Техническое задание на разработку веб-приложения «Платформа для изучения иностранных языков»

1. Введение

1.1 Цель документа

Цель данного документа — определить требования к разработке веб-приложения «Платформа для изучения иностранных языков», предназначенного для упрощения процесса обучения, взаимодействия между преподавателями и студентами, а также для управления образовательным контентом.

1.2 Актуальность и назначение проекта

Современные технологии предоставляют широкие возможности для онлайн-обучения, что особенно актуально в условиях удаленной работы и учебы. Разрабатываемая платформа будет способствовать повышению доступности качественного обучения языкам, предоставляя инструменты для управления уроками, тестами и анализом прогресса студентов.

1.3 Основные пользователи системы

Администраторы — управление пользователями, курсами и аналитикой.

Преподаватели — создание курсов, уроков и тестов.

Студенты — прохождение уроков, выполнение тестов, анализ прогресса.

2. Цели и задачи проекта

2.1 Основные цели

Обеспечение удобного процесса изучения иностранных языков через централизованную платформу.

Упрощение управления курсами и студентами для преподавателей.

Предоставление инструментов для анализа прогресса обучения.

2.2 Конкретные задачи

Разработка интерфейса для просмотра и прохождения курсов.

Реализация функционала для создания и управления образовательным контентом.

Создание системы тестирования знаний студентов.

Внедрение аналитических инструментов для администраторов и преподавателей.

3. Требования к системе

3.1 Функциональные требования

Управление пользователями:

Регистрация, авторизация и управление профилями пользователей.

Разграничение прав доступа: администраторы, преподаватели, студенты.

Курсы и уроки:

Добавление, редактирование и удаление курсов.

Управление уроками.

Тестирование:

Создание тестов с вариантами ответов.

Автоматическая проверка тестов.

Отзывы:

Возможность студентам оставлять отзывы с рейтингами для курсов.

Статистика и отчеты:

Пример: При прохождении урока студент должен видеть индикатор прогресса и возможность повторного прохождения теста.

3.2 Нефункциональные требования

Производительность: время отклика интерфейса не более 2 секунд.

Масштабируемость: поддержка до 10,000 пользователей одновременно.

Безопасность:

Шифрование данных через SSL.

Хранение паролей в хешированном виде.

4. Требования к пользовательскому интерфейсу

4.1 Основные экраны

Главная страница: отображение доступных курсов с фильтрацией по уровню сложности.

Страница курса: список уроков, прогресс, отзывы и тесты.

Административная панель: управление пользователями.

Личный кабинет студента: пройденные уроки, личные данные.

4.2 Навигация

Удобное меню с основными разделами: «Курсы», «Профиль».

4.3 Юзабилити

Адаптивный дизайн для настольных устройств.

Интуитивно понятный интерфейс с минимальным количеством действий для выполнения задач.

Пример: На странице курса студент должен видеть прогресс по урокам и возможность начать следующий урок.

5. Требования к технической реализации

5.1 Языки и технологии

Backend: Laravel.

Frontend: Чистый JavaScript или Vue.js/React, AJAX-запросы через jQuery.

База данных: MySQL.

5.2 Архитектура системы

REST API для взаимодействия между клиентом и сервером.

Модульная архитектура с разделением логики.

5.3 Интеграции

Внешние уведомления через email (например, SendGrid).

6. Требования к безопасности

6.1 Аутентификация и авторизация

Разделение прав доступа.

6.2 Шифрование данных

HTTPS для всех соединений.

Хранение паролей через bcrypt.

6.3 Политики доступа

SQL-инъекции и XSS должны быть предотвращены.

Ограничение по IP для административного доступа.

7. Ограничения и допущения

7.1 Технические ограничения

7.2 Сроки выполнения

Разработка — месяц.

Тестирование — хз.

8. Требования к тестированию и приемке

8.1 Типы тестирования

Функциональное: проверка всех функций.

Тестирование безопасности: защита от уязвимостей.

8.2 Критерии приемки

Все требования выполнены.

9. Требования к документации

9.1 Пользовательская документация

Инструкции по регистрации, прохождению курсов и тестов.

9.2 Техническая документация

Архитектура, структура базы данных.

Диаграмма UseCase представлена на рисунке 1.

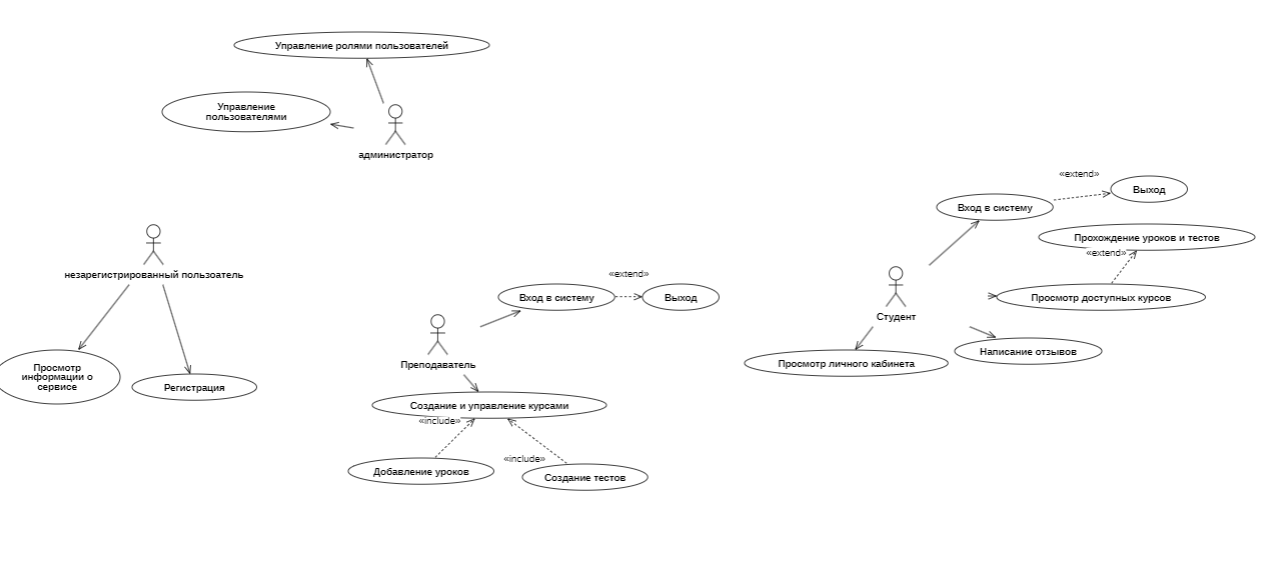


Рисунок 1 UseCase

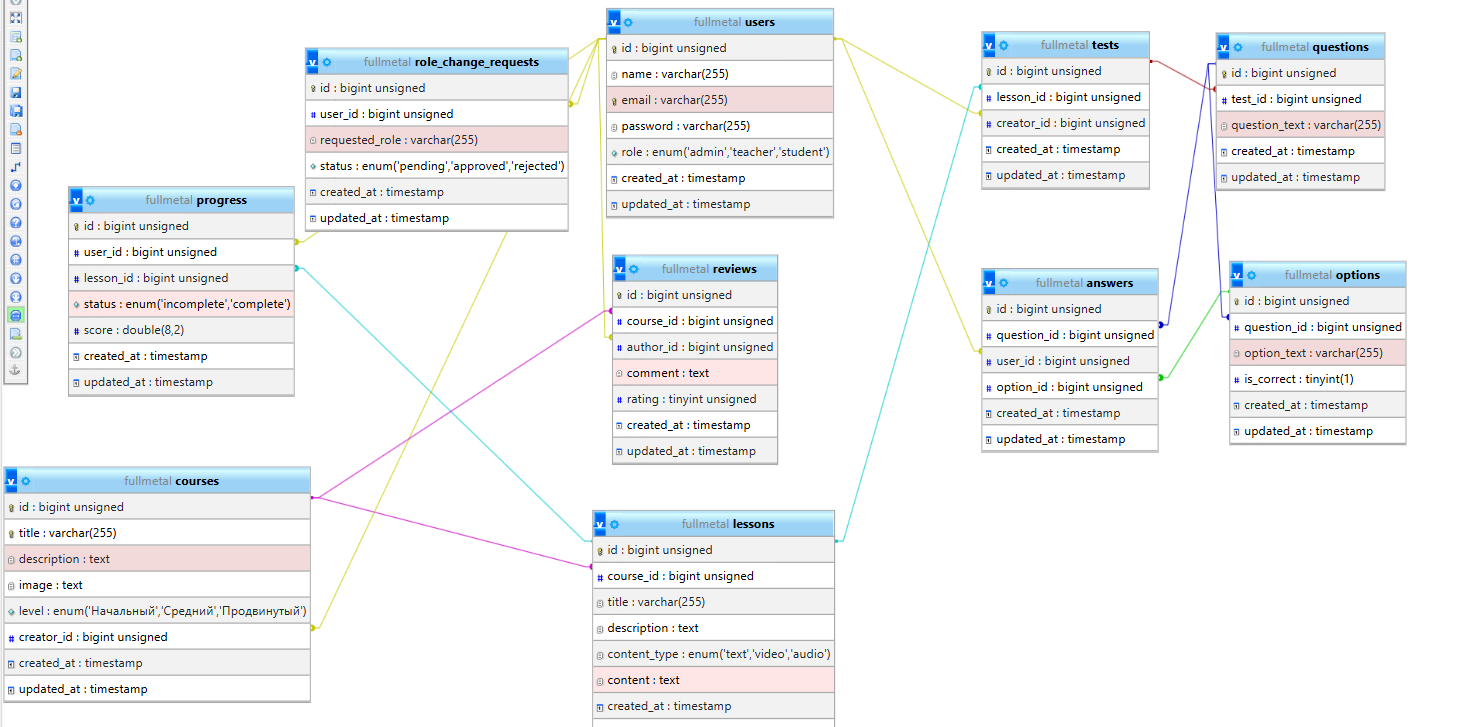


Таблица users

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Уникальный код | id | INT(11) | Первичный ключ, автоинкремент |
| Имя пользователя | name | VARCHAR(255) |  |
| Электронная почта | email | VARCHAR(255) | Уникальное |
| Пароль | password | VARCHAR(255) | Хранится в хэшированном виде |
| Роль пользователя | role | ENUM('admin', 'teacher', 'student') |  |
| Дата создания записи | created\_at | TIMESTAMP | Автоматически заполняется |
| Дата обновления записи | updated\_at | TIMESTAMP | Автоматически заполняется |

Таблица courses

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Уникальный код | id | INT(11) | Первичный ключ, автоинкремент |
| Название курса | title | VARCHAR(255) | Уникальное |
| Описание курса | description | TEXT | Nullable |
| Изображение курса | image | TEXT | Nullable |
| Уровень курса | level | ENUM('Начальный', 'Средний', 'Продвинутый') |  |
| Создатель курса | creator\_id | INT(11) | Внешний ключ на `users.id` |
| Дата создания записи | created\_at | TIMESTAMP | Автоматически заполняется |
| Дата обновления записи | updated\_at | TIMESTAMP | Автоматически заполняется |

Таблица lessons

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Уникальный код | id | INT(11) | Первичный ключ, автоинкремент |
| Код курса | course\_id | INT(11) | Внешний ключ на `courses.id` |
| Название урока | title | VARCHAR(255) |  |
| Описание урока | description | TEXT | Nullable |
| Тип контента | content\_type | ENUM('text', 'video', 'audio') |  |
| Содержимое урока | content | TEXT | Nullable |
| Дата создания записи | created\_at | TIMESTAMP | Автоматически заполняется |
| Дата обновления записи | updated\_at | TIMESTAMP | Автоматически заполняется |

Таблица tests

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Уникальный код | id | INT(11) | Первичный ключ, автоинкремент |
| Код урока | lesson\_id | INT(11) | Внешний ключ на `lessons.id` |
| Создатель теста | creator\_id | INT(11) | Внешний ключ на `users.id` |
| Дата создания записи | created\_at | TIMESTAMP | Автоматически заполняется |
| Дата обновления записи | updated\_at | TIMESTAMP | Автоматически заполняется |

Таблица reviews

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Уникальный код | id | INT(11) | Первичный ключ, автоинкремент |
| Код курса | course\_id | INT(11) | Внешний ключ на `courses.id` |
| Автор отзыва | author\_id | INT(11) | Внешний ключ на `users.id` |
| Комментарий | comment | TEXT | Nullable |
| Рейтинг | rating | TINYINT(1) | Значение от 1 до 5 |
| Дата создания записи | created\_at | TIMESTAMP | Автоматически заполняется |
| Дата обновления записи | updated\_at | TIMESTAMP | Автоматически заполняется |

Таблица progress

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Уникальный код | id | INT(11) | Первичный ключ, автоинкремент |
| Код пользователя | user\_id | INT(11) | Внешний ключ на `users.id` |
| Код урока | lesson\_id | INT(11) | Внешний ключ на `lessons.id` |
| Статус прохождения | status | ENUM('incomplete', 'complete') |  |
| Оценка | score | FLOAT | Nullable |
| Дата создания записи | created\_at | TIMESTAMP | Автоматически заполняется |
| Дата обновления записи | updated\_at | TIMESTAMP | Автоматически заполняется |

Таблица questions

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Уникальный код | id | INT(11) | Первичный ключ, автоинкремент |
| Код теста | test\_id | INT(11) | Внешний ключ на `tests.id` |
| Текст вопроса | question\_text | VARCHAR(255) |  |
| Дата создания записи | created\_at | TIMESTAMP | Автоматически заполняется |
| Дата обновления записи | updated\_at | TIMESTAMP | Автоматически заполняется |

Таблица options

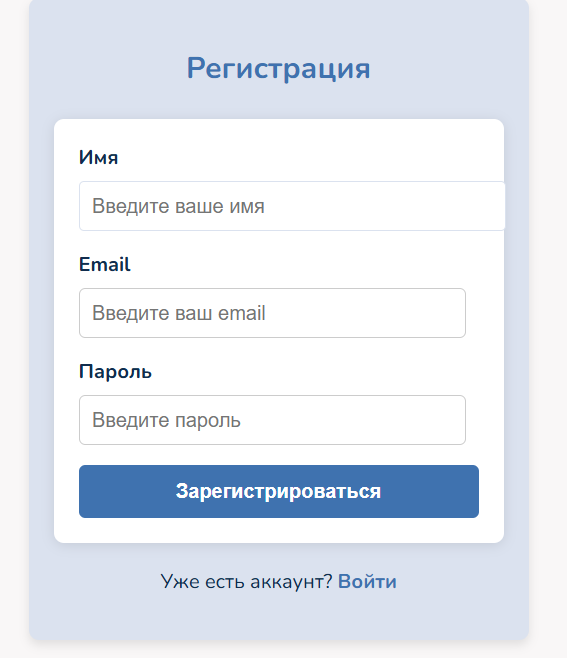
|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Уникальный код | id | INT(11) | Первичный ключ, автоинкремент |
| Код вопроса | question\_id | INT(11) | Внешний ключ на `questions.id` |
| Текст варианта ответа | option\_text | VARCHAR(255) |  |
| Правильный ответ | is\_correct | BOOLEAN | По умолчанию `false` |
| Дата создания записи | created\_at | TIMESTAMP | Автоматически заполняется |
| Дата обновления записи | updated\_at | TIMESTAMP | Автоматически заполняется |

Таблица role\_change\_requests

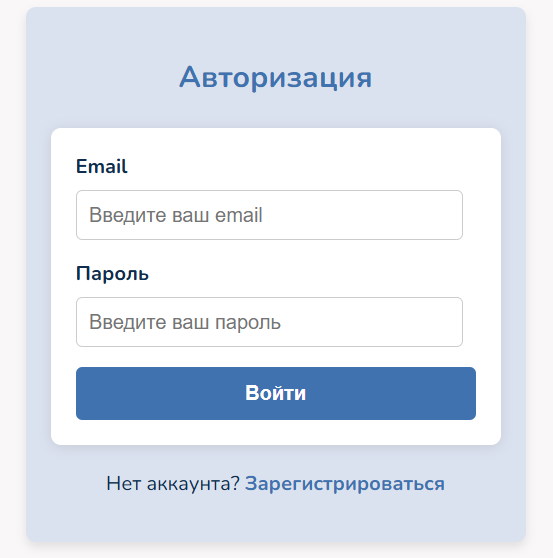
|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Уникальный код | id | INT(11) | Первичный ключ, автоинкремент |
| Код пользователя | user\_id | INT(11) | Внешний ключ на `users.id` |
| Запрашиваемая роль | requested\_role | VARCHAR(255) |  |
| Статус запроса | status | ENUM('pending', 'approved', 'rejected') | По умолчанию `pending` |
| Дата создания записи | created\_at | TIMESTAMP | Автоматически заполняется |
| Дата обновления записи | updated\_at | TIMESTAMP | Автоматически заполняется |

1. Вход / Регистрация:

Регистрация пользователя: Для того, чтобы начать пользоваться программой, необходимо зарегистрироваться. В разделе регистрации требуется указать ваше имя, адрес электронной почты и пароль. После успешной регистрации получить доступ к личному кабинету (ЛК)(рисунок 1).

  
Рисунок 1-форма регистрации

Вход в систему: Для доступа к программным возможностям необходимо войти в систему, используя ваши учетные данные (email и пароль). После входа вы будете перенаправлены на главную страницу (рисунок 2).

  
рисунок 2- форма авторизации

2. Личный кабинет (ЛК):

Личный кабинет предоставляет пользователю доступ к основным функциям программы. Здесь можно:

* просматривать и редактировать личную информацию;
* доступ к настройкам изменения роли;
* составлять и отслеживать историю выполненных курсов, уроков.

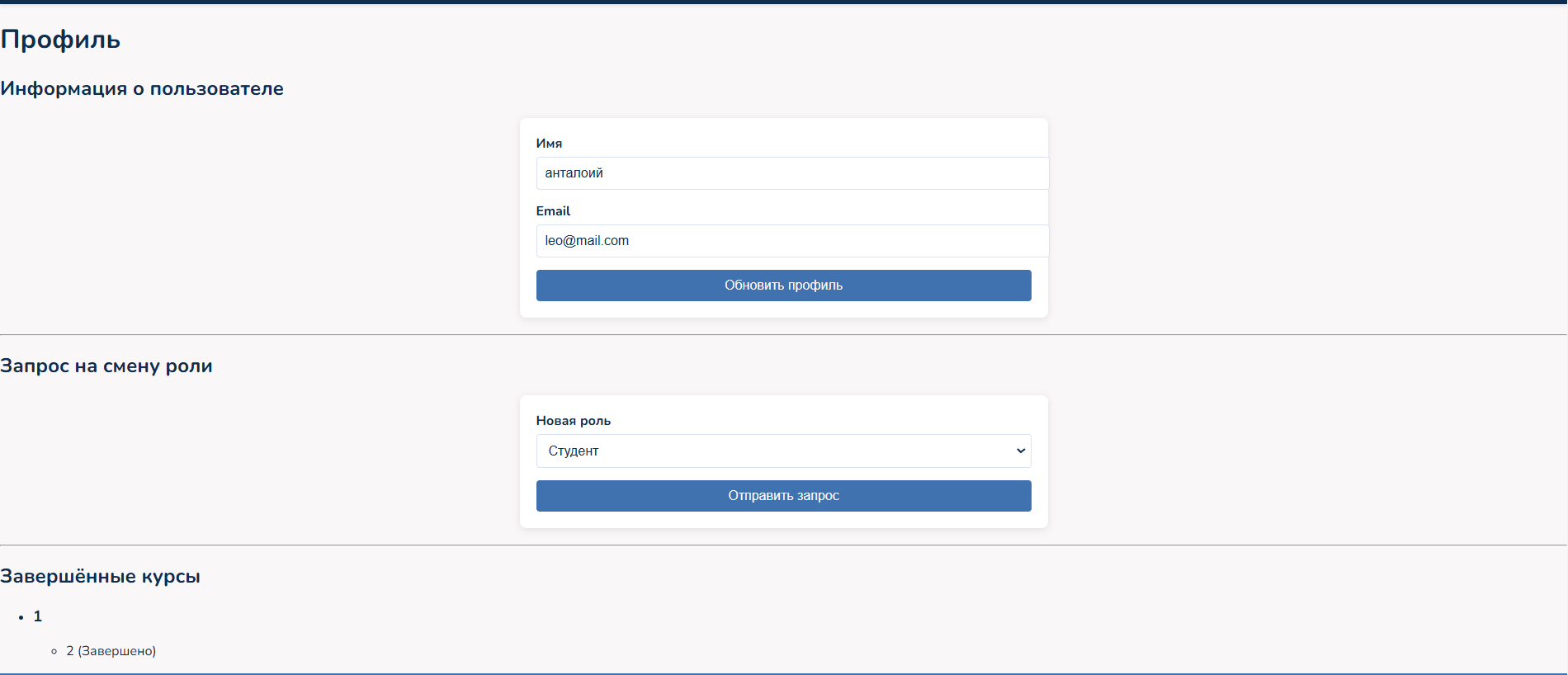


Рисунок 3- личный кабинет

Фильтрация, сортировка, пагинация, поиск:

Для улучшения работы с большими объемами данных система предоставляет различные функции:

Фильтрация: Пользователь может настроить фильтры для отображения только тех данных, которые ему нужны (рисунок 4).

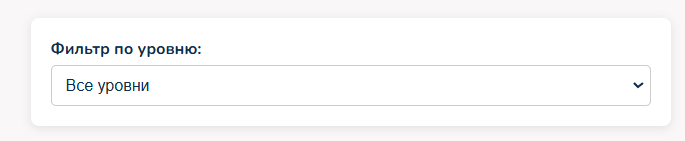


Рисунок 4- фильтрация

Сортировка: Возможность отсортировать данные по дате создания (рисунок 5).

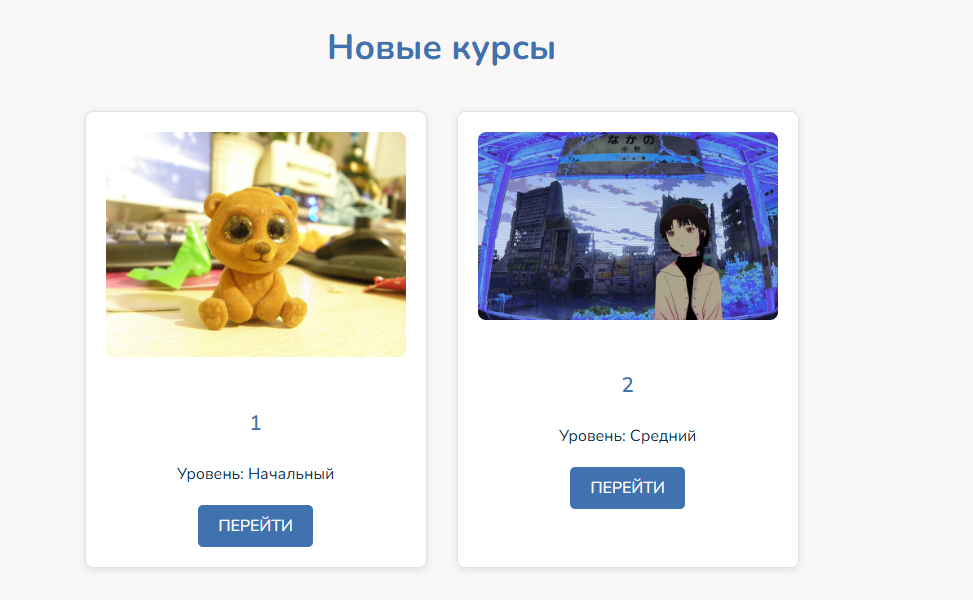


Рисунок 5- новые курсы

Пагинация: Если данных слишком много, система будет разделять их на страницы, показывая только 6 записей на одной странице. Для навигации между страницами используется пагинация (рисунок 6).

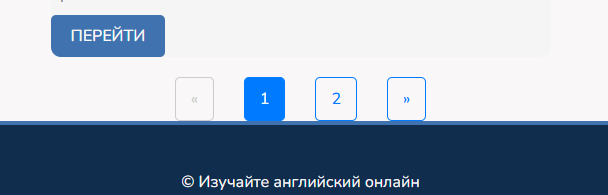


Рисунок 6- пагинация

Формирование выходного документа / статистики / данных / письма на почту:

Статистика по выполненным курсам, результаты тестов (рисунок 7,8).

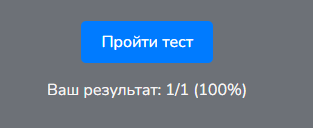


Рисунок 7- результат теста

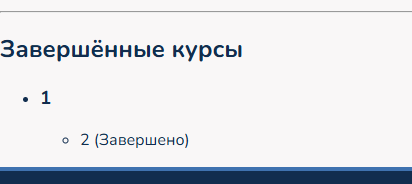


Рисунок 8- завершенные курсы, уроки

3. CRUD (Create, Read, Update, Delete):

Основной функционал приложения реализован через операции CRUD. Это позволяет пользователю создавать, просматривать, редактировать и удалять данные в системе.

Создание записи: Учитель может создать новый курс, урок, тест (рисунок 9).

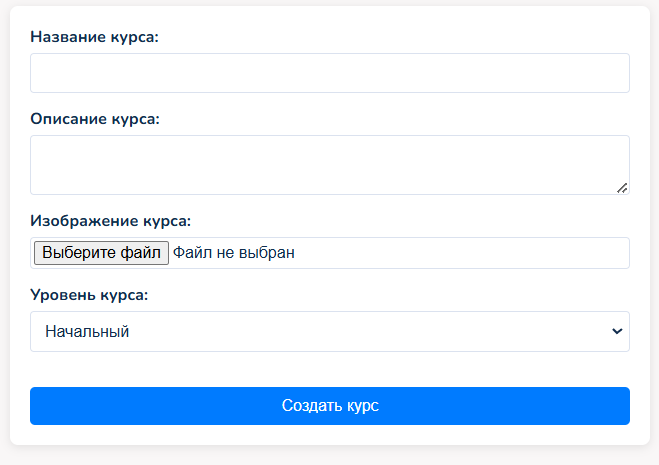
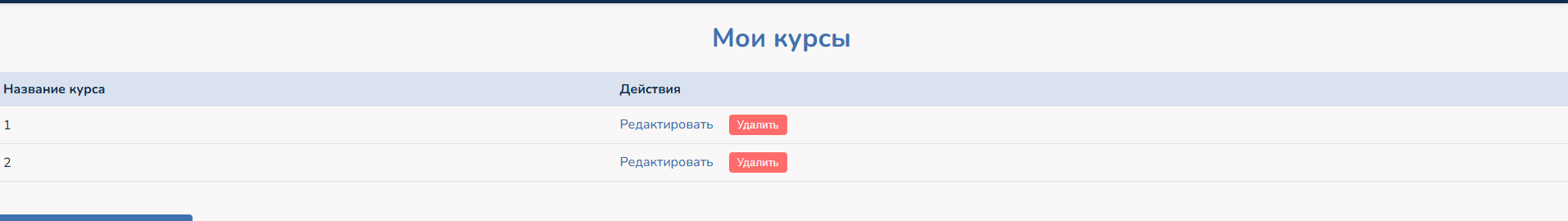


Рисунок 9- форма создания курса

Чтение записи: Просмотр всех или конкретных записей из базы данных (рисунок 10).



Обновление записи: Редактирование уже существующих записей, например, изменение описания курса или обновление информации (рисунок 11).

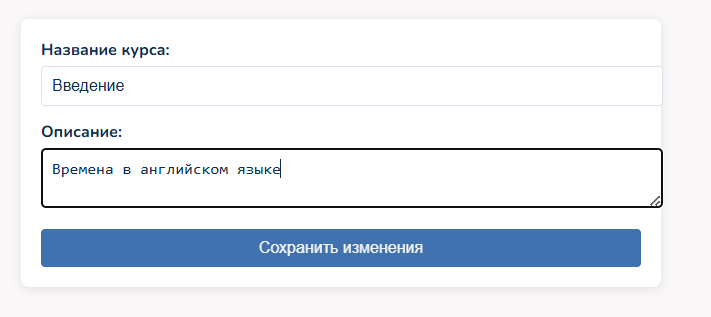


Рисунок 11- форма редактирования

Удаление записи: Удаление курса из базы данных (рисунок 10).