Министерство образования и науки Республики Башкортостан

Государственное автономное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

ОТЧЁТ

по специальности 09.02.07 Информационные системы и программирование

|  |  |
| --- | --- |
| Методист по учебно-производственной работе  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/А.И.Файзулова/ | Руководитель практики  от предприятия  \_\_\_\_\_\_\_\_\_\_\_\_/ Д.А. Медведев/  подпись ФИО |
| Руководитель практики  от учебного заведения  \_\_\_\_\_\_\_\_\_\_\_\_\_\_/А.И. Файзулова /  подпись ФИО | Студент группы 21ВЕБ-1  \_\_\_\_\_\_\_\_\_/ Валикаева П.Г./  подпись ФИО  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025 год |

УФА - 2025 год

АННОТАЦИЯ

Пояснительная записка к дипломному проекту содержит постановку и программу решения задачи «Проектирование и разработка образовательной платформы».

Программа написана на языке PHP, с использованием фреймворка Laravel и библиотек JavaScript: jQuery, html2pdf.js, Chart.js. Программа разработана в среде программирования Microsoft Visual Studio 2019.

Программа предназначена для работы в современных браузерах.

СОДЕРЖАНИЕ

|  |  |
| --- | --- |
|  | лист |
| Введение | 4 |
| 1 Постановка задачи | 6 |
| 1.1 Описание предметной области | 6 |
| 1.2 Проектирование бизнес-процессов предметной области | 9 |
| 1.3 Описание входной информации | 10 |
| 1.4 Описание выходной информации | 11 |
| 1.5 Техническое задание | 12 |
| 1.6 Описание структуры базы данных | 14 |
| 1.7 Контрольный пример | 18 |
| 2 Экспериментальный раздел | 22 |
| 2.1 Описание программы | 22 |
| 2.2 Протокол тестирования программного продукта | 25 |
| 2.3 Руководство пользователя | 32 |
| Заключение | 39 |
| Список использованных источников  Приложение А  Приложение Б | 40  43  44 |

### ВВЕДЕНИЕ

Современное общество характеризуется стремительным развитием информационных технологий, которые активно внедряются в различные сферы человеческой деятельности, включая образование. В условиях цифровизации все больше людей предпочитают получать знания в онлайн-формате, что обусловлено доступностью, гибкостью и эффективностью такого подхода. Одним из ключевых инструментов дистанционного обучения являются специализированные платформы, позволяющие создавать и проходить образовательные курсы.

Актуальность разработки образовательной платформы обусловлена возрастающим спросом на системы, обеспечивающие удобство формирования и прохождения курсов, автоматизацию проверки знаний, выдачу сертификатов, подтверждающих освоение материала.

Цель данного проекта – упрощение процесса создания, прохождения и администрирования курсов путем разработки веб-приложения.

Задачи, которые необходимо решить для достижения поставленной цели:

* спроектировать информационную систему образовательной платформы;
* определить сущности и составить наборы данных, соответствующих им;
* спроектировать базу данных;
* разработать программный продукт;
* составить контрольные примеры;
* провести тестирование;
* составить документацию.

Разработанное веб-приложение позволит сделать прохождение курсов доступным в любое время, в любом месте, в удобном для пользователя темпе. Наличие возможности тестирования позволит более эффективно усваивать материал курсов.

Основной областью применения образовательной платформы является самостоятельное обучение пользователей. Также областью применения программного продукта является использование прохождения курсов студентами в рамках образовательной программы организации. Повышение квалификации за счет прохождения курса и получения сертификата – также один из вариантов применения программного продукта. Все это возможно за счет гибкости содержания курса. Содержание может регулироваться авторами. Автором может стать сотрудник организации и составить собственный курс, на который может ссылаться организация для обучения своих сотрудников.

1. Постановка задачи

1.1 Описание предметной области

Необходимо разработать веб-приложение, предназначенное для организации образовательного процесса. Образовательная платформа должна предоставить пользователям возможность прохождения курсов. Система должна предусматривать просмотр каталога курсов с использованием поиска, фильтрации и сортировки по категориям.

В системе можно выделить следующие сущности:

* пользователи;
* категории курсов;
* курсы;
* заявки пользователей;
* изображения для описания курсов;
* изображения для контента урока;
* заявки курсов;
* уроки и тесты;
* результаты тестирований
* отзывы.

Пользователи информационной системы должны характеризоваться следующими параметрами:

* идентификатор пользователя;
* имя;
* почта;
* роль;
* пароль;
* все начатые курсы;
* завершенные курсы;
* завершенные уроки;
* статус блокировки.

Каталог курсов должен состоять из курсов разной категории. Каждая категория в системе должна характеризоваться следующими параметрами:

* идентификатор категории;
* название;
* статус наличия.

В приложении должна храниться следующая информация о курсе:

* идентификатор курса;
* категория;
* заголовок;
* описание;
* дополнительные блоки;
* изображение;
* автор;
* количество обучающихся;
* наличие заявки на вывод;
* статус доступа.

После прохождения курса пользователь может оценить курс и оставить отзыв. Отзыв характеризуется следующими атрибутами:

* идентификатор отзыва;
* идентификатор пользователя;
* идентификатор курса;
* оценка курса;
* отзыв.

Перед тем как создать дополнительные блоки курса, нужно внести изображения в определенную директорию, для хранения в базе данных и дальнейшего выбора. Изображения дополнительных блоков содержат следующие поля:

* идентификатор изображения дополнительных блоков;
* имя файла.

Также в базе данных хранятся изображения, необходимые для создания контента урока. Они характеризуются следующими полями:

* идентификатор изображения;
* имя файла.

Для публикации курса автор курса должен отправить заявку. Заявка на публикацию курса должна в системе характеризоваться следующими параметрами:

* идентификатор заявки;
* идентификатор курса;
* желаемый статус отображения;
* статус заявки.

Курс должен включать в себя контент в виде уроков и тестов. Уроки и тесты должны характеризоваться следующими параметрами:

* идентификатор занятия;
* идентификатор курса;
* тип контента (урок, тест);
* заголовок;
* содержимое;
* позиция в курсе;
* таймер.

После прохождения тестирования все результаты должны храниться в системе. Результаты должны характеризоваться следующими параметрами:

* идентификатор результата;
* идентификатор теста;
* идентификатор пользователя
* ответы;
* балл;
* оценка;
* коэффициент правильных ответов.

Необходимо предусмотреть следующую группу пользователей образовательной платформы:

* студент;
* администратор;
* автор.

Студент должен иметь следующие функциональные возможности в системе:

* авторизация;
* управление профилем;
* просмотр каталога курсов с учетом фильтрации, сортировки и поиска;
* просмотр подробной информации о курсе;
* запись на курс;
* прохождение курса;
* прохождение тестирования в рамках курса;
* просмотр прогресса по курсам и урокам;
* просмотр статистики по прохождению тестирований;
* просмотр результатов тестирования;
* оценка пройденного курса;
* получение сертификата о прохождении курса.

Автор курсов должен иметь следующие функциональные возможности:

* авторизация;
* управление профилем;
* просмотр своих курсов с учетом фильтрации, поиска и сортировки;
* управление своими курсами;
* управление контентом курсов (уроки и тесты);
* управление каталогом изображений;
* просмотр заявок на отображение курсов в каталоге с учетом статуса;
* отправка заявки на отображение курсов в каталоге.

Администратор должен иметь в системе следующие функциональные возможности:

* авторизация;
* управление профилем;
* просмотр информации о пользователях;
* управление доступом пользователей;
* управление категориями курсов;
* управление заявками на отображение курсов в каталоге;
* управление доступом курсов.

Необходимо предусмотреть следующие ограничения в информационной системе:

* записаться на курс может только авторизованный пользователь.

1.2 Проектирование бизнес-процессов предметной области

Диаграмма вариантов использования является исходным концептуальным представлением системы в процессе ее проектирования и разработки. Диаграмма вариантов использования содержит варианты использования системы, действия лиц и связи между ними. Она представляет собой граф, в вершинах которого расположены актеры и прецеденты. Связи между ними это разного вида отношения.

Диаграмма прецедентов представлена на рисунке 1.2.1.

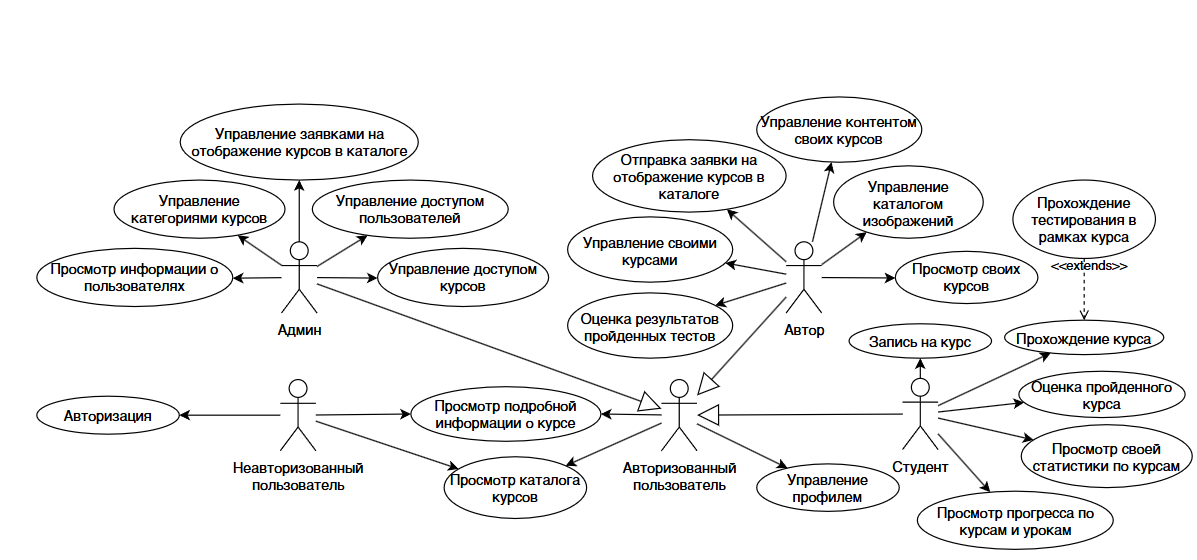


Рисунок 1.2.1 – Диаграмма прецедентов

1.3 Описание входной информации

Входная информация – ключевой элемент системы. Входные данные служат для создания записей в базы данных, хранения в системе и дальнейшего использования в работе системы.

Входной информацией для пользователей всех ролей является информация о пользователе, содержащая поля:

* имя;
* почта;
* пароль.

Входной информацией, вводимой администратором, является информация о категориях курсов, которая содержит поля:

* название курса.

Входная информация автора о курсе сдержит:

* заголовок;
* категория;
* описание;
* изображение;
* дополнительные блоки.

Входная информация об уроке содержит:

* заголовок;
* контент.

Входная информация о тесте содержит:

* заголовок;
* контент;
* таймер.

Входная информация с изображением для внесения в директорию содержит:

* назначение изображения;
* изображение из других директорий.

Помимо персональных данных студент вносит следующую входную информацию:

* ответ на задачу.

1.4 Описание выходной информации

Выходной информацией является сертификат о прохождении курса, сообщение со сбросом пароля и статистика пользователя по завершенным и начатым курсам.

Описание выходных документов представлено в таблице 1.4.1.

Таблица 1.4.1 – Описание выходных документов

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование документа (шифр) | Периодичность выдачи документа | Количество экземпляров | Куда передаются |
| Восстановление аккаунта | После запроса | 1 | Почта пользователя |
| Сертификат о прохождении курса | После завершения курса | 1 | Аккаунт пользователя |
| Статистика по результатам тестов | После запроса | 1 | Страница «Моя статистика» |
| Прогресс по курсам | После запроса | 1 | Страница «Моя статистика» |

Шаблон письма о восстановлении пароля представлен на рисунке 1.4.1.

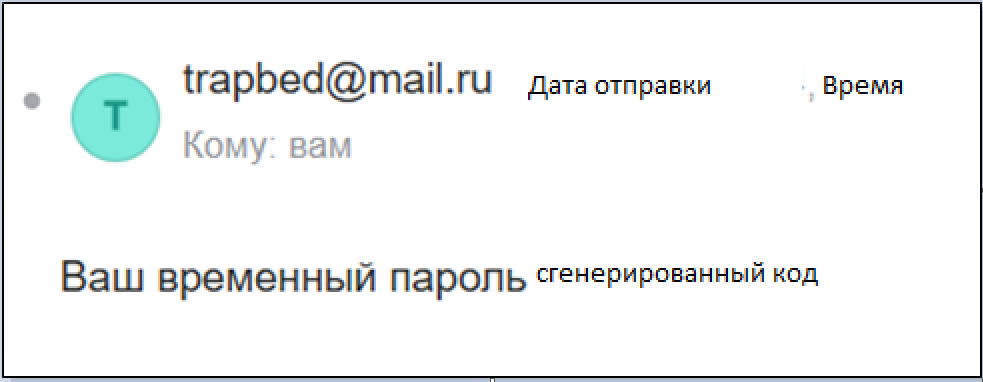


Рисунок 1.4.1 – Шаблон письма о восстановлении пароля

Шаблон сертификата о прохождении курса представлен в приложении А на рисунке А.1.

1.5 Общие требования к программному продукту

Данный программный продукт предназначен для формирования и прохождения образовательных курсов.

Система предназначена для использования следующими группами пользователей:

* администратор: управление доступом пользователей и курсов, создание, редактирование, удаление категорий;
* автор курсов: управление контентом курсов, уроков, тестовых блоков, отправка заявок на доступ курса в каталоге;
* студент: прохождение курсов и просмотр контента курсов, отслеживание своей статистики.
* у всех пользователей имеется личный кабинет.

Основные цели проекта:

* упрощение процесса прохождения онлайн-обучения;
* упрощение процесса управления контентом курсов;
* автоматизация процесса оценки знаний;
* обеспечение доступности обучающих материалов.

Функциональные требования к системе:

* управление курсами: возможность добавления, редактирования и удаления уроками;
* разграничение доступа и ролей: доступ пользователей к возможностям и страницам зависит от его роли;
* управление материалами курса: автор курсов создает и редактирует материалы курса, уроков и тестов к нему;
* управление доступом курса: администратор может управлять доступом курса (скрывать и восстанавливать), пользователей и управлять категориями.

Нефункциональные требования:

* безопасность: система должна быть защищена от несанкционированного доступа;
* юзабилити: система должна быть простой и понятной в использовании.

Требования к пользовательскому интерфейсу

Основные экраны:

* станица авторизации и регистрации: отображение форм авторизации и регистрации;
* главная страница: отображение информации о платформе;
* каталог: отображение списка курсов с учетом фильтрации по категориям, возможностью сортировки по новизне, популярности и алфавиту;
* страница курса: подробная информация о курсе с заголовками уроков и тестов, информацией о прохождении курса студентом;
* страница статистики пользователя: статистика по прохождению тестов, отслеживание прогресса по курсам;
* страница урока: материалы занятия;
* страница тестирования: материалы тестирования;
* панель автора курсов: интерфейс для создания курсов, уроков и материалов к ним, отправление заявок на вывод данных;
* панель администратора: интерфейс для управления категориями, доступом курсов и пользователей.

Система должна включать меню навигации со следующими разделами:

* администратор: пользователи, категории, курсы, аккаунт;
* автор: курсы, изображение курсов, заявки на доступ, аккаунт;
* студент: главная, аккаунт, моя статистика.

Требования к юзабилити:

* простой и понятный интерфейс для быстрого доступа к курсам и их материалам.

Конечный программный продукт должен иметь:

‒достаточно удобный интерфейс;

‒надежное хранение, обработка, редактирование данных;

‒ достаточно понятная и полная документация;

‒ операции, описанные в предметной области.

Требуются следующие программные обеспечения для корректной работы программы:

‒СУБД MySQL 5.7;

‒локальный сервер с поддержкой модулей Apache 2.4, PHP 8.0.

Минимальные требования к аппаратному обеспечению для производительной работы программного продукта:

‒ операционная система не ниже Windows 10;

‒ разрядность системы x64;

‒ ОЗУ от 2ГБ;

‒ монитор;

‒ клавиатура;

‒ манипулятор типа «мышь».

1.6 Описание структуры базы данных

Описание структуры базы данных предоставляет информацию о полях в создаваемой базе данных, описание таблиц, полей, находящихся в них, их тип, длину и примечания к ним.

Для работы с базой данных необходимо использовать СУБД MySQL.

Структура базы данных изображена на рисунке 1.6.1.

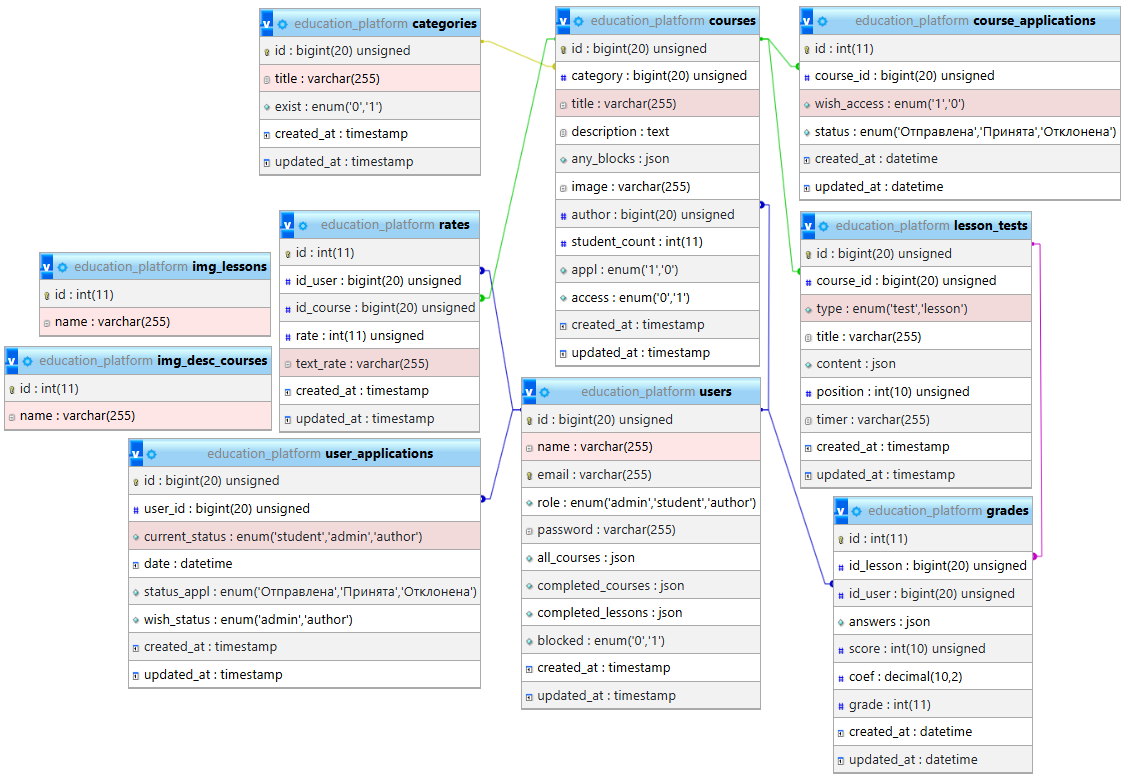


Рисунок 1.6.1 – Структура базы данных

Описание структуры базы данных представлено в таблицах 1.6.1-1.6.10.

Таблица 1.6.1 - «Users» (Пользователи)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор пользователя | ID | BIGINT(20) | Первичный ключ |
| Имя | NAME | VARCHAR(255) |  |
| Почта | EMAIL | VARCHAR(255) |  |
| Роль | ROLE | ENUM(‘admin’, ‘author’, ‘student’) |  |
| Пароль | PASSWORD | VARCHAR(255) |  |
| Начатые курсы | ALL\_COURSES | JSON |  |
| Завершенные курсы | COMPLETED\_  COURSES | JSON |  |
| Пройденные уроки | COMPLETED\_  LESSONS | JSON |  |
| Статус блокировки | BLOCKED | ENUM(‘0’, ‘1’) |  |

Таблица 1.6.2 – «Categories» (Категория курса)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор категории | ID | BIGINT(20) | Первичный ключ |
| Название категории | TITLE | VARCHAR(255) |  |
| Наличие категории | EXIST | ENUM(‘0’, ‘1’) |  |

Таблица 1.6.3 – «Courses» (Курс)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор курса | ID | BIGINT(20) | Первичный ключ |
| Категория курса | CATEGORY | BIGINT(20) | Внешний ключ к Categories |
| Заголовок курса | TITLE | VARCHAR(255) |  |
| Описание | DESCRIPTION | TEXT |  |
| Дополнительные блоки | ANY\_BLOCKS | JSON |  |
| Изображение | IMAGE | VARCHAR(255) |  |
| Автор курса | AUTHOR | BIGINT(20) | Внешний ключ к Users |
| Количество студентов | STUDENT\_COUNT | INT(11) |  |
| Статус заявки на вывод | APPL | ENUM(‘0’, ‘1’) |  |
| Отображение | ACCESS | ENUM(‘0’, ‘1’) |  |

Таблица 1.6.4 – «User\_applications» (Заявка пользователя)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор заявки | ID | BIGINT(20) | Первичный ключ |
| Идентификатор пользователя | USER\_ID | BIGINT(20) | Внешний ключ к Users |
| Текущая роль пользователя | CURRENT\_STATUS | ENUM(‘admin’, ‘author’, ‘student’) |  |
| Дата заявки | DATE | DATETIME |  |
| Статус заявки | STATUS\_APPL | ENUM(‘Отправлена’, ‘Принята’, ‘Отклонена’) |  |
| Желаемая роль | WISH\_STATUS | ENUM(‘admin’, ‘author’) |  |

Таблица 1.6.5 – «Lesson\_tests» (Уроки и тесты)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор урока | ID | BIGINT(20) | Первичный ключ |
| Идентификатор курса | COURSE\_ID | BIGINT(20) | Внешний ключ к Courses |
| Тип | TYPE | ENUM(‘test’, ‘lesson’) |  |
| Заголовок | TITLE | VARCHAR(255) |  |
| Контент | CONTENT | TEXT |  |
| Позиция | POSITION | INT(10) |  |
| Таймер | TIMER | VARCHAR(255) |  |

Таблица 1.6.6 – «Grades» (Оценки)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор теста | ID | BIGINT(20) | Первичный ключ |
| Идентификатор урока | ID\_LESSON | BIGINT(20) | Внешний ключ к Lesson\_tasks |
| Идентификатор пользователя | ID\_USER | BIGINT(20) |  |
| Ответы | ANSWERS | JSON |  |
| Баллы | SCORE | INT(10( |  |
| Коэффициент | COEF | DECIMAL(10,2) |  |
| Оценка | GRADE | INT(10) |  |

Таблица 1.6.7 – «Course\_applications» (Заявка на вывод)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор заявки | ID | BIGINT(20) | Первичный ключ |
| Идентификатор курса | COURSE\_ID | BIGINT(20) | Внешний ключ к Courses |
| Желаемый статус | WISH\_STATUS | ENUM(‘0’, ‘1’) |  |
| Статус заявки | STATUS | ENUM(‘Отправлена’, ‘Принята’, ‘Отклонена’) |  |

Таблица 1.6.8 – «Rates» (Оценки и отзывы)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор заяки | ID | BIGINT(20) | Первичный ключ |
| Идентификатор пользователя | ID\_USER | BIGINT(20) | Внешний ключ к Users |
| Идентификатор курса | ID\_COURSE | BIGINT(20) | Внешний ключ к Courses |
| Оценка | RATE | INT(11) |  |
| Отзыв | TEXT\_RATE | VARCHAR(255) |  |

Таблица 1.6.9 – «Img\_desc\_courses» (Изображения к курсам)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор заяки | ID | BIGINT(20) | Первичный ключ |
| Имя файла | NAME | VARCHAR(255) |  |

Таблица 1.6.10 – «Img\_lessons» (Изображения к урокам)

|  |  |  |  |
| --- | --- | --- | --- |
| Содержание поля | Имя поля | Тип, длина | Примечания |
| Идентификатор заяки | ID | BIGINT(20) | Первичный ключ |
| Имя файла | NAME | VARCHAR(255) |  |

1.7 Контрольный пример

Контрольный пример предоставляет входные данные для проверки работы сайта. Входные данные контрольного примера представлены в таблицах 1.7.1-1.7.6 и в приложении Б, таблицах Б.1-Б.2.

Таблица 1.7.1 - Входные данные контрольного примера «Пользователи»

|  |  |  |
| --- | --- | --- |
| Имя | Почта | Пароль |
| Петров Константин Иванович | pekoiv@mail.ru | 12345678 |
| Арзамасова Лиза Петровна | lizipe@gmail.com | lizipe |
| Мазур Кирилл Андреевич | kiri@mail.ru | kirikiri |
| Степанов Леша Дмитриевич | stled@gmail.com | MinskTime |
| Фролова Лена Николаевна | froleni@mail.com | froleni |

Таблица 1.7.2 - Входные данные контрольного примера «Заявки пользователей на смену роли»

|  |  |  |
| --- | --- | --- |
| Имя пользователя | Текущая роль | Желаемая роль |
| pekoiv@mail.ru | Студент | Автор |
| lizipe@gmail.com | Студент | Админ |
| kiri@mail.ru | Автор | Админ |
| stled@gmail.com | Студент | Автор |
| vitya@mail.com | Автор | Админ |

Таблица 1.7.3- Входные данные контрольного примера «Заявка курса на вывод в каталог»

|  |  |
| --- | --- |
| Наименование курса | Желаемый статус |
| История Византии | Вывод |
| Скорочтение | Скрыть |
| Основы программирования | Скрыть |
| Греческий язык | Вывод |
| Привлекательный дизайн | Вывод |

Таблица 1.7.4 - Входные данные контрольного примера «Оценки»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование урока | Имя пользователя | Баллы | Коэффициент | Оценка |
| Греческий язык для начинающих | Петров Константин Иванович | 1 | 1.00 | 5 |
| Школьная программа по истории Византии | Пазиев Саша Игоревич | 3 | 0.66 | 4 |
| Скорочтение | Иванов Кирилл Юрьевич | 4 | 0.88 | 5 |
| Основы основ программирования | Степанов Леша Дмитриевич | 3 | 0.97 | 5 |
| Привлекающий дизайн | Фролова Лена Николаевна | 1 | 0.40 | 3 |

Таблица 1.7.5 – Выходные данные контрольного примера «Сертификат о прохождении курса»

|  |  |  |
| --- | --- | --- |
| Имя пользователя | Наименование курса | Дата выдачи |
| Петров Константин Иванович | Греческий язык для начинающих | 10.05.2025 |
| Пазиев Саша Игоревич | Школьная программа по истории Византии | 12.05.2025 |
| Иванов Кирилл Юрьевич | Скорочтение | 11.05.2025 |

Таблица 1.7.6 – Выходные данные контрольного примера «Восстановление аккаунта»

|  |  |  |
| --- | --- | --- |
| Почта пользователя | Временный пароль | Дата выдачи |
| pekoiv@mail.ru | drvBDHQ3 | 21.04.2025 |
| pazalig@gmail.com | hjuALSTU | 30.04.2025 |
| ivkiyu@mail.ru | jmuvzFQ8 | 03.05.2025 |

Входные данные контрольного примера для создания курса представлены в приложении Б, таблице Б.1.

Входные данные контрольного примера «Уроки и тесты» представлены в приложении Б, таблице Б.2.

2 Экспериментальный раздел

2.1 Описание программы

Для разработки веб-приложения были использованы веб-технологии HTML5, CSS3, языки программирования PHP 8.1, JavaScript, фреймворк Laravel 10.

Laravel ‒ это популярный PHP-фреймворк, который следует принципам MVC. Он предоставляет удобный и элегантный синтаксис, что ускоряет процесс разработки. Laravel включает множество встроенных функций, таких как маршрутизация, миграции базы данных и система аутентификации, что делает его мощным инструментом для создания веб-приложений.

PHP ‒ это серверный язык программирования, который идеально подходит для веб-разработки. Он широко используется, имеет обширную документацию и большое сообщество разработчиков. PHP отлично интегрируется с различными базами данных и серверными технологиями, что делает его универсальным выбором для создания динамических веб-приложений.

Bootstrap ‒ это популярный фреймворк для разработки адаптивных веб-интерфейсов. Он предоставляет готовые к использованию CSS и JavaScript компоненты, которые позволяют быстро создавать красивые и функциональные веб-страницы. Использование Bootstrap значительно ускоряет процесс разработки интерфейсов и обеспечивает кроссбраузерную совместимость.

Описание программы включает в себя описание модулей. Описание модулей- таблица, отражающая все используемые классы и их содержимое- методы. Прописывается файл и хранящиеся в нем методы с кратким описанием функций, происходящих внутри класса (модуля). Описание модулей представлено в таблице 2.1.1. Физическая структура представлена в приложении В на рисунке В.1.

Таблица 2.1.1 – Описание модулей (классов)

|  |  |
| --- | --- |
| Методы | Назначение |
| 1 | 2 |
| Класс AuthController.php | |
| signup(Request $request) | Регистрация по полученным данным из формы. Содержит валидацию и шифрование пароля. |
| login\_db(Request $request) | Авторизация по данным из формы |
| logout() | Выход из аккаунта |
| recover\_acc(request $request) | Восстановление аккаунта по почте. Включает создание пароля из рандомной комбинации цифр и букв и отправка на почту пользователя. |
| CategoryController.php | |
| categories\_admin() | Вывод категорий на страницу админа |
| create\_category(Request $request) | Создание категории по данным из формы |
| change\_exist\_category($exist, $id) | Изменение доступа категорий курсов |
| edit\_cat\_show($id) | Данные для внесения в форму изменения информации о категории курса |
| edit\_cat(Request $request) | Изменение данных категории курса по id |
| CourseController.php | |
| main() | Вывод первых пяти популярных курсов |
| main\_courses(Request $request) | Вывод всех курсов в каталог с учетом фильтрации, сортировки и поиска |
| one\_course\_main($id\_course) | Получение данных курса и вывод на страницу одного курса |
| get\_all\_admin() | Получение информации о курсах в панель администратора с расширенными данными |
| change\_access\_course($access, $id\_course) | Изменение статуса доступа курсов для вывода |

Продолжение таблицы 2.1.1.

|  |  |
| --- | --- |
| 1 | 2 |
| author\_more\_info\_course($id) | Получение подробной информации о курсе, тестах и уроках для вывода в административную панель |
| create\_course\_show() | Получение доступных категорий для создания курса и вывод формы для его создания |
| create\_course(Request $request) | Получение данных из формы для создания курса, создание курса с проведением валидации |
| update\_course\_show($id) | Получение информации об одном курсе для дальнейшего изменение его данных |
| update\_course(Request $request) | Получение данных из формы и изменение их в базе данных |
| data\_for\_create\_course($id) | Вывод информации о курсе и его содержимом в виде уроков и тестов для дальнейшего их изменения |
| data\_for\_create\_test($id) | Вывод данных о курсе и количестве уже имеющихся тестов на страницу для создания нового теста |
| send\_access($course\_id, $wish\_access) | Метод для передачи данных о курсе и желаемом статусе доступа в форму с заявками на вывод курса в каталог или наоборот запрет на вывод |
| application\_courses() | Получение данных о своих заявках для вывода в панели автора |
| get\_course\_applications() | Метод для получения всех заявок и вывода их на стороне администратора |
| set\_access($id\_course, $id\_appl, $wish, $act) | Изменение статуса заявки и статуса доступа по определенной заявке |
| create\_test\_db(Request $request) | Метод для создания теста на основе данных формы с обработкой и созданием json-объекта с контентом |
| send\_rate\_course(Request $request) | Метод, сохраняющий отзыв пользователя в БД. |
| send\_img\_to\_desc\_course(Request $request) | Метод, добавляющий изображение в БД и директорию. |
| LessonController.php | |
| images\_lesson() | Вывод формы для добавления изображения в директорию |
| add\_to\_dir(Request $request) | Метод для добавления изображения в директорию, содержащий валидацию формата файла |
| create\_lesson(Request $request) | Метод для создания урока на основе данных формы содержащий обработку и создание json-объекта |
| remove\_lesson($id\_lesson, $id\_course) | Метод для удаления урока |
| one\_lesson($id) | Вывод информации по одному уроку на стороне автора |
| one\_lesson\_student($id, $course) | Вывод информации по одному уроку на стороне студента |
| check\_test\_student(Request $request) | Метод, проверяющий ответы пользователя и оценивающий их. |
| certificate($id\_course) | Метод, собирающий данные для формирования сертификата. |
| UserController.php | |
| account\_info() | Метод для вывода информации о пользователе в его аккаунт |
| edit\_account(Request $request) | Метод, изменяющий данные аккаунта по переданным данным из формы |
| all\_users\_admin() | Вывод информации о всех пользователях |
| change\_blocked($id\_user, $blocked) | Изменение статуса доступа пользователя |
| users\_appl() | Вывод всех заявок на смену роли пользователя |
| new\_pass(Request $request) | Метод, получающий новый пароль из формы и смена его в базе данных |
| change\_role($id\_user, $id\_appl, $role, $status\_appl) | Изменение статуса заявки на смену роли и изменение роли пользователя в базе данных |
| start\_study($id\_course) | Изменение json-объекта со всеми курсами пользователя, открывающий доступ к курсу |
| complete\_course($id\_course) | Метод, изменяющий json-объект с завершенными курсами пользователя |
| my\_statistics() | Метод, собирающий данные о пользователе для формирования статистики и прогресса. |

2.2 Протокол тестирования программного продукта

Целью создания протокола тестирования является проверка ответов системы на неожиданные вводимые данные. Тестирование проводится для выявления ошибок и решения их в дальнейшем. Это важный этап в создании программного продукта. Тестирование системы лучше всего проводить по окончанию этапа создания одного блока, после чего можно переходить к следующему.

Протестирована регистрация на корректных данных контрольного примера. Протокол тестирования представлен в таблице 2.2.1.

Таблица 2.2.1 – Протокол тестирования регистрации на корректных данных

|  |  |
| --- | --- |
| Поле | Описание |
| Дата теста | 09.05.2025 |
| Приоритет тестирования *(Низкий/ Средний/ Высокий)* | Высокий |
| Заголовок/ название теста | Проверка регистрации на корректных данных |
| Резюме испытания | Успешная попытка регистрации, вход в профиль. |
| Этапы теста | 1. Нажатие на кнопку «Зарегистрироваться» в шапке сайта;  2. Ввод корректных данных для регистрации;  3. Нажатие на кнопку «Зарегистрироваться»; в нижней части формы  4. Получение сообщения об успехе и переадресация на главную страницу. |
| Тестовые данные | Входные данные:  1. Имя: Степанов Леша Дмитриевич;  2. Почта: stled@gmail.com;  3. Пароль: MinskTime. |
| Ожидаемый результат | Пользователь должен успешно зарегистрироваться, войти в свой профиль и быть переадресованным на главную страницу. |
| Фактический результат | Пользователь переадресован на главную, выведено сообщение об успешной регистрации, вход в профиль. |

Результат тестирования представлен на рисунке 2.2.1.

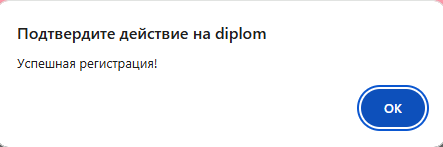


Рисунок 2.2.1 – Сообщение об успешной регистрации

Проведено тестирование формы регистрации на основе данных контрольного примера. Протокол тестирования представлен в таблице 2.2.2.

Таблица 2.2.2 – Протокол тестирования регистрации на некорректных данных

|  |  |
| --- | --- |
| Поле | Описание |
| Дата теста | 09.05.2025 |
| Приоритет тестирования *(Низкий/ Средний/ Высокий)* | Высокий |
| Заголовок/ название теста | Проверка регистрации на некорректных данных |
| Резюме испытания | Безуспешная регистрация с выводом предупредительного сообщения для пользователя. |
| Этапы теста | 1. Нажатие на кнопку «Зарегистрироваться» в шапке сайта;  2. Ввод некорректных данных для регистрации;  3. Нажатие на кнопку «Зарегистрироваться»; в нижней части формы  4. Получение сообщения об ошибке;  5. Возврат к форме с сохранением предыдущих данных. |
| Тестовые данные | Входные данные:  1. Имя: Пазиев Саша Игоревич;  2. Почта: pazalig@gmail.com;  3. Пароль: dimkalovv. |
| Ожидаемый результат | Пользователь должен получить сообщение об ошибке с возвратом к форме с сохранением предыдущих данных. |
| Фактический результат | Пользователь получил сообщение об ошибке, был возвращен к форме регистрации с предыдущими данными. |

Результат тестирования изображен на рисунке 2.2.2.

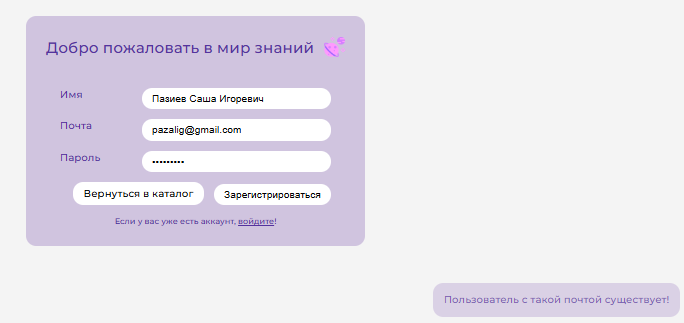


Рисунок 2.2.2 – Возврат к форме с данными и сообщение об ошибке

Протокол тестирования авторизации на корректных данных контрольного примера представлен в таблице 2.2.3.

Таблица 2.2.3 – Протокол тестирования авторизации на корректных данных

|  |  |
| --- | --- |
| Поле | Описание |
| Дата теста | 09.05.2025 |
| Приоритет тестирования *(Низкий/ Средний/ Высокий)* | Высокий |
| Заголовок/ название теста | Проверка ответа формы авторизации на корректных данных. |
| Резюме испытания | Успешная попытка авторизации студента, вход в профиль. |
| Этапы теста | 1. Нажатие на кнопку «Войти» в шапке сайта;  2. Ввод корректных данных авторизации;  3. Нажатие на кнопку «Войти»; в нижней части формы  4. Получение сообщения об успехе;  5. Переадресация на главную страницу. |
| Тестовые данные | Входные данные:  1. Почта: stled@gmail.com;  2. Пароль: MinskTime. |
| Ожидаемый результат | Успешная авторизация пользователя с выводом сообщения об успешной авторизации, переадресации на главную страницу. |
| Фактический результат | Пользователь переадресован на главную, выведено сообщение об успешной авторизации, вход в профиль. |

Результат тестирования представлен на рисунке 2.2.3.

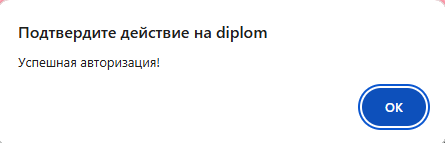


Рисунок 2.2.3 – Сообщение об успешной авторизации

Протокол тестирования авторизации на некорректных данных представлен в таблице 2.2.4.

Таблица 2.2.4 – Протокол тестирования авторизации на некорректных данных

|  |  |
| --- | --- |
| Поле | Описание |
| Дата теста | 09.05.2025 |
| Приоритет тестирования *(Низкий/ Средний/ Высокий)* | Высокий |
| Заголовок/ название теста | Проверка авторизации на некорректных данных |
| Резюме испытания | Безуспешная авторизация с выводом предупредительного сообщения для пользователя. |
| Этапы теста | 1. Нажатие на кнопку «Войти» в шапке сайта;  2. Ввод некорректных данных для авторизации;  3. Нажатие на кнопку «Войти»; в нижней части формы  4. Получение сообщения об ошибке;  5. Возврат к форме с сохранением предыдущих данных. |
| Тестовые данные | Входные данные:  1. Почта: pazalig@gmail.com;  2. Пароль: ew76. |
| Ожидаемый результат | Пользователь должен получить сообщение об ошибке с возвратом к форме с сохранением предыдущих данных. |
| Фактический результат | Пользователь получил сообщение об ошибке, был возвращен к форме авторизации с предыдущими данными. |

Результат тестирования формы авторизации на некорректных данных представлен на рисунке 2.2.4.

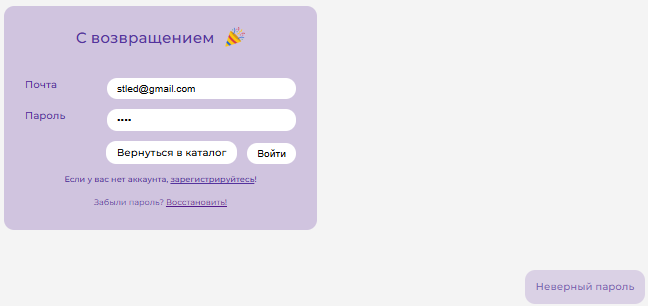


Рисунок 2.2.4 – Возврат к форме с данными и сообщение об ошибке

Протокол тестирования формы изменения данных профиля представлен в таблице 2.2.5.

Таблица 2.2.5 – Протокол тестирования формы изменения данных профиля на корректных данных

|  |  |
| --- | --- |
| Поле | Описание |
| Дата теста | 09.05.2025 |
| Приоритет тестирования *(Низкий/ Средний/ Высокий)* | Средний |
| Заголовок/ название теста | Проверка формы изменения данных профиля на корректных данных |
| Резюме испытания | Возврат на страницу профиля с обновленными данными и сообщением об успешном изменении данных. |
| Этапы теста | 1. Переход в профиль;  2. Изменение данных профиля. |
| Тестовые данные | Входные данные:  1. Имя: Степанов Алексей Дмитриевич;  2. Почта: stled@gmail.com. |
| Ожидаемый результат | Уведомление пользователя об успешной смене данных, обновление данных. |
| Фактический результат | Данные пользователя в профиле обновлены и получено сообщение об успешной смене данных |

Результат тестирования формы для изменения данных профиля на корректных данных изображен на рисунке 2.2.5.

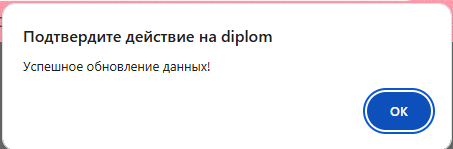


Рисунок 2.2.5 – Сообщения об успешном изменении данных

Протокол тестирования формы изменения данных профиля на некорректных данных представлен в таблице 2.2.6.

Таблица 2.2.6 – Протокол тестирования формы для изменения данных профиля на некорректных данных

|  |  |
| --- | --- |
| Поле | Описание |
| Дата теста | 09.05.2025 |
| Приоритет тестирования *(Низкий/ Средний/ Высокий)* | Средний |
| Заголовок/ название теста | Проверка формы изменения данных профиля на некорректных данных |
| Резюме испытания | Возврат на страницу профиля с выводом ошибки во введенных. |
| Этапы теста | 1. Переход в профиль;  2. Изменение данных профиля. |
| Тестовые данные | Входные данные:  1. Имя: Степанов Алексей Дмитриевич;  2. Почта:stled. |
| Ожидаемый результат | Уведомление пользователя с предупреждением. |
| Фактический результат | Данные пользователя в профиле не обновлены и получено сообщение об ошибке во введенных данных. |

Результат тестирования формы для изменения данных профиля на некорректных данных изображен на рисунке 2.2.6.

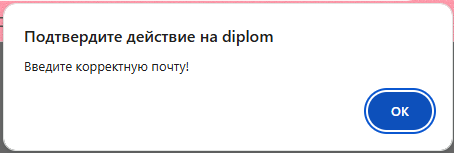


Рисунок 2.2.6 – Сообщения об ошибке во введенных данных

2.3 Руководство пользователя

Автоматизированная информационная система «Образовательная платформа» предназначена для упрощения процесса создания, прохождения и администрирования курсов путем разработки веб-приложения.

Для корректной работы системы необходимо:

* компьютер/ноутбук;
* доступ в интернет;
* клавиатура;
* компьютерная мышь/тачпад;
* модуль PHP\_8.1;
* модуль MySQL-5.7-Win10;
* модуль Apache\_2.4-PHP\_8.1.

Для работы с автоматизированной информационной системой необходимо перейти по ссылке «https://loweve.ct.ws» или ввести в поисковую строку адрес.

На главной странице доступен переход к страницам регистрации, авторизации и каталога курсов.

Начать изучение курса может только авторизованный пользователь.

Чтобы авторизоваться в системе, необходимо ввести почту и пароль в форму авторизации (рисунок 2.3.1).

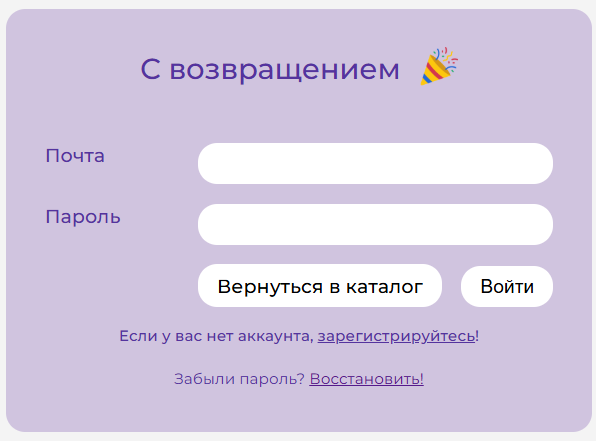


Рисунок 2.3.1 – Страница формы авторизации

Переход к форме регистрации доступен по кнопке «Зарегистрироваться» в навигационной панели или по ссылке в нижней части главной страницы (рисунок 2.3.2).

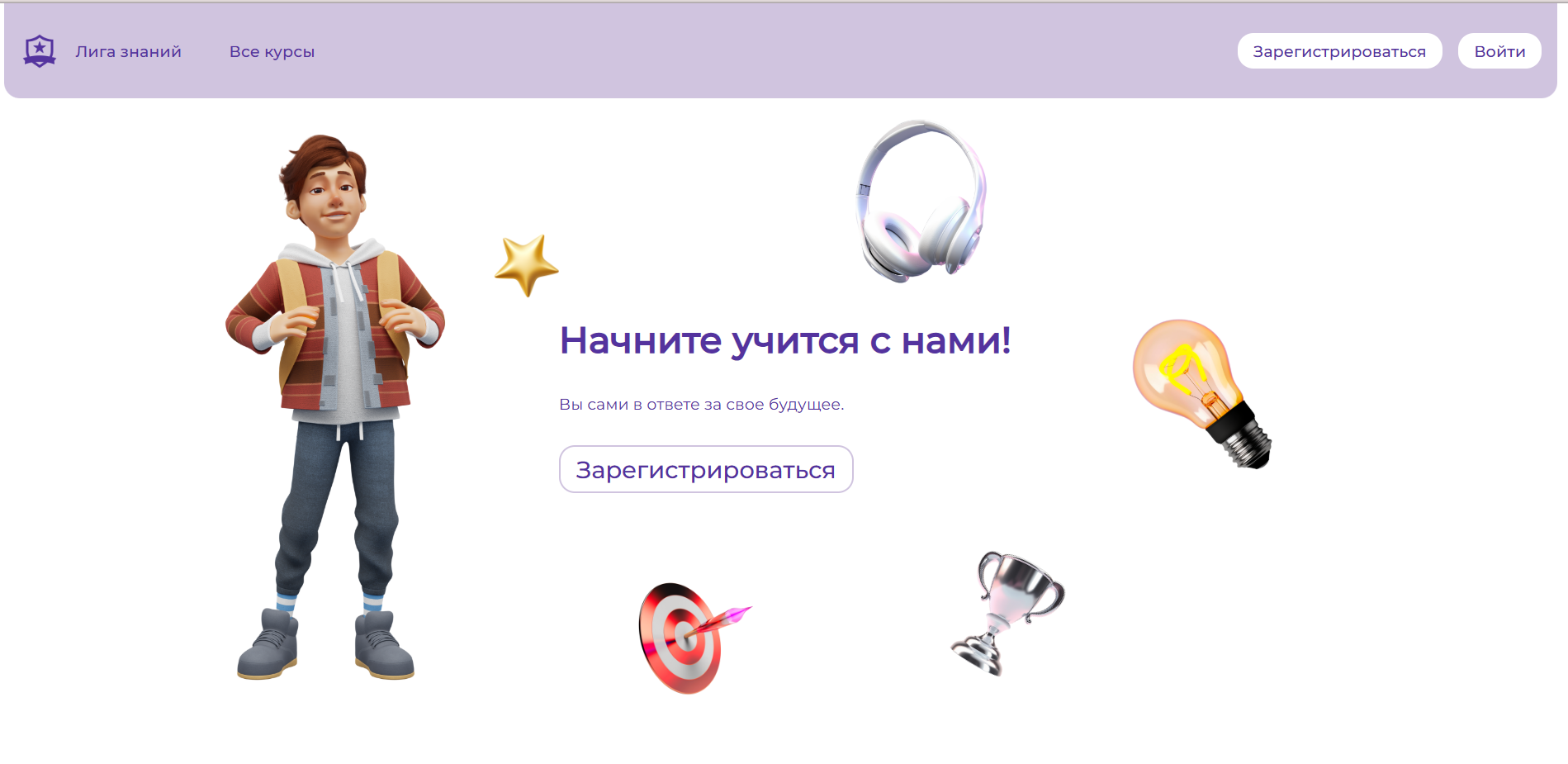


Рисунок 2.3.2 – Переход к странице «Зарегистрироваться»

Каталог доступен по переходу в пункт «Все курсы» или по нажатию на кнопку «Вернутся в каталог» в формах авторизации и регистрации (рисунок 2.3.3).



Рисунок 2.3.3 – Навигационная панель неавторизованного пользователя

После авторизации как студент, навигационная панель меняет вид и содержание (рисунок 2.3.4) Появляется пункт «Моя статистика», при переходе в который, пользователь видит свой прогресс по курсам и статистику по результатам пройденных тестирований (рисунок 2.3.5).



Рисунок 2.3.4 – Навигационная панель авторизованного пользователя

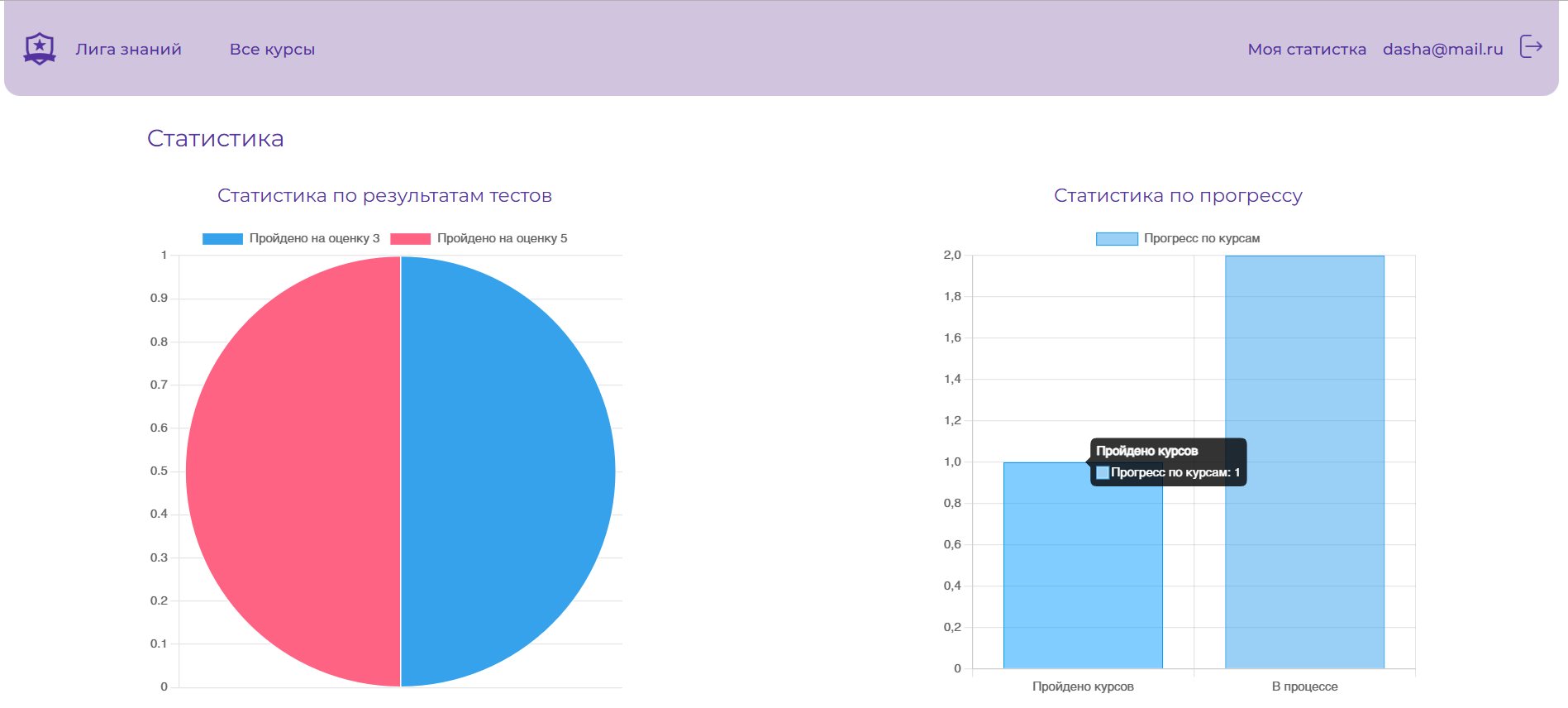


Рисунок 2.3.5 – Страница «Моя статистика»

Для начала изучения курса необходимо нажать на кнопку «Начать изучать». При успешной обработке заявки пользователь видит сообщение о том, что курс получен (рисунок 2.3.6).

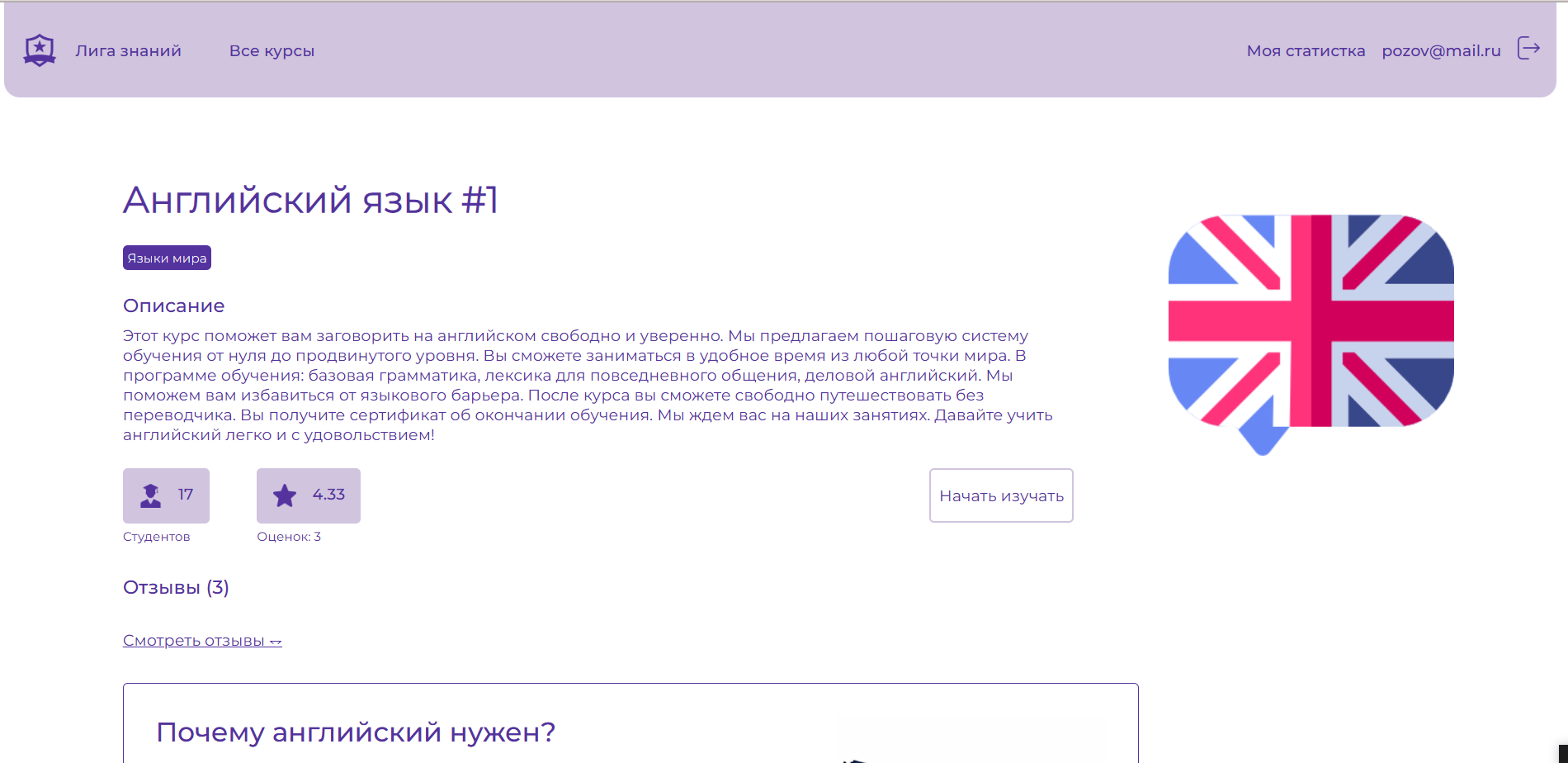


Рисунок 2.3.6 – Страница курса

После прохождения курса или теста прогресс курса меняется, и пройденные уроки и тесты подсвечиваются зеленым (рисунок 2.3.7).

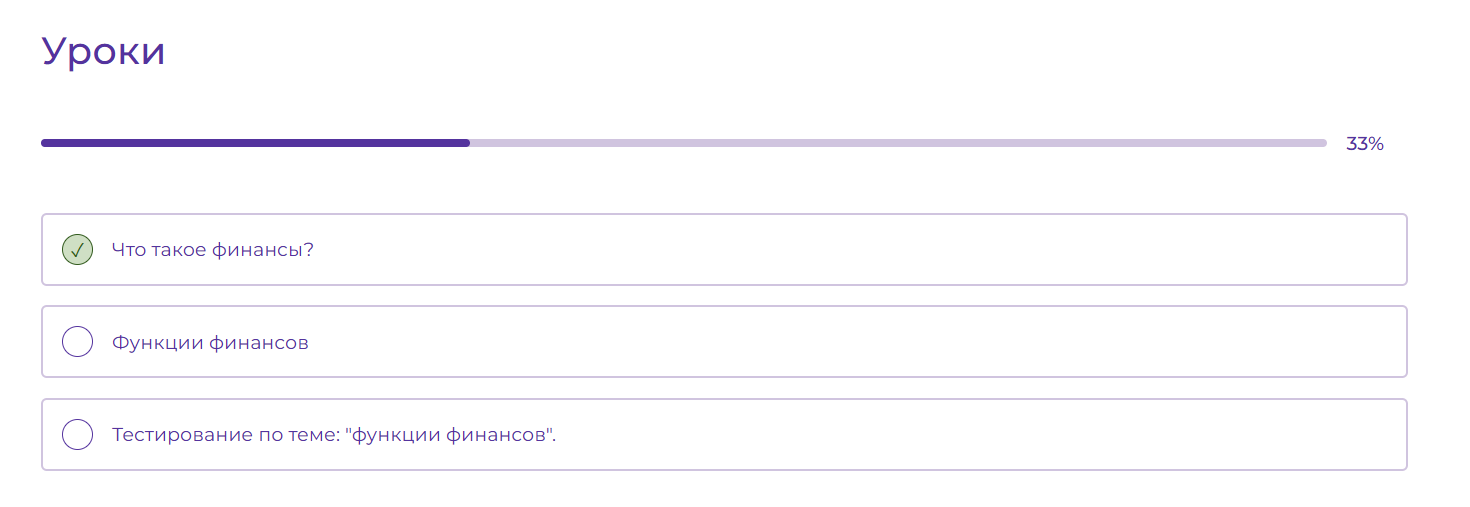


Рисунок 2.3.7 – Отображение прогресса

Все пользователи имеют возможность просматривать и изменять данные профиля (рисунок 2.3.8)

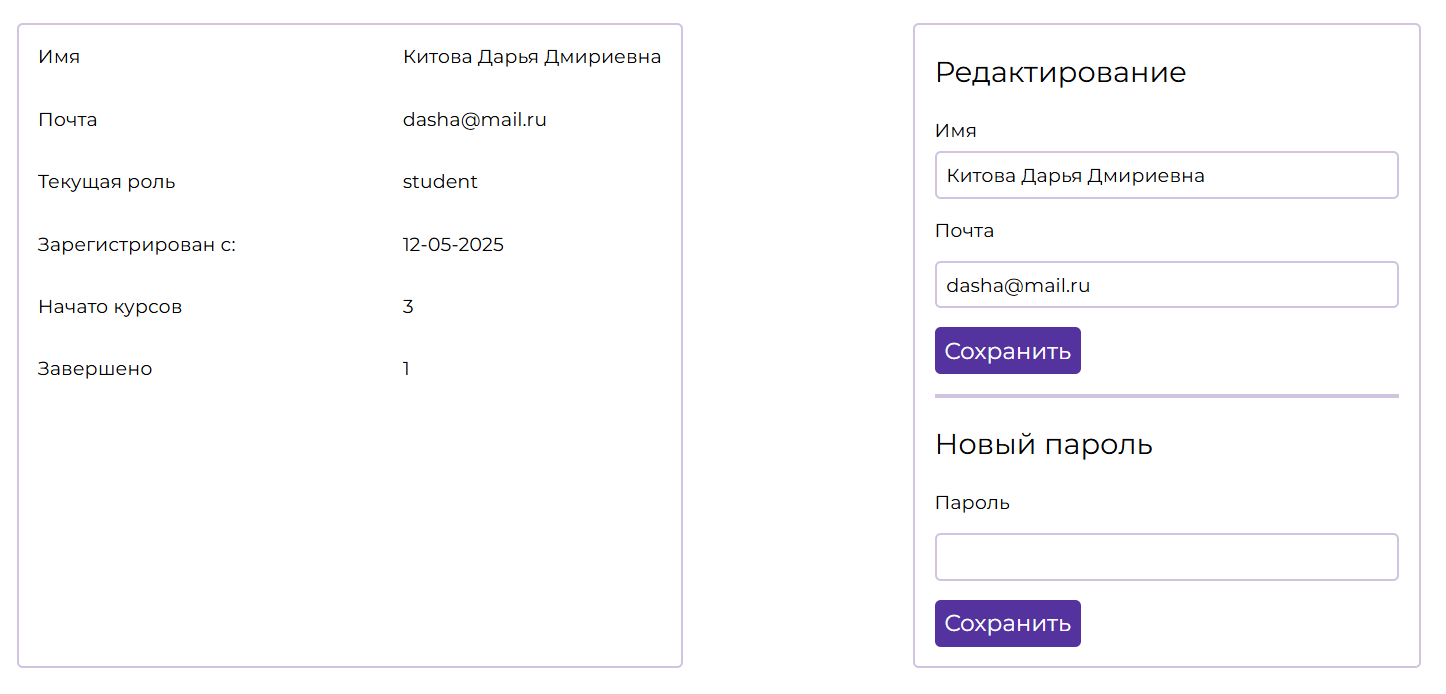


Рисунок 2.3.8 – Данные профиля

При авторизации в системе как администратор навигационная панель меняет свой внешний вид. После авторизации администратор попадает на страницу пользователей, где он может управлять статусом доступа пользователей (рисунок 2.3.9)

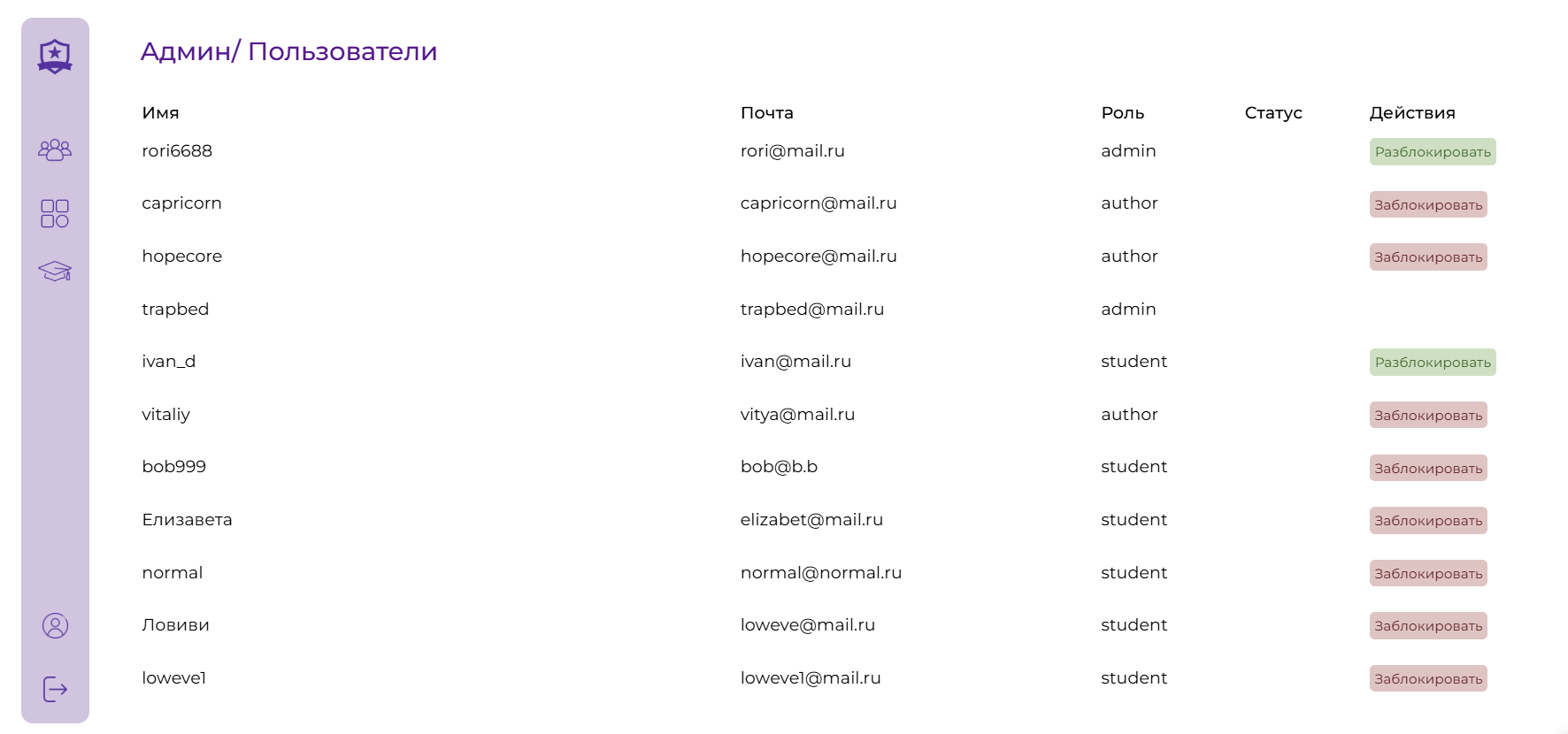


Рисунок 2.3.9 – Страница пользователей в админ панели

Администратор также может управлять категориями курсов и их статусами доступа (рисунок 2.3.10)

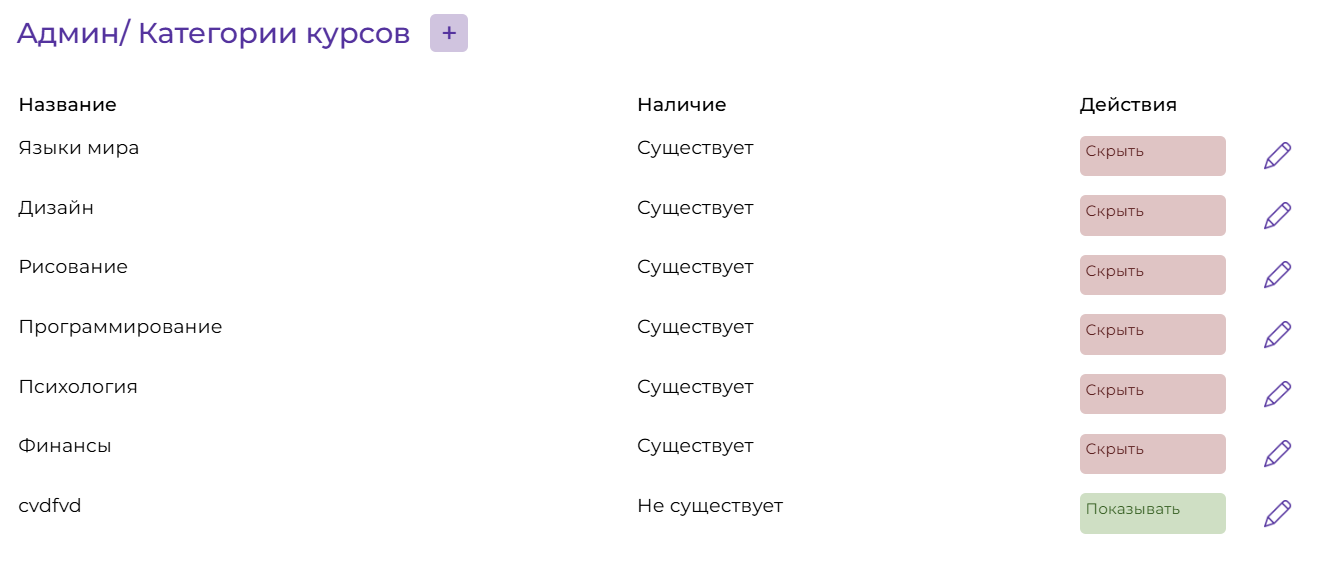


Рисунок 2.3.10 – Категории курсов в админ панели

Только администратор может управлять доступом курсов. Автор может только отправлять заявку на отображение курса (рисунок 2.3.11 - 2.3.12).

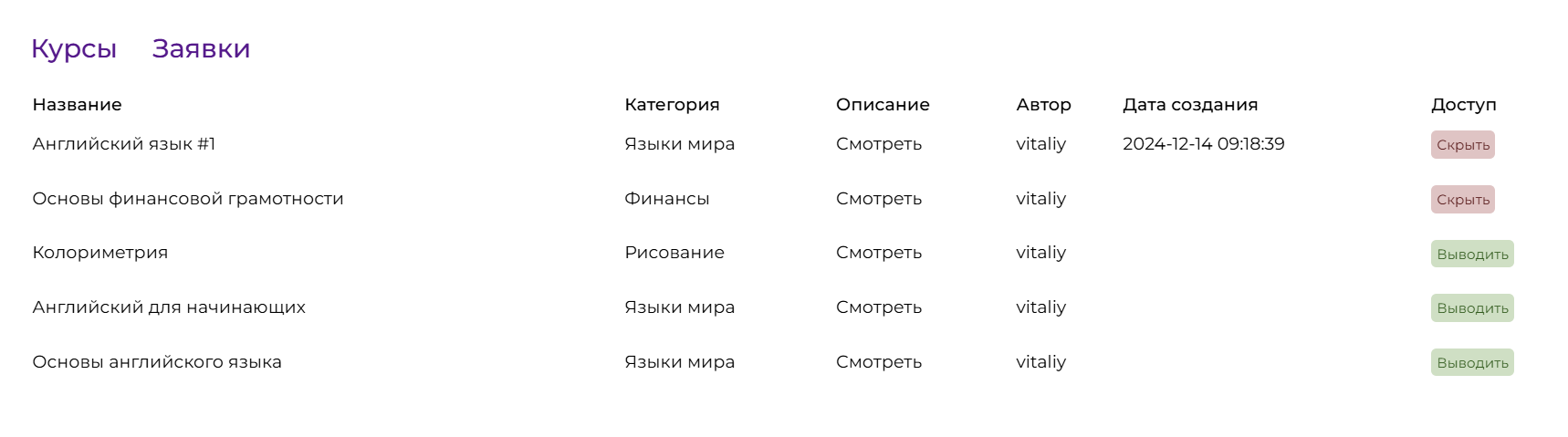


Рисунок 2.3.11 – Управление доступом курсов в админ панели

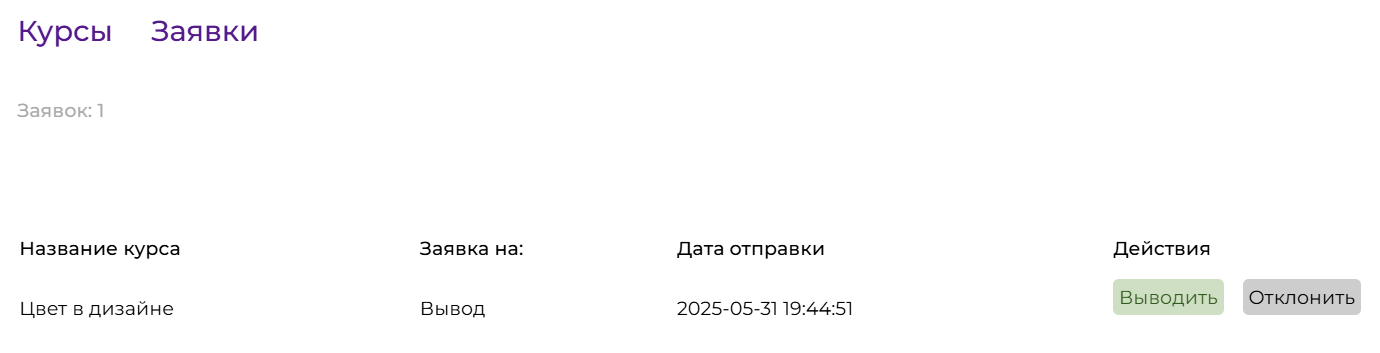


Рисунок 2.3.12 – Страница заявок на смену статуса доступа

Авторизовавшись как автор, пользователь оказывается на странице своих курсов. С этой страницы автор может отправлять заявку на доступ. Заявку на доступ можно отправить только когда курс заполнен. Скрыть курс невозможно, если проходящих курс студентов больше, чем завершивших. До заполнения курса невозможно установить статус «в доступе» (рисунок 2.3.13).

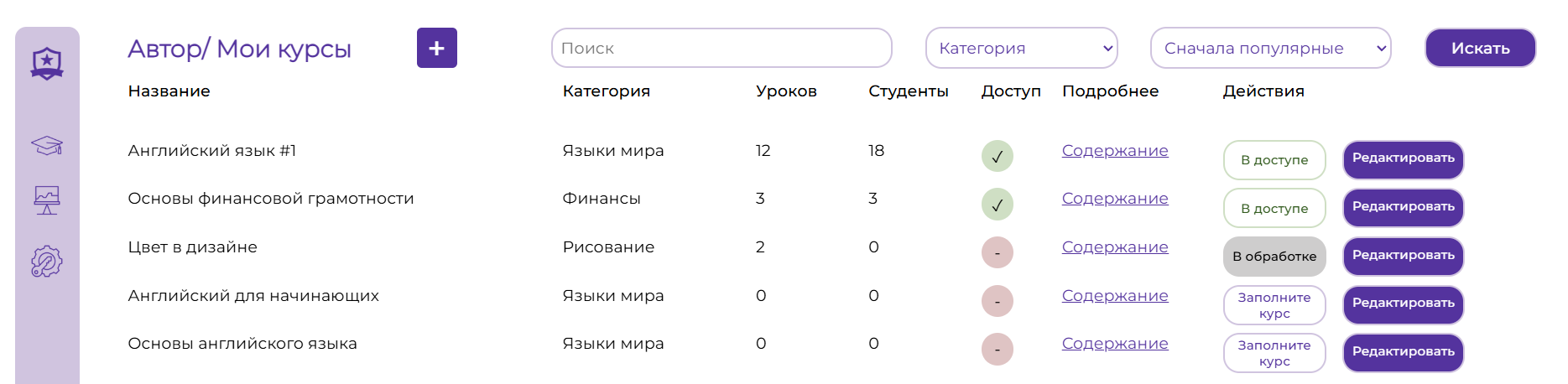


Рисунок 2.3.13

Для создания курса необходимо обязательно заполнить все обязательные поля (рисунок 2.3.14). Помимо обязательных блоков можно добавить блоки, раскрывающие содержание курса для привлечения студентов (рисунок 2.3.15)

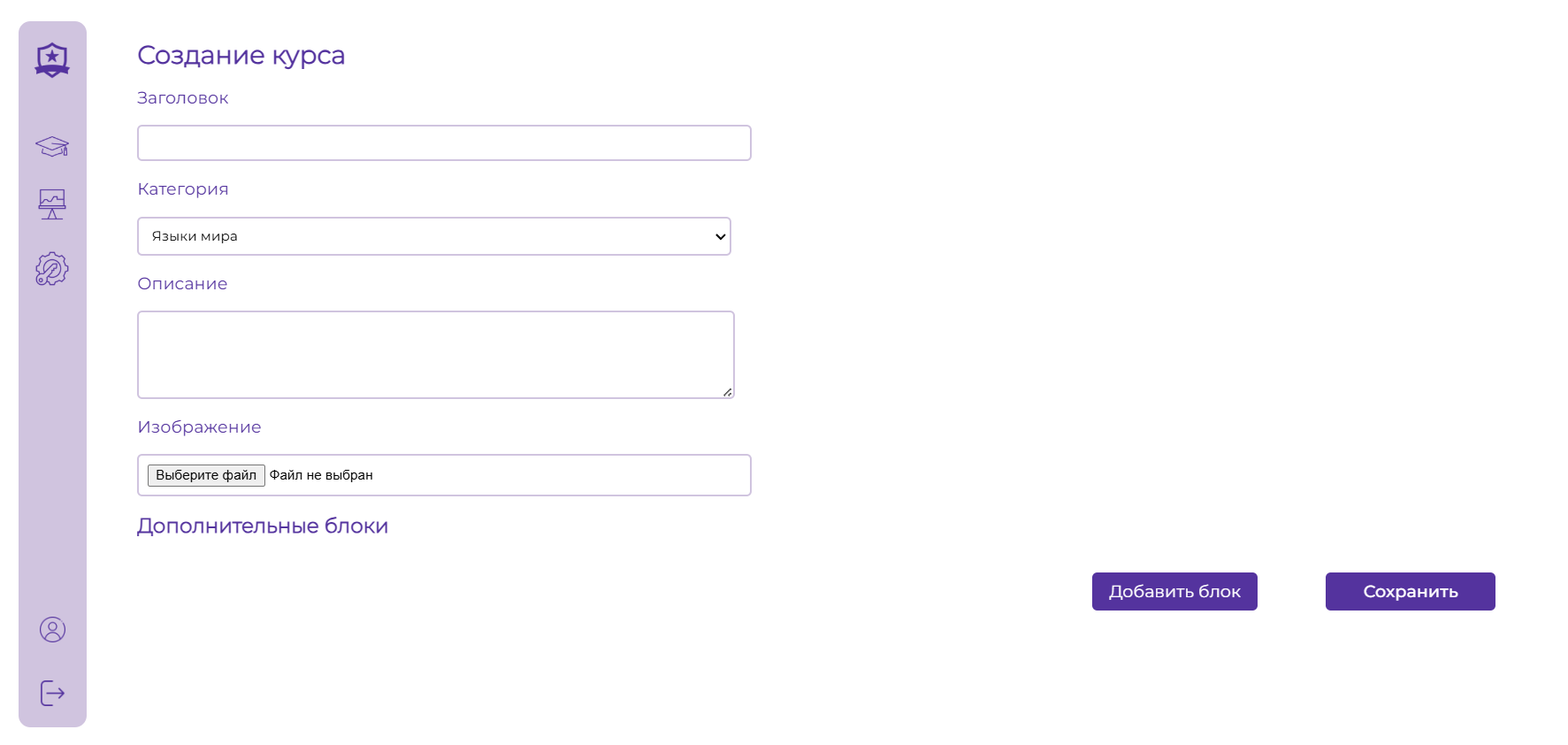


Рисунок 2.3.14 – Обязательные поля для создания курса

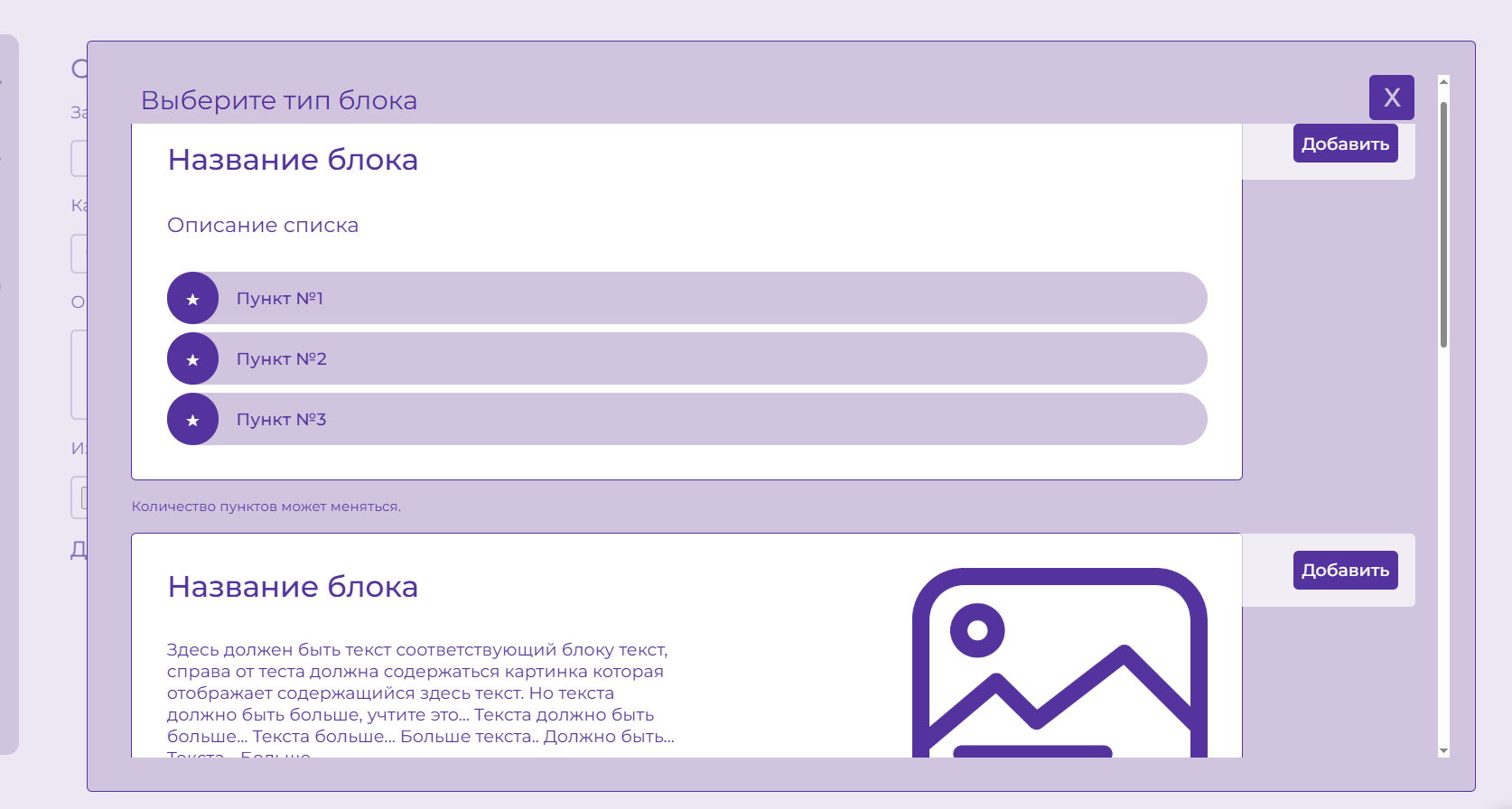


Рисунок 2.3.15 – Модальное окно для выбора дополнительных блоков

Что бы выбрать изображение для дополнительных блоков курса или контента урока необходимо сначала внести изображение в БД и директорию (рисунок 2.3.16).

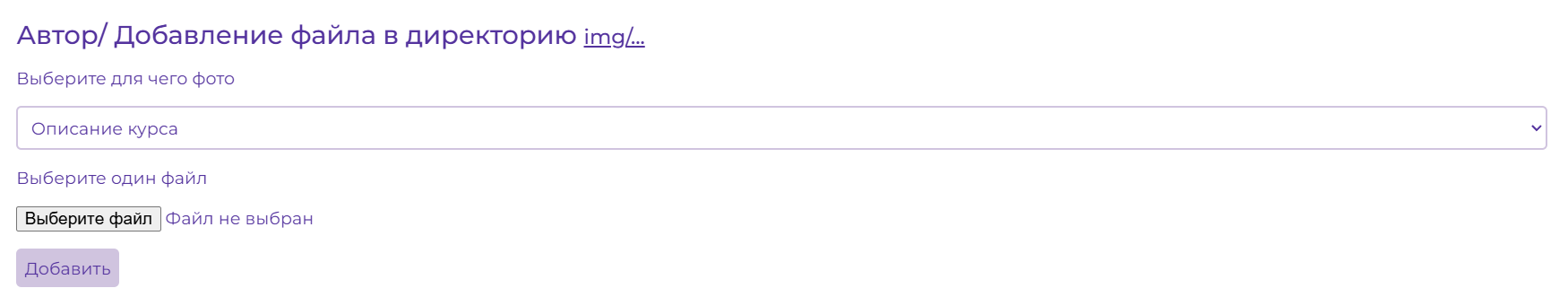


Рисунок 2.3.16 – Форма добавления изображения в БД и директорию

Автор может отследить статус отправленных заявок (рисунок 2.3.17).

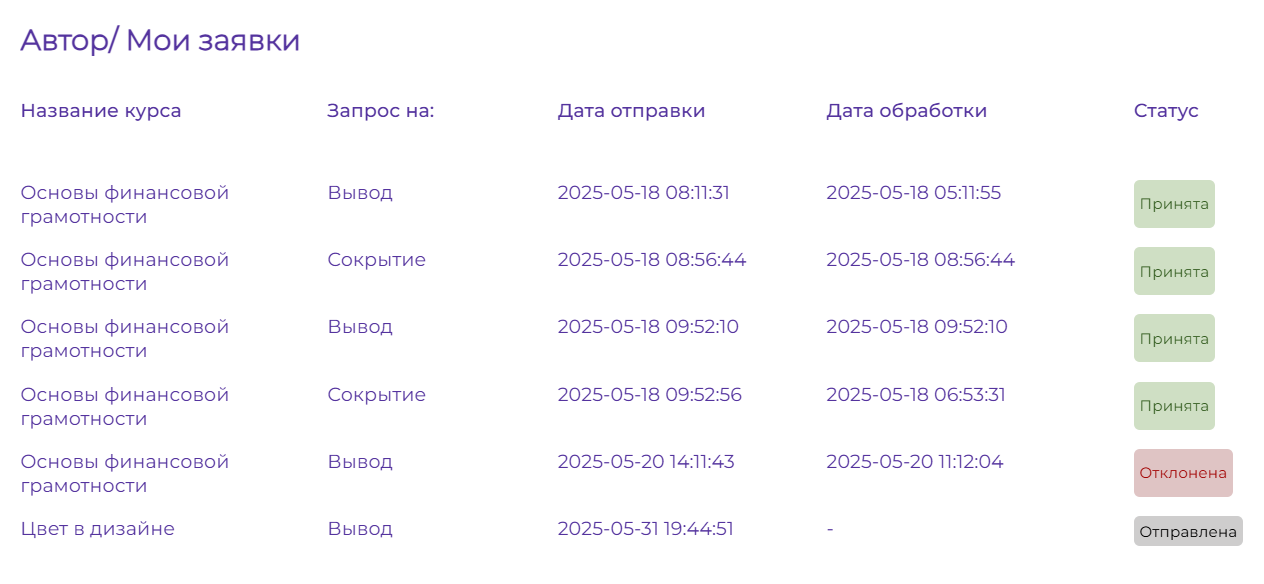


Рисунок 2.3.17 – Статусы заявок

Управлять содержанием курсов можно только в том случае, когда она не отображается в каталоге. Для изменения содержания курса необходимо со страницы курсов перейти в содержание необходимого курса по ссылке «содержание» (рисунок 2.3.18). Контент курса включает в себя тестирования и уроки. По кнопкам «Добавить урок» или «Добавить тест» автор переходит к страницам для создания урока или теста (рисунок 2.3.19 – 2.3.20).

Тестирование состоит из заданий четырех вариантов задач (рисунок 2.3.21). К тестированию также можно добавить ограничение времени, изменив переключатель и внеся необходимое ограничение в минутах.

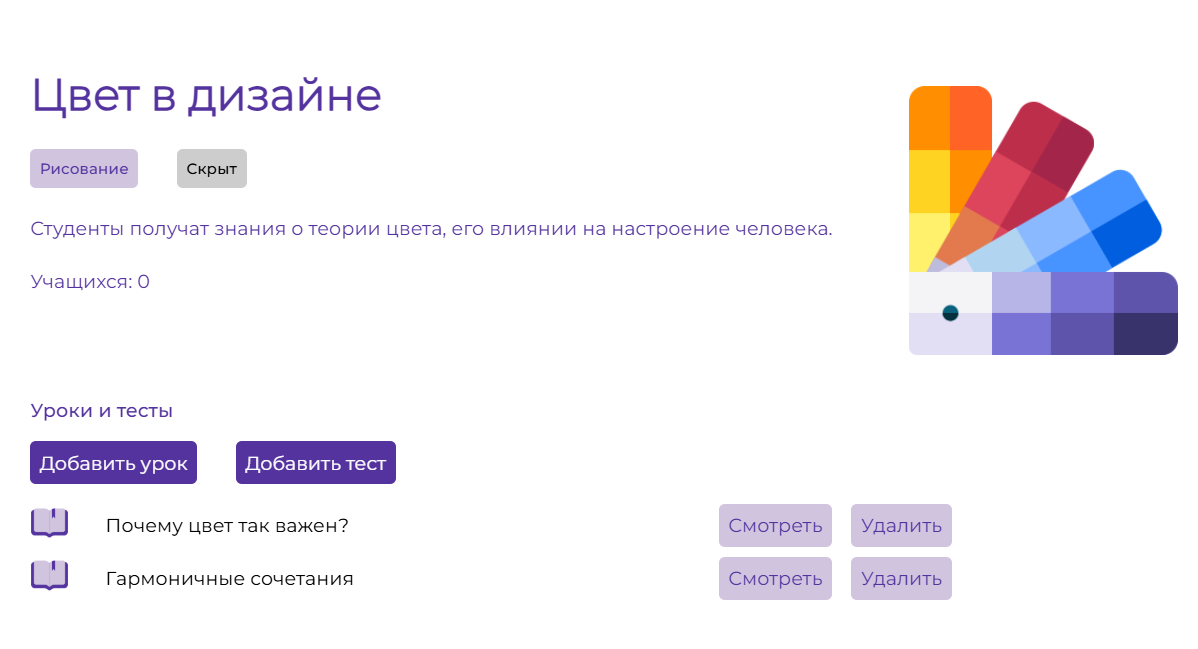


Рисунок 2.3.18 – Страница курса на странице автора

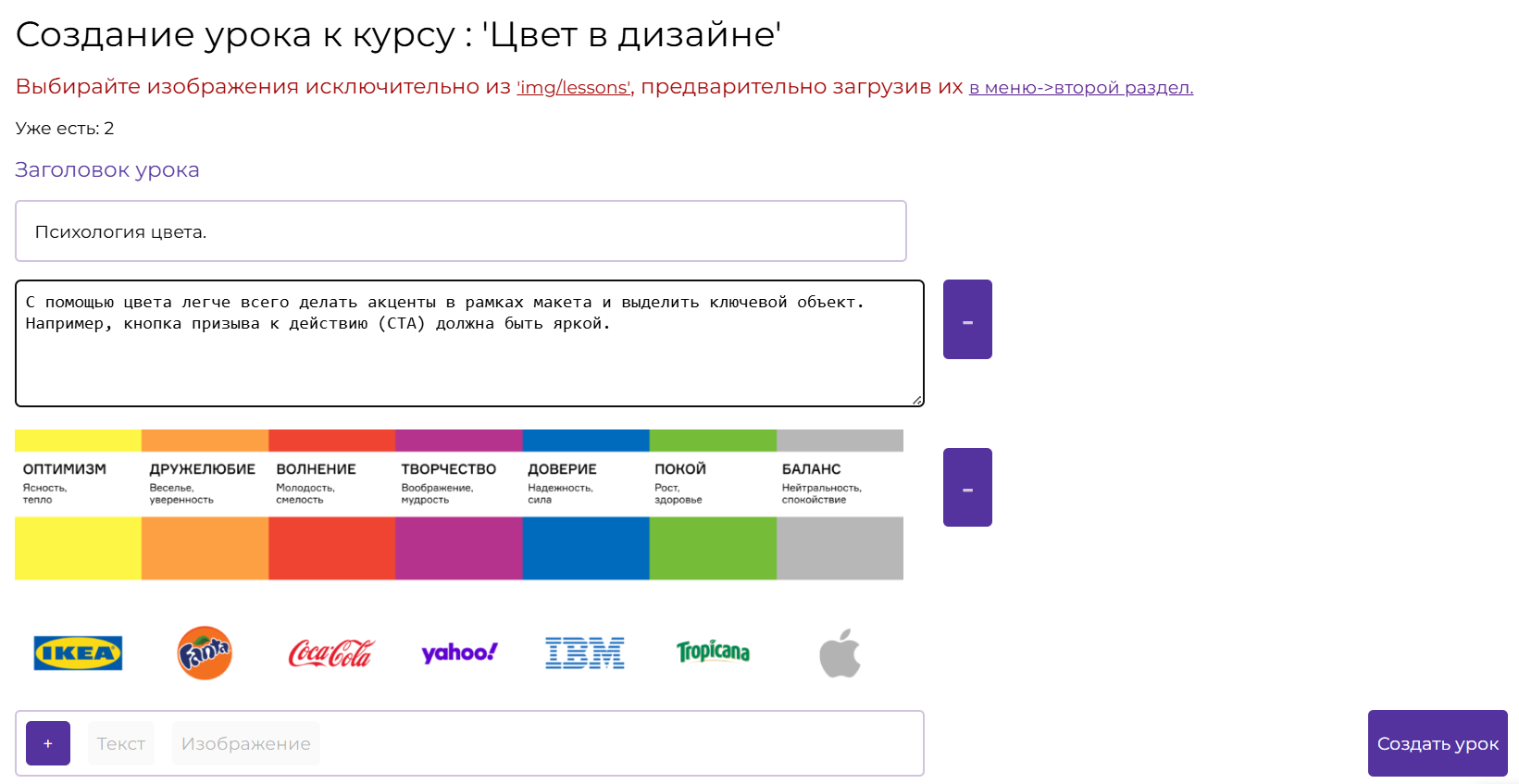


Рисунок 2.3.19 – Страница создания урока

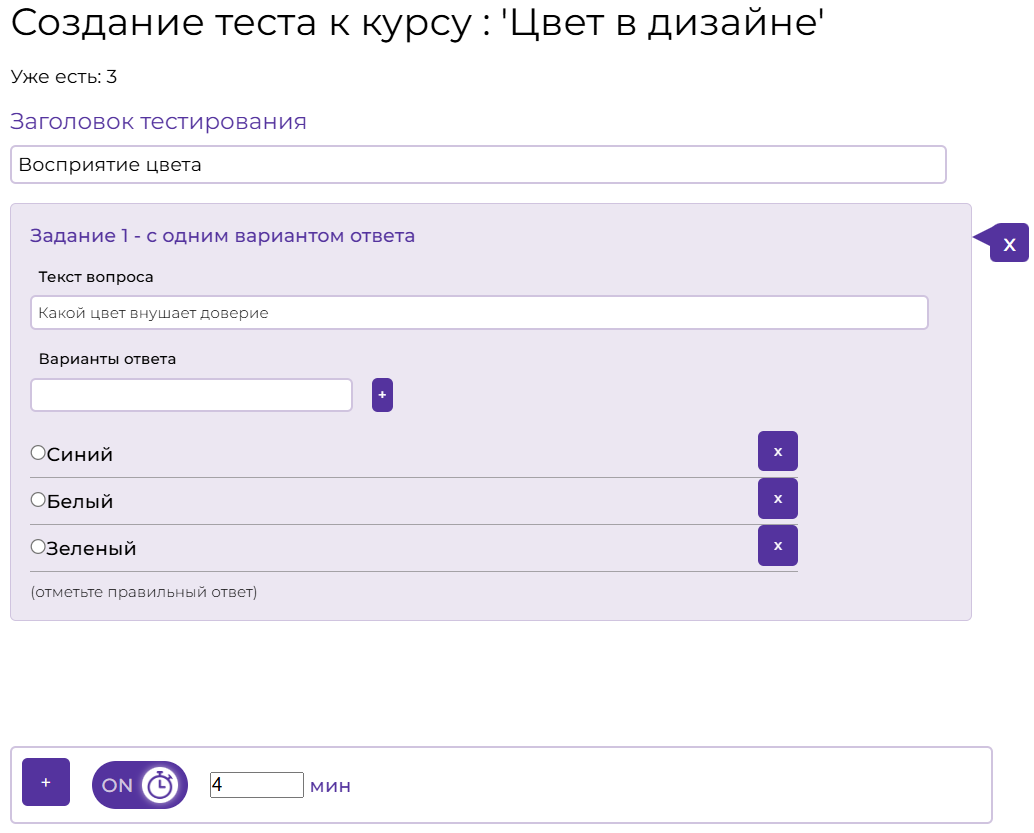


Рисунок 2.3.20 – Страница создания тестирования

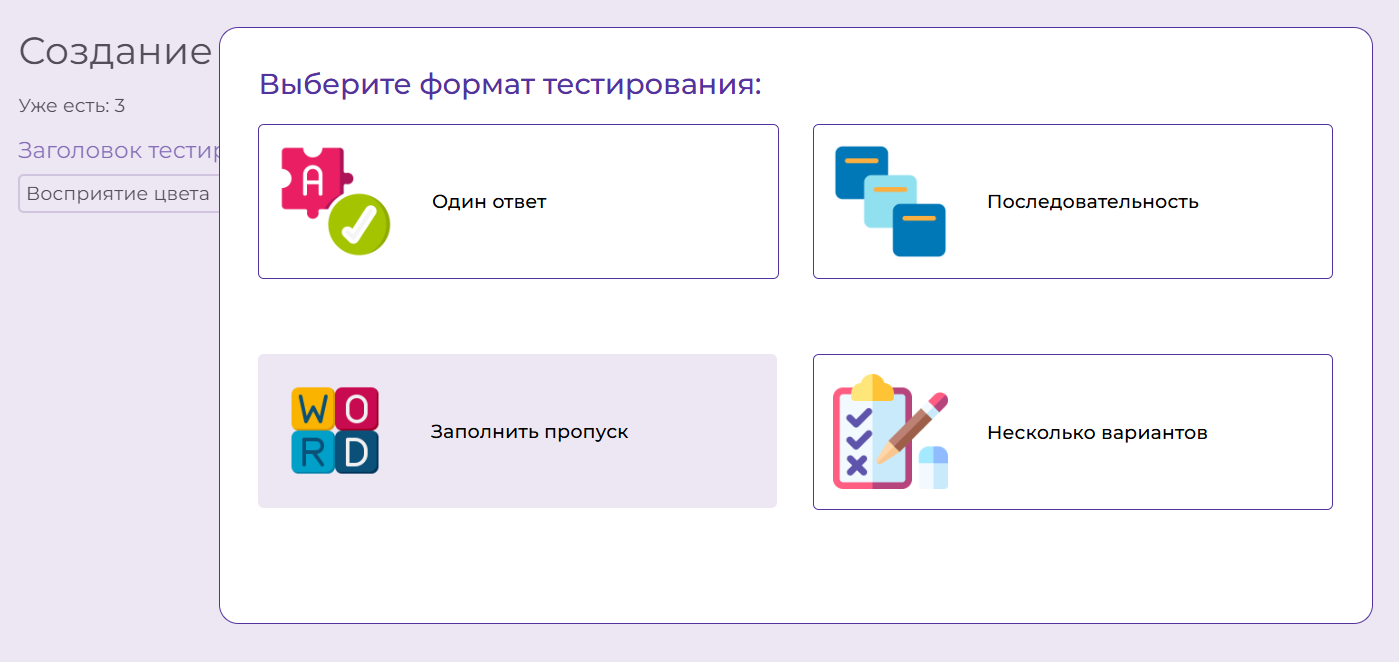


Рисунок 2.3.21 – Четыре варианта задач

ЗАКЛЮЧЕНИЕ

В процессе выполнения проекта были разработаны структура и алгоритм работы веб-сервиса для образовательной платформы, выполняющий функции поиска курсов, фильтрации, сортировки, управления курсами, уроками и тестированиями, просмотра результатов тестирования, управления и просмотра категорий курсов, обработки заявок на вывод курсов в каталог и заявок на смену роли пользователя, управления доступом пользователей, изменения данных профиля. Результатом работы стало создание программного продукта, написанного на языке программирования PHP и с использованием СУБД MySQL, фреймворка Laravel, библиотеки jQuery, html2pdf, Chart.js.

Была проведена проверка и отладочное тестирование информационной системы. По результатам отладочного тестирования были устранены некоторые недостатки.

С помощью системы на основании данных контрольного примера были получены результаты, которые полностью совпали с выходной информацией контрольного примера.

В процессе выполнения проекта были выполнены следующие задачи:

* проанализирована предметная область и выполнено описание предметной области;

Изм.

Лист

№ докум.

Подпись

Дата

Лист

PAGE \\* MERGEFORMAT3

*40. Н-1073-25 09.02.07 КП-ПЗ*

* спроектирована АИС;
* разработана БД;
* разработаны программные модули;
* проведено тестирование АИС;
* разработана техническая документация.

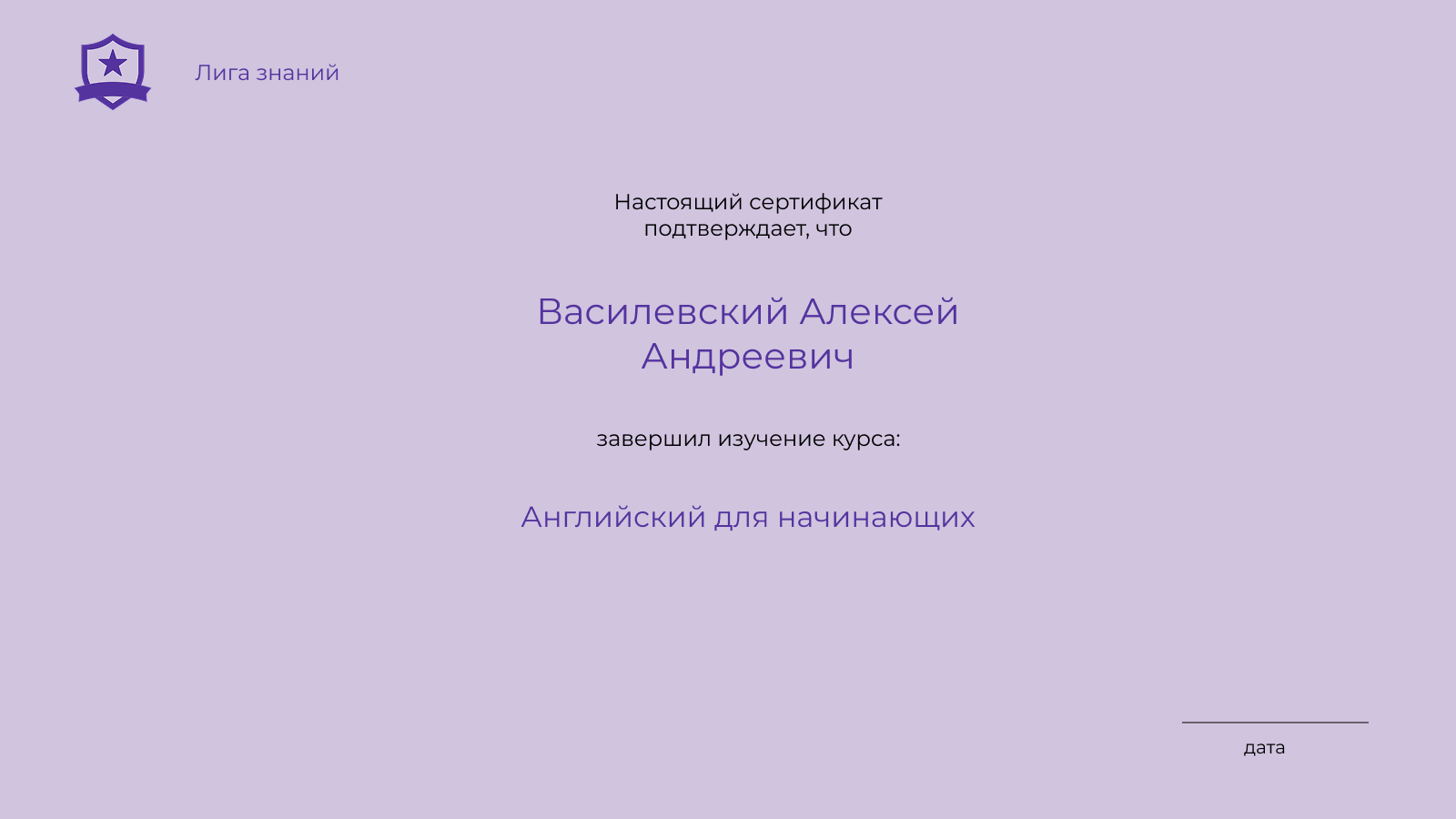
Поставленная цель данного проекта – упрощение процесса создания, прохождения и администрирования курсов путем разработки веб-приложения – была достигнута в ходе выполнения работы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ Р 7.0.108-2022. Система стандартов по информации, библиотечному и издательскому делу. Библиографическая ссылка. Общие требования и правила составления : национальный стандарт Российской Федерации : дата введения 2022-06-01 / Федеральное агентство по техническому регулированию. – Москва : Стандартинформ, 2022. – 24 с. – Текст : непосредственный.
2. ГОСТ Р 51583-2014. Защита информации. Порядок создания автоматизированных систем в защищенном исполнении. Общие положения : национальный стандарт Российской Федерации : дата введения 2014-09-01 / Федеральное агентство по техническому регулированию. – Москва : Стандартинформ, 2014. – 18 с. – Текст : непосредственный.
3. ГОСТ Р 59853-2021. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения : национальный стандарт Российской Федерации : дата введения 2022-01-01 / Федеральное агентство по техническому регулированию. – Москва : Стандартинформ, 2021. – 16 с. – Текст : непосредственный.
4. Галиаскаров, Э. Г. Анализ и проектирование систем с использованием UML : учебник для вузов / Э. Г. Галиаскаров, А. С. Воробьев. – Москва : Юрайт, 2025. – 125 с. – Текст : непосредственный.
5. Зараменских, Е. П. Проектирование информационных систем : учебник и практикум для среднего профессионального образования / Е. П. Зараменских. – 2-е изд. – Москва : Юрайт, 2025. – 119 с. – Текст : непосредственный.
6. Зараменских, Е. П. Разработка информационных систем : учебник и практикум для среднего профессионального образования / Е. П. Зараменских. – 2-е изд. – Москва : Юрайт, 2025. – 78 с. – Текст : непосредственный.
7. Зыков, С. В. Объектно-ориентированное программирование : учебник и практикум для вузов / С. В. Зыков. – 2-е изд. – Москва : Юрайт, 2025. – 151 с. – Текст : непосредственный.
8. Илюшечкин, В. М. Основы использования и проектирования баз данных : учебник для среднего профессионального образования / В. М. Илюшечкин. – Москва : Юрайт, 2025. – 213 с. – Текст : непосредственный.
9. Маркин, А. В. Программирование на SQL : учебник и практикум для вузов / А. В. Маркин. – 3-е изд., перераб. и доп. – Москва : Юрайт, 2025. – 805 с. – Текст : непосредственный.
10. Меженин, А. В. Проектирование, разработка и оптимизация веб-приложений : учебник для студ. учреждений сред. проф. образования / А. В. Меженин, Д. А. Меженин. – Москва : Академия, 2020. – 272 с – Текст : непосредственный.
11. Нестеров, С. А. Базы данных : учебник и практикум для среднего профессионального образования / С. А. Нестеров. – 2-е изд. – Москва : Юрайт, 2025. – 258 с. – Текст : непосредственный.
12. Полуэктова, Н. Р. Разработка веб-приложений : учебник для среднего профессионального образования / Н. Р. Полуэктова. – 2-е изд. – Москва : Юрайт, 2025. – 204 с. –Текст : непосредственный.
13. Проектирование информационных систем : учебник и практикум для вузов / Д. В. Чистов, П. П. Мельников, А. В. Золотарюк, Н. Б. Ничепорук. – 2-е изд., перераб. и доп. – Москва : Юрайт, 2025. – 273 с. – Текст : непосредственный.
14. Скляр, Д. Изучаем PHP 7: руководство по созданию интерактивных веб-приложений / Д. Скляр. – Москва : Диалектика, 2018. – 456 с. – Текст : непосредственный.
15. Сысолетин, Е. Г. Разработка интернет-приложений : учебное пособие для СПО / Е. Г. Сысолетин, С. Д. Ростунцев. – Москва : Юрайт, 2019. – 90 с. – Текст : непосредственный.
16. Тузовский, А. Ф. Объектно-ориентированное программирование : учебник для вузов / А. Ф. Тузовский. – Москва : Юрайт, 2025. – 213 с. – Текст : непосредственный.
17. Щербак, А. В. Тестирование программного обеспечения : учебник для вузов / А. В. Щербак. – Москва : Юрайт, 2025. – 145 с. – Текст : непосредственный.
18. jQuery API [официальная документация]. – URL: https://api.jquery.com/ (дата обращения: 22.04.2025). – Текст : электронный.
19. Laravel [официальная документация фреймворка]. – URL: https://laravel.com/ (дата обращения: 22.04.2025). – Текст : электронный.
20. PHP [официальная документация языка]. – URL: https://www.php.net/manual/ru/index.php (дата обращения: 22.04.2025). – Текст : электронный.

Приложение А

На рисунке А.1 изображен шаблон сертификата о прохождении тестирования.



(изменить)

Рисунок А.1 – Шаблон сертификата о прохождении тестирования

Приложение Б

Входные данные для создания курса представлены в таблице Б.1.

Таблица Б.1 – Входные данные для создания курса

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Заголовок | Категория | Описание | Дополнительные блоки | Изображение |
| 1 | 2 | 3 | 4 | 5 |
| Греческий язык | Языки мира | Разговаривайте на языке богов. Изучайте анатомию благодаря | {"1":{"text\_img":{"img": "greek.jpg", "desc": "оно открывает доступ к древнейшей цивилизации, а во-вторых, способствует когнитивному развитию и улучшению памяти, внимания и способности к обучению","title": "Почему стоит учить греческий язык"}} | greek\_lang.png |
| История Византии | История | Византи́я, также известная как Восточная Римская империя. | {"1": {"four\_blocks": {"1": {"img": "greek\_5.jpg","text": "Византии не было?"},"2": {"img": "greek\_6.jpeg","text": "Византийцы – римляне?" },"title": "Рассмотрим:"}}} | vizantiya.png |

Продолжение таблицы Б.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Скорочтение | Школьный курс | Шесть техник быстрого чтения, которые помогут эффективно запоминать прочитанное | {"1":{"text\_img":{"img": "greek.jpg", "desc": "Скорочтение полезно всем, кто часто работает с текстами и хочет экономить время, повысить эффективность и качество усвоения информации.","title": "Для кого необходимо:"}} | speedread.png |
| Основы программирования | Информатика | Изучим основы, которые проложат дорогу к знаниям | {"1": {"four\_blocks": {"1": {"img": "front.jpg","text": "Бэкенд"},"2": {"img": "back.jpeg","text": "Фронтенд" },"title": "Рассмотрим:"}}} | base\_code.png |
| Привлекающий дизайн | Дизайн | Рассмотрим примеры дизайнов и выделим основные тенденции. | {"1": {"four\_blocks": {"1": {"img": "3d.jpg","text": "3-D"},"2": {"img": "back.jpeg","text": "Ретро" },"title": "Тенденции 2025:"}}} | pretty\_des.png |

Таблица Б.2 – Входные данные контрольного примера «Уроки и тесты»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование курса | Тип | Заголовок теста | Содержимое | Таймер |
| Греческий язык для начинающих | test | Начальный тест | {"timer": "1","content": {"1": {"word": {"current": "альфа","question": "Напишите название буквы α}, "2":{"some\_answer": {"correct": {"2": "Средиземное", "5": "Ионическое"}, "question": "Какие из перечисленных морей омывают Грецию?", "incorrect": {"1": "Каспийское", "3": "B", "4": "Черное"}}} | 1 |
| Школьная программа по истории Византии | lesson | Первая глава | {{"txt":"Византия образовалась в 395 году в результате распада огромной Римской империи, переживавшей глубокий внутренний кризис, на две обособленные части. Случилось это в год смерти императора Феодосия Великого, разделившего между двумя сыновьями Западную и Восточную Римскую империю."},{ "img":"vizanty.png"}} | OFF |
| Скорочтение | lesson | Виды скорочтения | {{"txt": "Скорочтение — это набор техник, позволяющих увеличить скорость чтения в 3–5 раз (до 600–1000 слов/мин) без потери понимания текста."},{"img":"speed\_read.jpg"},{"txt":"Линейное — классическое чтение слева направо, но с высокой скоростью. Диагональное — взгляд скользит по тексту наискосок, выхватывая ключевые фразы. Вертикальное — чтение сверху вниз по центру страницы (для таблиц, списков)."}} | OFF |

Продолжение таблицы Б.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Основы основ программирования | lesson | Вводная лекция | {"1": {"txt": ""Этот курс создан для тех, кто делает первые шаги в мире IT. Здесь вы не только познакомитесь с базовыми концепциями программирования, но и научитесь мыслить, как разработчик.},"2":{"img":"lang.png"}} | OFF |
| Привлекающий дизайн | test | Как привлечь аудиторию | {"timer": "OFF","content": {"1": {"one\_answer": {"answers": ["Красный","Синий","Желтый",],"current": " Синий ", "question": "Какой цвет чаще всего ассоциируется с надежностью и доверием?"}},"2": {"word": {"current": "Отзыв","question": " Какой прием повышает доверие пользователей? "}}},"title\_test": " Как привлечь аудиторию "} | OFF |

Приложение В

На рисунке В.1 представлена физическая структура веб-приложения.

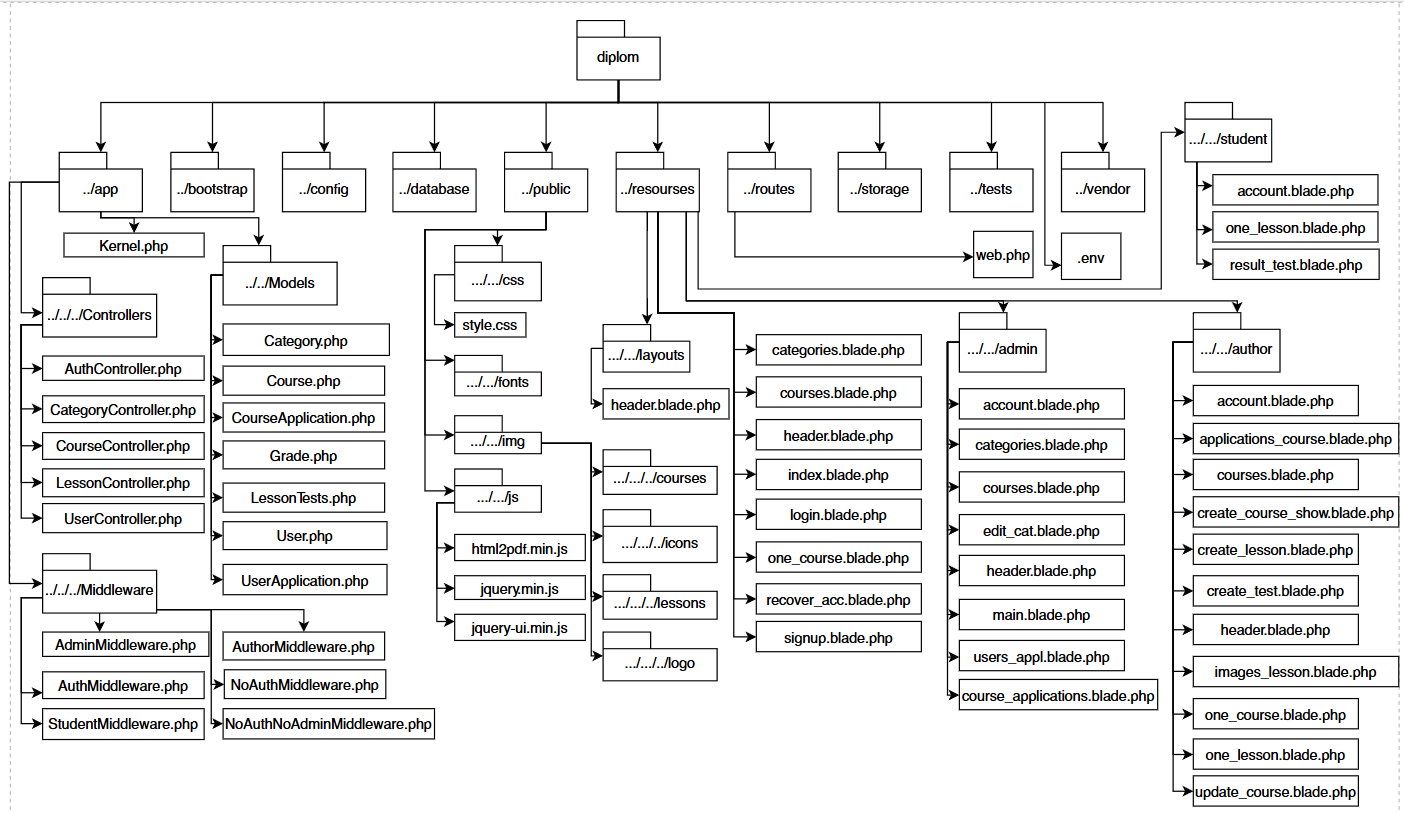


Рисунок В.1 – Физическая структура веб-приложения