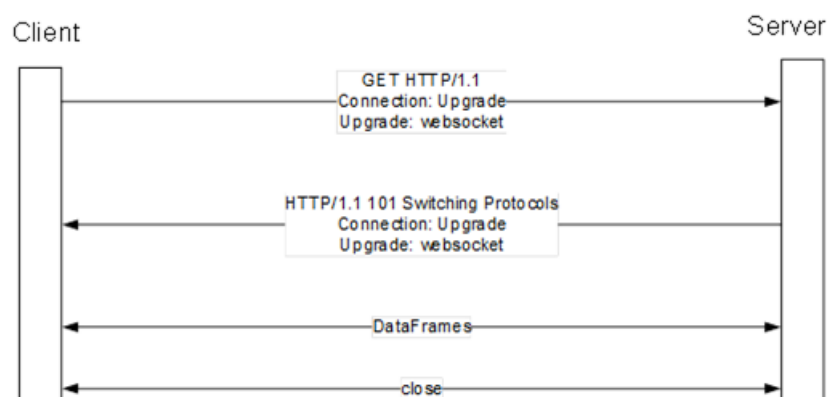


Рисунок 3.3 – Схема установки WebSocket соединения

Принцип установки WebSocket соединения можно описать следующим образом:

1) открытие соединения: браузер клиента отправляет HTTP-запрос на сервер, содержащий заголовок Upgrade со значением websocket, а также другие необходимые заголовки, такие как Connection, Host и Sec-WebSocket-Key. Этот запрос отправляется через обычное HTTP-соединение на порт сервера, обычно 80 или 443 для защищенного соединения (HTTPS);

2) подтверждение соединения: если сервер поддерживает WebSocket и может установить соединение, он отправляет ответное сообщение с кодом состояния 101 Switching Protocols и заголовками Upgrade и Connection, указывающими на переключение на протокол WebSocket;

3) установка WebSocket-соединения: после получения ответа от сервера браузер и сервер обмениваются специальными заголовками, чтобы установить полноценное WebSocket-соединение. Ключевым заголовком является Sec-WebSocket-Key, который используется для генерации уникального ключа для шифрования и дешифрования сообщений в процессе передачи;

4) обмен данными: после установки WebSocket-соединения браузер и сервер могут обмениваться данными в режиме реального времени, отправляя сообщения в формате WebSocket. Браузер и сервер могут отправлять и получать сообщения независимо друг от друга в любой момент времени без необходимости создания новых HTTP-запросов;

5) закрытие соединения: в случае необходимости закрытия соединения одна из сторон отправляет специальное сообщение с указанием кода состояния. Затем

другая сторона отправляет подтверждение закрытия соединения. После этого WebSocket-соединение полностью закрывается.

Это основные шаги установки WebSocket соединения. При использовании WebSocket-библиотек и фреймворков, таких как Socket.IO или SignalR, процесс может быть абстрагирован и упрощен для разработчика.

3.4 Реализация документации

Swagger (также известный как OpenAPI) — это набор инструментов и спецификаций для разработки, описания, документирования и использования веб-сервисов RESTful API. Swagger позволяет создавать понятную и интерактивную документацию для вашего API, которая может быть использована разработчиками и потребителями API. Swagger схема представлена на рисунке 3.4.

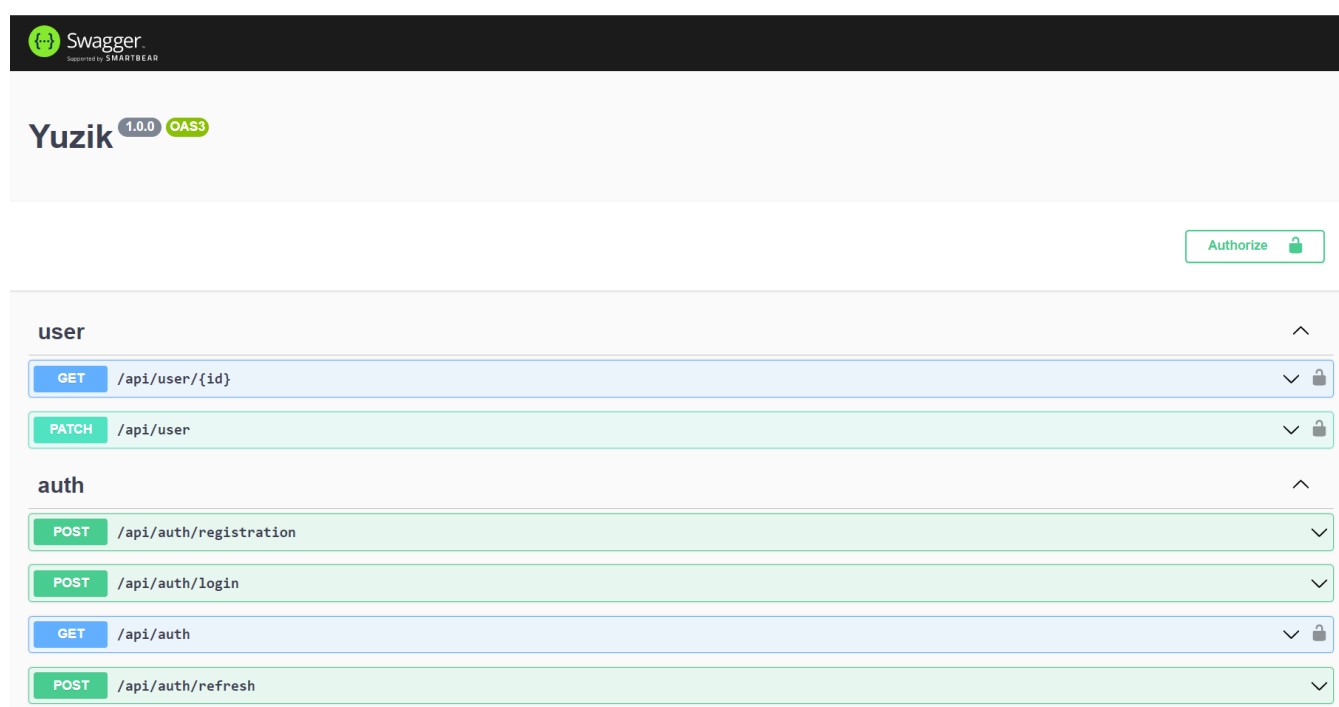


Рисунок 3.4 – Swagger схема

Реализации библиотеки swagger представлена в листинге 3.1.

```
const configService: ConfigService = app.get(ConfigService);
const PORT: number = configService.get('APP_PORT');
const HOST: string = configService.get('APP_HOST');
const config = new DocumentBuilder()
  .setTitle('Yuzik')
  .setVersion('1.0.0')
  .addBearerAuth()
  .build();
const document = SwaggerModule.createDocument(app, config);
SwaggerModule.setup('api/swagger', app, document);
```

Листинг 3.1 – Реализация библиотеки swagger

Swagger генерирует читаемую и интерактивную документацию на основе спецификации API. Это помогает разработчикам быстрее понять и использовать ваше API. Swagger спецификация служит в качестве одного источника истины для вашего API. Это помогает разработчикам, клиентам и другим заинтересованным сторонам иметь единое понимание вашего API и его возможностей. Генерация клиентского и серверного кода на основе спецификации Swagger ускоряет разработку приложений.

3.5 Проектирование клиентской части приложения

Для клиентской части приложения был использован React.js.

React — это JavaScript библиотека для создания пользовательских интерфейсов, которая позволяет разработчикам строить мощные и эффективные веб-приложения. Она использует компонентный подход, что делает код более модульным и легко поддерживаемым. Структура клиентской части приложения отображения на рисунке 3.5.

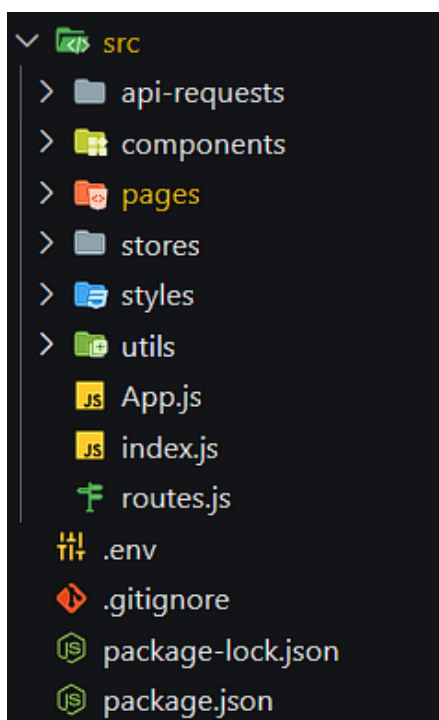


Рисунок 3.5 – Структура клиентской части приложения

Директория src имеет следующее:

- папка api-requests содержит файлы с расширением js в которых написаны запросы к серверной части приложения;
- папка components содержит React-components файлы с расширением jsx;
- папка pages содержит React-components файлы с расширением jsx, которые представляют страницы приложения;
- папка stores содержит хранилища, являющиеся централизованными контейнерами для хранения состояния приложения;
- папка styles содержит файлы с расширением css для стилизации разметки;

– папка `utils` содержит вспомогательные файлы, такие как файлы с константными значениями ошибок или регулярные выражения.

Листинг файла `package.json` представлен в приложении В.

3.6 Конечные точки HTTP API

Для упрощения взаимодействия сторонних приложений с сервисом было разработано API, содержащее необходимые методы для взаимодействия с частями приложения. Данное API используется в клиентской части приложения.

Всего разработано 11 контроллеров, каждый из которых методы, обрабатывающие входящие запросы. Для взаимодействия с частью методов необходима авторизация.

Всего в проекте 33 REST API метода.

Описание доступных методов приведено в таблице 3.1.

Таблица 3.1 – Описание методов REST API

Адрес	Метод	Описание
<code>/api/user/{id}</code>	GET	Получение данных пользователя по его id
<code>/api/user</code>	PATCH	Обновление пароля пользователя
<code>/api/auth/registration</code>	POST	Регистрация нового пользователя
<code>/api/auth/login</code>	POST	Вход в аккаунт
<code>/api/auth/refresh</code>	POST	Обновление токенов по рефреш токену
<code>/api/auth/logout</code>	POST	Выход из аккаунта
<code>/api/dictionary/</code>	POST	Создание словаря
<code>/api/dictionary/admin</code>	GET	Получение списка словарей, созданных администратором
<code>/api/dictionary/user</code>	GET	Получение списка словарей, созданных пользователем
<code>/api/dictionary/review</code>	GET	Получение списка словарей, созданных пользователем и администратором
<code>/api/dictionary/learn</code>	GET	Получение списка словарей для обучения
<code>/api/dictionary/{id}</code>	PATCH	Обновление словаря по id
<code>/api/dictionary/{id}</code>	DELETE	Удаление словаря по id
<code>/api/word/{id}</code>	POST	Создание слова
<code>/api/word/{id}</code>	PATCH	Обновление слова по id
<code>/api/word/{id}/dictionary/{dictionaryId}</code>	DELETE	Удаление слова из словаря по id слова и id словаря
<code>/api/lesson</code>	POST	Создание урока
<code>/api/lesson/admin</code>	GET	Получение всех уроков

Окончание таблицы 3.1

Адрес	Метод	Описание
/api/lesson/learn	GET	Получение доступных уроков для изучения пользователем
/api/lesson/{id}	PATCH	Обновление урока по id
/api/lesson/{id}	DELETE	Удаление урока по id
/api/grammar-progress	POST	Создание записи изученного урока
/api/task	POST	Создать задания
/api/task/{id}	PATCH	Обновление задания по id
/api/task/{id}	DELETE	Удаление задания по id
/api/statistics	POST	Создание статистики
/api/statistics/all	GET	Получение всех статистик
/api/statistics/user-all	GET	Получение всех статистик пользователя
/api/statistics/user-today	GET	Получение статистики пользователя за текущий день
/api/user-settings	GET	Получение настроек пользователя
/api/user-settings	PATCH	Обновление настроек пользователя
/api/lexicon-progress	POST	Создание записи изученного слова

Наглядно таблица REST API методов сервера может быть представлена с помощью библиотеки Swagger.

4. Тестирование

В этой главе рассмотрены и протестированы основные элементы интерфейса веб-приложения.

4.1 Страница словари

Страница состоит из таблиц словарей и слов и из форм для добавления словарей и слов. Формы добавления состоят из полей для ввода и кнопки для добавления. Интерфейс страницы представлен на рисунке 4.1.

The image shows two screenshots of a web application interface. The left screenshot displays a table titled 'Словари' (Dictionaries) with columns 'название' (name) and 'описание' (description). The first row shows 'Цвета' (Colors) with an edit icon and a delete icon. Below the table is a pagination control showing '1 to 1 of 1'. The right screenshot displays a table titled 'Слова' (Words) with columns 'английский' (English), 'транскрипция' (Transcription), 'русский' (Russian), and 'описание' (Description). The first row shows 'snow-white', 'ˈsnaʊ waɪt', 'белоснежный', and an edit icon. Below the table is a pagination control showing '6 to 6 of 6'. Below each table is a form for adding new entries. The 'Словари' form has fields for 'Название' (Name) and 'Описание' (Description, optional), and a 'Добавить' (Add) button. The 'Слова' form has fields for 'английский', 'транскрипция', 'русский', and 'описание' (optional), and a 'Добавить' (Add) button.

Рисунок 4.1 – Страница словари

При вводе данных в поля происходит проверка по определённому регулярному выражению. В случае ввода ошибочных данных и попытке добавления, под полем ввода появляется сообщение об ошибке. Результат ввода ошибочных значений в формы добавления представлен на рисунке 4.2.

The image shows two screenshots of the web application interface, focusing on the error handling in the add forms. The left screenshot shows the 'Словари' add form. The 'Название' (Name) field contains 'qqqqqqqqqqqqqqqqqqqqqqqqqqqqqq'. Below the field is a red error message: 'Некорректное название словаря' (Incorrect dictionary name). The 'Описание' (Description) field is empty, with the text 'необязательное поле' (optional field) below it. The 'Добавить' (Add) button is visible. The right screenshot shows the 'Слова' add form. The 'английский' (English) field contains 'мама' (mom). Below the field is a red error message: 'От 2 до 30 английских символов' (2 to 30 English symbols). The 'транскрипция' (Transcription) field is empty, with a red error message: 'От 2 до 50 символов' (2 to 50 symbols). The 'русский' (Russian) field contains 'mother'. Below the field is a red error message: 'От 2 до 30 русских символов' (2 to 30 Russian symbols). The 'описание' (Description) field is empty, with the text 'необязательное поле' (optional field) below it. The 'Добавить' (Add) button is visible.

Рисунок 4.2 – Результат ввода ошибочных значений на странице словарей

При попытке добавить в словарь слово, которое уже есть в этом словаре, будет выдана ошибка, представленная на рисунке 4.3.

The screenshot shows a web application for a word dictionary. At the top, there is a table titled "Слова" (Words) with columns: "английский" (English), "транскрипция" (Transcription), "русский" (Russian), and "описание" (Description). The table contains one entry: "snow-white", "'snəʊ waɪt", and "белоснежный". To the right of the table is a search bar labeled "Keyword Search". Above the table, a red error message box says "Ошибка: Вы ввели неверные данные." (Error: You entered incorrect data.). Below the table is a form titled "Форма для добавления слов" (Form for adding words) with four input fields: "английский" (containing "snow-white"), "транскрипция" (containing "'snəʊ waɪt"), "русский" (containing "белоснежный"), and "описание" (containing "необязательное поле"). A blue "Добавить" (Add) button is at the bottom of the form.

Рисунок 4.3 – Результат добавления слова в словарь второй раз

При исправлении введённых данных на корректные и нажатии на кнопку все сообщения об ошибках исчезнут.

4.2 Страница уроки

Страница состоит из таблиц уроков и заданий и из форм для добавления и изменения уроков и заданий. Формы добавления состоят из полей для ввода и кнопки для добавления. Интерфейс страницы представлен на рисунке 4.4.

The screenshot shows two side-by-side sections: "Уроки" (Lessons) and "Задания" (Tasks). The "Уроки" section has a table with columns "название" (Name) and "проходной процент" (Passing percentage). It lists three lessons: "Present Simple" (50%), "Past Simple" (60%), and "Future Simple" (45%). Below the table is a form titled "Форма для добавления уроков" (Form for adding lessons) with fields for "Название" (Name), "Проходной процент" (Passing percentage) with a slider, and "Теория к уроку" (Theory for the lesson). The "Задания" section has a table with columns "английский" (English) and "русский" (Russian). It shows "No results found". Below the table is a form titled "Форма для добавления заданий" (Form for adding tasks) with fields for "Предложение на английском языке" (Sentence in English) and "Предложение на русском языке" (Sentence in Russian). Both forms have "Изменить" (Edit) and "Добавить" (Add) buttons.

Рисунок 4.4 – Страница уроки

При вводе данных в поля происходит проверка по определённому регулярному выражению. В случае ввода ошибочных данных и попытке добавления, под полем ввода появляется сообщение об ошибке. Результат ввода ошибочных значений в формы добавления представлен на рисунке 4.5.

Рисунок 4.5 – Результат ввода ошибочных значений на странице уроков

При исправлении введённых данных на корректные и нажатии на кнопку все сообщения об ошибках исчезнут.

Заполнив поля ввода корректными данными и нажав на кнопку изменения, предварительно не выбрав урок или задание, появится соответствующая ошибка, представленная на рисунке 4.6.

английский ↑↓	русский ↑↓	
I live in London.	Я живу в Лондоне.	
The meeting starts at six o'clock.	Собрание начинается в шесть часов.	
We analyse what our clients may need, develop a new product, produce a sample, improve it and sell it.	Мы анализируем, что может понадобиться нашим клиентам, разрабатываем новый продукт, изготавливаем образец, дорабатываем его и продаем.	
The Earth is round.	Земля – круглая.	
The Volga is the longest river in Europe.	Волга – самая длинная река в Европе.	

Рисунок 4.6 – Изменение невыбранного задания

Ошибка не будет появляться, если урок или задание будут выбраны.

5. Руководство пользователя

5.1 Регистрация и авторизация

При первом переходе к приложению у неавторизованного пользователя есть возможность зарегистрироваться либо авторизоваться. Форма со страницы регистрации приведена на рисунке 5.1.

The registration form is titled "Регистрация". It contains two input fields: "Введите email" and "Введите имя". Below the email field is a red error message: "✗ Некорректный email". Below the name field is a red error message: "✗ Минимум 1 символ". At the bottom is a blue button labeled "Зарегистрироваться".

Рисунок 5.1 – Форма для регистрации пользователя

При заполнении формы корректными данными и последующем нажатии на кнопку регистрации, пользователь получит сообщение, показанное на рисунке 5.2.

A success message dialog box titled "Регистрация прошла успешно" with a close button (✗) in the top right corner. The text inside reads: "Ваш пароль отправлен на почту v.belashkov02@gmail.com. Для первого входа в аккаунт используйте его. В настройках аккаунта вы сможете изменить пароль."

Рисунок 5.2 – Сообщение об успешной регистрации

После закрытия диалогового окна пользователь перенаправляется на страницу входа, представленную на рисунке 5.3.

The authorization form is titled "Авторизация". It contains two input fields: the first contains the email "v.belashkov02@gmail.com" and the second contains masked characters ".....". Below the fields is a blue button labeled "Войти".

Рисунок 5.3 – Форма для авторизации пользователя

После введения корректных данных и нажатия кнопки входа, пользователь перенаправляется на страницу со статистикой.

5.2 Статистика

На странице статистики находится таблица со статистикой всех пользователей за всё время, личная статистика за всё время и за текущий день в виде диаграмм. Страница представлена на рисунке 5.4.

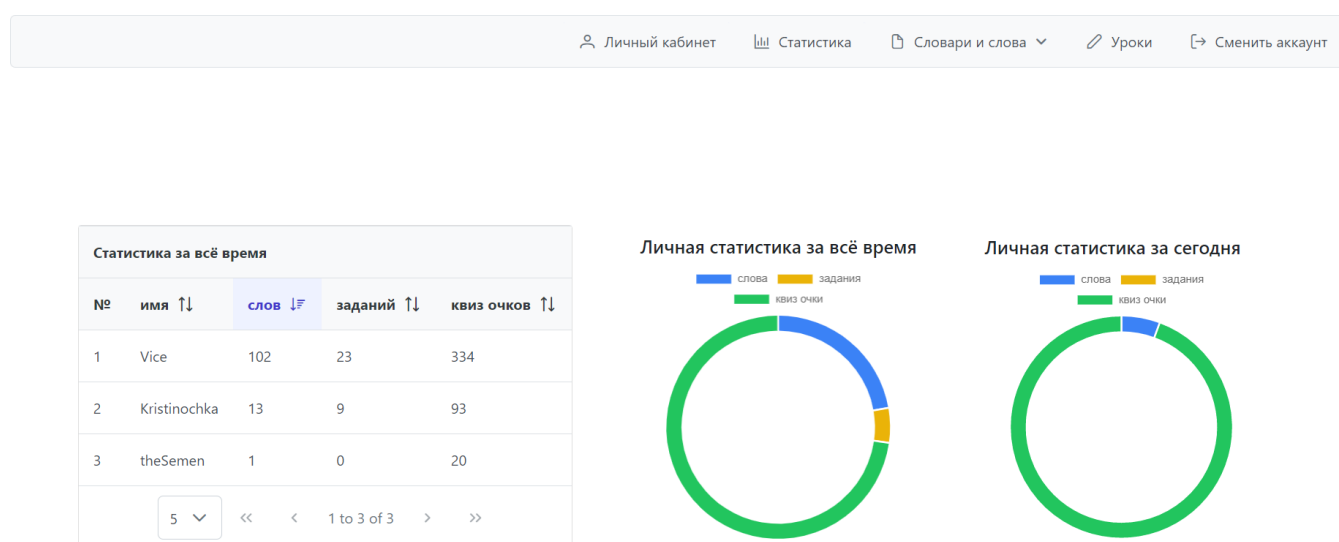


Рисунок 5.4 – Страница статистики

Перемещение по приложению осуществляется с помощью навигационной панели вверху страницы, которую можно увидеть на рисунке 5.4. Для изменения настроек аккаунта или обучения, пользователю необходимо перейти на страницу «Личный кабинет», которая представлена на рисунке 5.5.

Обновление данных аккаунта

Текущий пароль

Новый пароль

Повторите новый пароль

От 4 до 16 символов

Пароли не совпадают

Сменить пароль

Настройки обучения

Количество правильных ответов, следующих подряд, после которого слово считается выученным и исключается из списка, но остаётся в словаре

Количество слов изучаемых одновременно

Способ изучения лексикона

Способ изучения грамматики

Сохранить

Рисунок 5.5 – Страница «Личный кабинет»

На данной странице можно изменить пароль пользователя, заполнив все 3 поля верными и валидными данными. Чтобы изменить настройки обучения,

необходимо выставить нужные параметры на ползунках и выпадающих списках и нажать на кнопку «Сохранить».

5.3 Страница изменения словарей и слов

В выпадающем меню «Словари и слов», нажав на кнопку «Изменить», пользователю откроется страница, изображенная на рисунке 5.6. На этой странице представлены 2 таблицы и 2 формы для добавления словарей и слов. При введении некорректных данных, пользователь получит сообщение об ошибке. Для добавления слова дополнительно необходимо выбрать словарь, нажав на необходимые в таблице словарей.

Рисунок 5.6 – Страница изменения словарей и слов

Для обновления и удаления словарей и слов есть соответствующие кнопки в каждой строке таблиц. Пример изменения слова изображен на рисунке 5.7.

Рисунок 5.7 – Изменение слова

После ввода всех изменений необходимо нажать на «галочку», чтобы изменения применились или на «крестик», чтобы изменения не были сохранены.

Также в таблицах доступны сортировки по всем столбцам, при нажатии на название столбца, и поиск по таблице с помощью поля ввода в шапке таблицы.

5.4 Страница просмотра словарей и слов

Для перехода на данную страницу необходимо в выпадающем меню «Словари и слов» нажать на кнопку «Просмотреть». Страница представлена на рисунке 5.8.

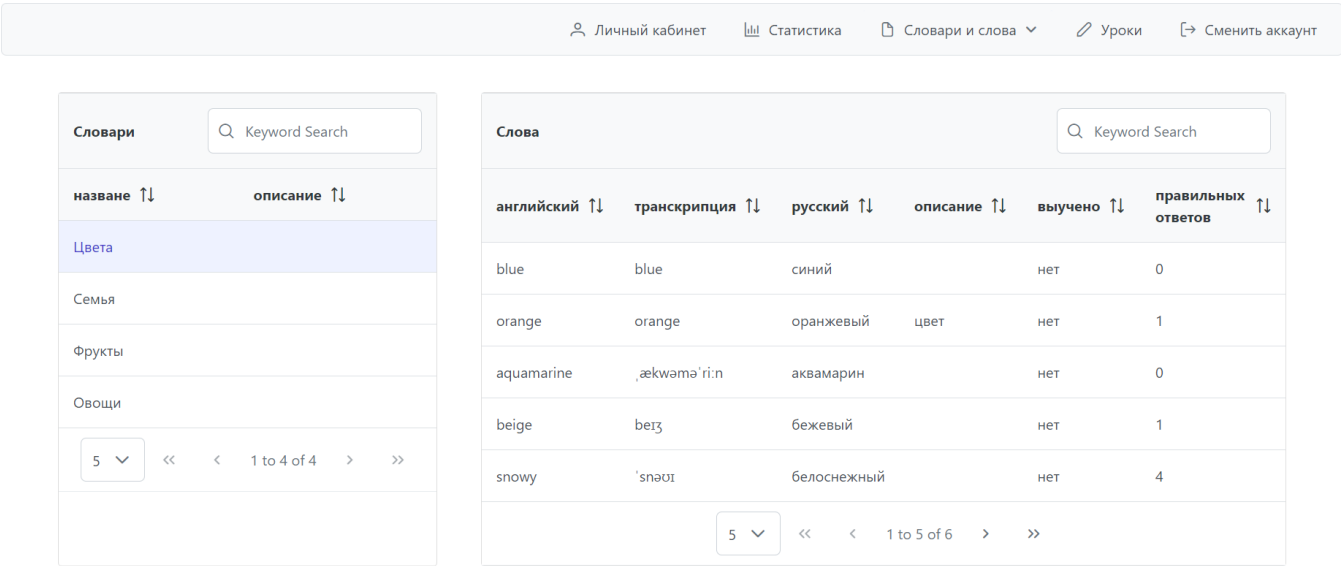


Рисунок 5.8 – Страница просмотра словарей и слов

На данной странице в таблицах отображены словари пользователя и администратора. Для каждого слова дополнительно к основной информации показан текущий прогресс изучения и выучено слово или нет.

5.5 Страница изучения слов

В выпадающем меню «Словари и слов», нажав на кнопку «Изменить», пользователю откроется страница, изображенная на рисунке 5.9.

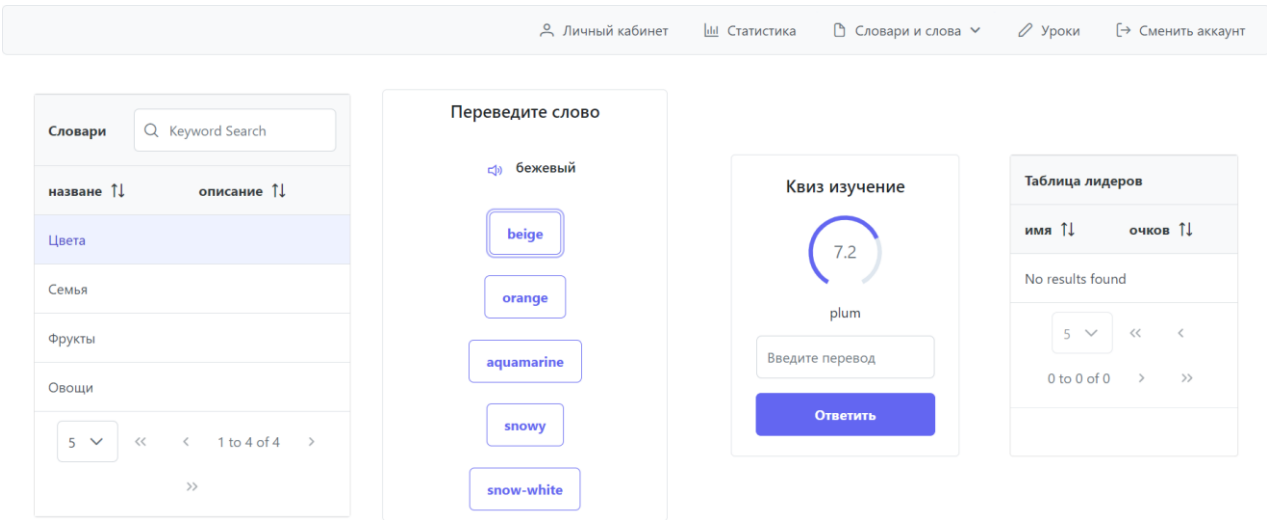


Рисунок 5.9 – Страница изучения слов

На странице доступны два варианта изучения слов: выбор из пяти вариантов и перевод слова в виде квиза. Выбрав нужный словарь, пользователю в случайном порядке подбирается слово и 5 вариантов перевода, один из которых правильны. Также доступна функция прослушивания текущего слова. При правильном ответе слово меняется, в обратном случае пользователь получит сообщение, представленное на рисунке 5.10.

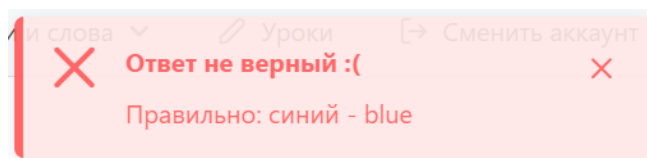


Рисунок 5.10 – Сообщение при неверном переводе слова

Второй вариант изучение подразумевает под собой генерацию случайного слова раз в 10 секунд, перевод которого нужно ввести в текстовое поле. При неверном переводе пользователь получит сообщение, аналогичное сообщению на рисунке 5.10, в обратном случае страница изменится так, как показано на рисунке 5.11.

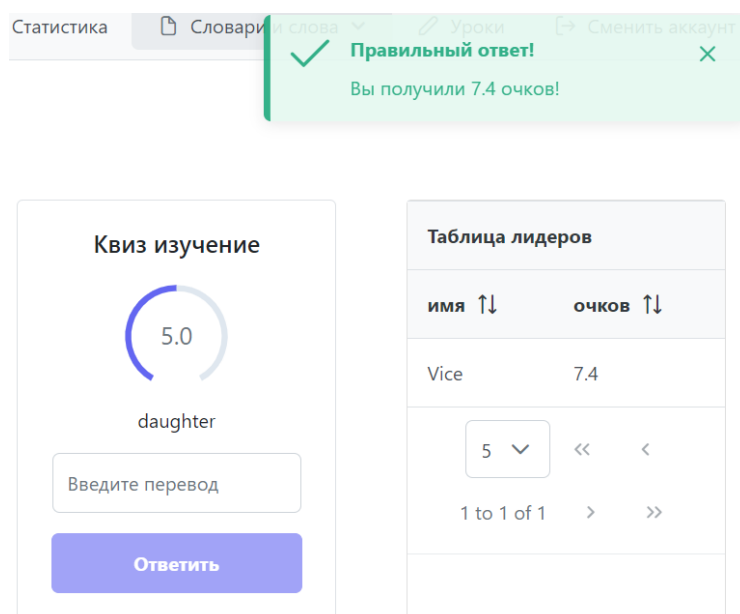


Рисунок 5.11 – Верный перевод слова в квизе

После верного ответа пользователь получает квиз очки, количество которых равно оставшемуся времени на ответ, попадёт в таблицу лидеров и кнопка отправления ответа станет неактивна. Таблица лидеров обновляется для каждого слова.

5.6 Страница уроков

В навигационном меню, нажав на кнопку «Уроки», пользователь будет направлен на одноименную страницу. Интерфейс данной страницы представлен на рисунке 5.12.

Уроки

Keyword Search

название ↑↓	проходной процент ↑↓
Present Simple	50
Past Simple	60

5

<<

<

1 to 2 of 2

>

>>

Переведите предложение, расставив слова в нужном порядке

Волга – самая длинная река в Европе.

in

Europe

river

Volga

is

the

longest

The

– Теория

Present Simple - простое настоящее время. Время Present Simple обозначает действие в настоящем в широком смысле слова. Оно употребляется для обозначения обычных, регулярно повторяющихся или постоянных действий, например, когда мы говорим о чьих-либо привычках, режиме дня, расписании и т. д., т. е. Present Simple обозначает действия, которые происходят в настоящее время, но не привязаны именно к моменту речи. I live in London. Я живу в Лондоне. The meeting starts at 6

Рисунок 5.12 – Страница «Уроки»

Выбрав один из доступных уроков, форма с заданиями будет заполнена случайным заданием из выбранного урока. Из слов, находящихся в кнопках, нужно составить перевод предложения, нажимая в нужном порядке на необходимые слова. В случае правильного перевода предложения будет выбрано следующее задание, в обратном случае пользователь получит ошибку, которая представлена на рисунке 5.13.

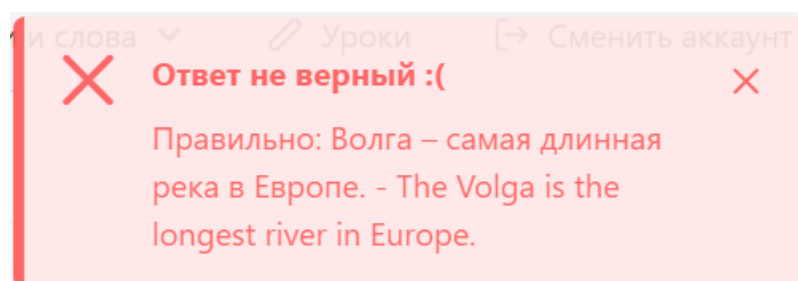


Рисунок 5.13 – Сообщение при неверном переводе предложения

Когда задания в уроке закончатся, то пользователю будет предложено выбрать новый урок, если предыдущий выполнен на проходной процент или пройти урок заново.

5.7 Страница изменения уроков и заданий

Данная страница доступна только для пользователя с ролью администратора. Для перехода на неё, администратору необходимо нажать на кнопку «Уроки», находящуюся в навигационной панели. На рисунке 5.14 представлен интерфейс данной страницы.

Уроки

название ↑↓	проходной процент ↑↓	
Present Simple	50	
Past Simple	60	
Future Simple	45	

5 ▾ << < 1 to 3 of 3 > >>

Задания

английский ↑↓	русский ↑↓
No results found	

5 ▾ << < 0 to 0 of 0 > >>

Форма для добавления уроков

Название

Проходной процент

Теория к уроку

Изменить Добавить

Форма для добавления заданий

Предложение на английском языке

Предложение на русском языке

Изменить Добавить

Рисунок 5.14 – Страница «Уроки» для администратора

На странице находятся две таблицы для уроков и заданий и две формы для добавления уроков и заданий. Для удаления урока в его строке нужно нажать на кнопку со значком мусорки. Удаление задания происходит аналогично. Выбрав урок или задание форма будет заполнена данными, как показано на рисунке 5.15.

уроки

название	процент	
Present Simple	50	
Past Simple	60	
Future Simple	45	

5 ▾ << < 1 to 3 of 3 > >>

Задания

английский	русский
When you see a big green house, turn left.	Когда вы увидите большой зеленый дом, поверните налево.

5 ▾ << < 6 to 6 of 6 > >>

Форма для добавления уроков

Название

Проходной процент

Теория к уроку

Present Simple - простое настоящее время
 Время Present Simple обозначает действие в настоящем в широком смысле слова. Оно употребляется для обозначения обычных, регулярно повторяющихся или постоянных действий, например, когда мы говорим о чьих-либо привычках, режиме дня, расписании и т.д. Present Simple обозначает действия, которые происходят в настоящее время на...

Изменить Добавить

Форма для добавления заданий

Предложение на английском языке

When you see a big green house, turn left.

Предложение на русском языке

Когда вы увидите большой зеленый дом, поверните налево.

Изменить Добавить

Рисунок 5.15 – Заполнение форм при выборе урока и задания

Для изменения выбранного урока или задания необходимо исправить данные в форме и нажать на кнопку «Изменить».

6. Заключение

Перед началом разработки был произведен аналитический обзор прототипов приложений подобной тематики и определение функциональных возможностей разрабатываемого приложения.

В результате написания курсового проекта, было разработано web-приложение, состоящее из двух проектов, а именно серверной части на Node.js с использованием фреймворка NestJS и клиентской части с использованием библиотеки React JS. Для взаимодействия с базой данных была использована ORM Prisma, что облегчило разработку приложения. Также было написано руководство пользователя для веб-приложения.

Для повышения безопасности передачи данных в приложении используется протокол HTTPS. В приложении были использованы сокеты и асинхронные запросы.

При разработке выполнены следующие задачи:

- исследованы преимущества и недостатки аналогичных приложений;
- разработана структура базы данных;
- разработана серверная часть приложения;
- разработана клиентская часть приложения;
- проведено тестирование приложения;
- написано руководство пользователя;

Представлено описание каждого этапа проектирования и реализации веб-приложения, которое позволяет изучать английский язык посредством расширения лексикона двумя доступными способами, изучение грамматики с помощью теоретических сведений и выполнения заданий, установить настройки обучения под свои способности, просматривать текущий личный прогресс.

В пояснительной записке к курсовому проекту приведено описание руководства пользователя разработанного веб-приложения.

В ходе тестирования приложения были выявлены и исправлены ошибки, что позволяет приложению работать бесперебойно. По итогу, веб-приложение успешно реализует свою главную цель – предоставление пользователям удобного сервиса для изучения английского языка.

7. Список использованной литературы

- 1 Документация NestJS [Электронный ресурс] – Режим доступа: <https://docs.nestjs.com/> – Дата доступа: 17.05.2023.
- 2 Документация ORM Prisma [Электронный ресурс] – <https://www.prisma.io/> – Дата доступа: 17.05.2023.
- 3 React [Электронный ресурс] – Режим доступа: <https://react.dev/> – Дата доступа: 17.05.2023.
- 4 Node.js [Электронный ресурс] – Режим доступа: <https://nodejs.org/en/about/> – Дата доступа: 17.05.2023.
- 5 Socket.io [Электронный ресурс] – Режим доступа: <https://socket.io/> – Дата доступа: 17.05.2023.
- 6 Асинхронный веб, или что такое веб-сокеты [Электронный ресурс] – Режим доступа: <https://tproger.ru/translations/what-are-web-sockets/> – Дата доступа: 17.05.2023.

Приложение А

```

generator client {
  provider = "prisma-client-js"
}

datasource db {
  provider = "postgresql"
  url      = env("DATABASE_URL")
}

model User {
  id          Int           @id @default(autoincrement())
  name        String
  email       String        @unique
  password    String
  roleId      Int           @default(1) @map("role_id")
  idEnableLesson Int        @default(1)
  @map("id_enable_lesson")
  role        Role          @relation(fields: [roleId],
references: [id], onDelete: Cascade)
  session     Session?
  grammarProgress GrammarProgress[]
  settings     UserSettings?
  dictionary   Dictionary[]
  lexiconProgress LexiconProgress[]
  statistics   Statistics[]

  @@map("users")
}

model UserSettings {
  userId      Int           @unique @map("user_id")
  user        User          @relation(fields:
[userId], references: [id], onDelete: Cascade)
  countRepeatWordForLearned Int
  @map("count_repeat_word_for_learned")
  countRepeatWordsSimultaneously Int
  @map("count_learning_words_simultaneously")
  learningModeWords      LearningMode
  learningModeTasks      LearningMode

  @@map("users_settings")
}

model Lesson {
  id          Int           @id @default(autoincrement())
  name        String
  theory      String
  passingPercent Int        @map("passing_percent")
  tasks       Task[]

  @@map("lessons")
}

```

```

model Task {
    id            Int            @id @default(autoincrement())
    englishSentence String        @map("english_sentence")
    russianSentence String        @map("russian_sentence")
    lessonId      Int            @map("lesson_id")
    lesson        Lesson         @relation(fields: [lessonId],
references: [id], onDelete: Cascade)
    grammarProgress GrammarProgress[]

    @@unique([englishSentence, russianSentence, lessonId])
    @@map("tasks")
}

model GrammarProgress {
    id      Int  @id @default(autoincrement())
    userId Int  @map("user_id")
    user   User @relation(fields: [userId], references: [id], onDelete:
Cascade)
    taskId Int  @map("task_id")
    task   Task @relation(fields: [taskId], references: [id], onDelete:
Cascade)

    @@unique([userId, taskId])
    @@map("grammar_progresses")
}

model Dictionary {
    id            Int            @id @default(autoincrement())
    name          String
    description    String?
    creatorId     Int            @map("creator_id")
    user          User           @relation(fields: [creatorId],
references: [id], onDelete: Cascade)
    dictionaryToWord DictionaryToWord[]
    words         Word[]

    @@map("dictionaries")
}

model Word {
    id            Int            @id @default(autoincrement())
    englishSpelling String        @map("english_spelling")
    transcription  String
    russianSpelling String        @map("russian_spelling")
    description    String?
    lexiconProgress LexiconProgress[]
    dictionaryToWord DictionaryToWord[]
    dictionaries   Dictionary[]

    @@unique([englishSpelling, russianSpelling])
    @@map("words")
}

```

```

model DictionaryToWord {
    id            Int            @id @default(autoincrement())
    dictionaryId  Int
    wordId        Int
    dictionary    Dictionary @relation(fields: [dictionaryId],
references: [id], onDelete: Cascade)
    word          Word          @relation(fields: [wordId], references:
[id], onDelete: Cascade)

    @@unique([dictionaryId, wordId])
    @@map("dictionary_to_word")
}

model LexiconProgress {
    id            Int            @id @default(autoincrement())
    progressCount Int            @map("progress_count")
    isLearned     Boolean        @map("is_learned")
    userId        Int            @map("user_id")
    user          User           @relation(fields: [userId], references: [id],
onDelete: Cascade)
    wordId        Int            @map("word_id")
    word          Word           @relation(fields: [wordId], references: [id],
onDelete: Cascade)

    @@unique([userId, wordId])
    @@map("lexicon_progresses")
}

model Statistics {
    id            Int            @id @default(autoincrement())
    date          DateTime
    words         Int?
    tasks         Int?
    quizPoints    Float?        @map("quiz_points")
    userId        Int            @map("user_id")
    user          User           @relation(fields: [userId], references: [id],
onDelete: Cascade)

    @@unique([date, userId])
    @@map("statistics")
}

model Session {
    userId        Int            @unique @map("user_id")
    user          User           @relation(fields: [userId], references: [id],
onDelete: Cascade)
    refreshToken  String         @map("refresh_token")

    @@map("sessions")
}

model Role {
    id            Int            @id @default(autoincrement())
    title         String

```

```

permissions Permission[]
users          User[]

  @@map("roles")
}

model Permission {
  id          Int          @id @default(autoincrement())
  descriptor  String
  context     String?
  method      Method
  roleId      Int          @map("role_id")
  role        Role        @relation(fields: [roleId], references: [id],
onDelete: Cascade)

  @@map("permissions")
}

enum Method {
  ALL
  GET
  POST
  DELETE
  PUT
  PATCH
}

enum LearningMode {
  TRANSLATE_FROM_ENGLISH
  TRANSLATE_FROM_RUSSIAN
  COMBINED
}

```

Приложение Б

```
{
  "name": "english-api",
  "version": "1.0.0",
  "description": "",
  "author": "",
  "private": true,
  "license": "UNLICENSED",
  "scripts": {
    "build": "nest build",
    "format": "prettier --write \"src/**/*.ts\" \"test/**/*.ts\"",
    "start": "nest start",
    "start:dev": "nest start --watch",
    "start:debug": "nest start --debug --watch",
    "start:prod": "node dist/main",
    "lint": "eslint \"{src,apps,libs,test}/**/*.ts\" --fix",
    "test": "jest",
    "test:watch": "jest --watch",
    "test:cov": "jest --coverage",
    "test:debug": "node --inspect-brk -r tsconfig-paths/register -r
ts-node/register node_modules/.bin/jest --runInBand",
    "test:e2e": "jest --config ./test/jest-e2e.json"
  },
  "dependencies": {
    "@nestjs/common": "^9.0.0",
    "@nestjs/config": "^2.2.0",
    "@nestjs/core": "^9.0.0",
    "@nestjs/jwt": "^10.0.2",
    "@nestjs/mapped-types": "*",
    "@nestjs/passport": "^9.0.3",
    "@nestjs/platform-express": "^9.0.0",
    "@nestjs/platform-socket.io": "^9.4.0",
    "@nestjs/swagger": "^6.1.4",
    "@nestjs/websockets": "^9.4.0",
    "@prisma/client": "^4.11.0",
    "bcrypt": "^5.1.0",
    "class-transformer": "^0.5.1",
    "class-validator": "^0.14.0",
    "cookie": "^0.5.0",
    "cookie-parser": "^1.4.6",
    "cors": "^2.8.5",
    "helmet": "^7.0.0",
    "https": "^1.0.0",
    "joi": "^17.9.0",
    "nodemailer": "^6.7.8",
    "passport": "^0.6.0",
    "passport-jwt": "^4.0.1",
    "passport-local": "^1.0.0",
    "reflect-metadata": "^0.1.13",
    "rxjs": "^7.8.1",
    "websocket": "^1.0.34"
  },
  "devDependencies": {

```

```

"@nestjs/cli": "^9.0.0",
"@nestjs/schematics": "^9.0.0",
"@nestjs/testing": "^9.0.0",
"@types/bcrypt": "^5.0.0",
"@types/cookie-parser": "^1.4.3",
"@types/express": "^4.17.13",
"@types/jest": "29.2.4",
"@types/node": "^18.11.18",
"@types/passport-jwt": "^3.0.8",
"@types/passport-local": "^1.0.35",
"@types/supertest": "^2.0.11",
"@types/ws": "^8.5.4",
"@typescript-eslint/eslint-plugin": "^5.0.0",
"@typescript-eslint/parser": "^5.0.0",
"eslint": "^8.0.1",
"eslint-config-prettier": "^8.3.0",
"eslint-plugin-prettier": "^4.0.0",
"jest": "29.3.1",
"prettier": "^2.3.2",
"prisma": "^4.12.0",
"source-map-support": "^0.5.20",
"supertest": "^6.1.3",
"ts-jest": "29.0.3",
"ts-loader": "^9.2.3",
"ts-node": "^10.9.1",
"tsconfig-paths": "4.1.1",
"typescript": "^4.9.5"
},
"jest": {
  "moduleFileExtensions": [
    "js",
    "json",
    "ts"
  ],
  "rootDir": "src",
  "testRegex": ".*\\.spec\\.ts$",
  "transform": {
    "^.+\\.?(t|j)s$": "ts-jest"
  },
  "collectCoverageFrom": [
    "**/*.?(t|j)s"
  ],
  "coverageDirectory": "../coverage",
  "testEnvironment": "node"
},
"prisma": {
  "seed": "ts-node prisma/seed.ts"
}
}

```


Приложение В

```
{
  "name": "english-front",
  "proxy": "https://localhost:5000/",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@fortawesome/fontawesome-svg-core": "^6.3.0",
    "@fortawesome/free-regular-svg-icons": "^6.3.0",
    "@fortawesome/free-solid-svg-icons": "^6.3.0",
    "@fortawesome/react-fontawesome": "^0.2.0",
    "@testing-library/jest-dom": "^5.16.4",
    "@testing-library/react": "^13.1.1",
    "@testing-library/user-event": "^13.5.0",
    "axios": "^1.3.3",
    "bootstrap": "^5.2.3",
    "chart.js": "^4.3.0",
    "date-fns": "^2.29.3",
    "dotenv": "^16.0.3",
    "js-cookie": "^3.0.1",
    "jwt-decode": "^3.1.2",
    "mobx": "^6.0.5",
    "mobx-react-lite": "^3.1.7",
    "primeicons": "^6.0.1",
    "primereact": "^9.2.3",
    "react": "^18.0.0",
    "react-bootstrap": "^2.7.4",
    "react-date-range": "^1.4.0",
    "react-dom": "^18.2.0",
    "react-router-dom": "^6.10.0",
    "react-scripts": "5.0.1",
    "rxjs": "^7.8.1",
    "socket.io-client": "^4.6.1",
    "web-vitals": "^2.1.4",
    "ws": "^8.13.0"
  },
  "scripts": {
    "start": "set HTTPS=true& set  
SSL_CERT_FILE=C:\\My\\OpenSSL\\CW.crt& set  
SSL_KEY_FILE=C:\\My\\OpenSSL\\CW.key& react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",

```

```
    "not dead",  
    "not op_mini all"  
  ],  
  "development": [  
    "last 1 chrome version",  
    "last 1 firefox version",  
    "last 1 safari version"  
  ]  
}  
}
```