

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский экономический университет им. Г.В. Плеханова»
Московский приборостроительный техникум

Курсовой проект

ПМ 02 Осуществление интеграции программных модулей

МДК 02.01 Технология разработки программного обеспечения

Специальность 09.02.07 «Информационные системы и
программирование»

Квалификация: Программист

Тема: «Разработка информационной системы «Повышение
квалификации сотрудников» (на примере ООО «Долфинс АЙ ТИ»)»

Пояснительная записка

Листов: 302

Руководитель

_____ / Л.А. Соколова
«____» _____ 2025 год

Исполнитель

_____ / К.М. Кириллин
«____» _____ 2025 год

2025

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Российский экономический университет имени Г.В. Плеханова»
МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ

«УТВЕРЖДАЮ»
Заместитель директора по учебной работе
_____ Д.А. Клопов
«____» 2025 г.

ЗАДАНИЕ

на выполнение курсового проекта (курсовой работы)

Кириллину Кириллу Михайловичу

(фамилия, имя, отчество студента — полностью)

студенту группы П50-1-21 специальности 09.02.07 «Информационные системы и программирование» по МДК 02.01 «Технология разработки программного обеспечения»

1. Исходные данные к проекту (работе):

1.1. Тема: «Разработка информационной системы «Повышение квалификации сотрудников» (на примере ООО «Долфинс АЙ ТИ»)

1.2. Состав курсового проекта:

1.2.1. Задание КП

1.2.2. Пояснительная записка

1.2.3. Программа (исходные данные) на электронном носителе

1.2.4. Презентация и инсталляционный пакет программы на электронном носителе

1.3. Содержание пояснительной записи:

ВВЕДЕНИЕ

1. ОБЩАЯ ЧАСТЬ

1.1. Цель разработки

1.2. Средства разработки

2. СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1. Постановка задачи

2.1.1. Входные данные

2.1.2. Выходные данные

2.1.3. Подробные требования к проекту

2.2. Внешняя спецификация

2.2.1. Описание задачи

2.2.2. Входные и выходные данные

2.2.3. Методы

2.2.4. Тесты

2.2.5. Контроль целостности данных

2.3. Проектирование

2.3.1. Схема архитектуры приложения

2.3.2. Логическая схема данных

2.3.3. Физическая схема данных

2.3.4. Структурная схема

2.3.5. Функциональная схема

2.3.6. Диаграмма классов

2.3.7. Схема тестирования

2.3.8. Схема пользовательского интерфейса

2.4. Результат работы программы

3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

3.1. Инstrumentальные средства

3.2. Отладка программы

3.3. Защитное программирование

3.4. Характеристики программы

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ

ПРИЛОЖЕНИЕ А. Техническое задание

ПРИЛОЖЕНИЕ Б. Текст программы

ПРИЛОЖЕНИЕ В. Программа испытаний

ПРИЛОЖЕНИЕ Г. Руководство пользователя

ПРИЛОЖЕНИЕ Д. Скрипт базы данных

2. Содержание задания по проекту (работе) — перечень вопросов, подлежащих разработке

	Разрабатываемый вопрос	Объем от всего задания, %	Срок выполнения
A	Описательная часть проекта (введение, общее описание и т. д.)	5	
1.	Введение	-	24.01.2025
2.	Цель разработки	-	24.01.2025
3.	Средства разработки	-	24.01.2025
B	Анализ задачи и её постановка	15	
1.	Определение требований к программе	-	31.01.2025
2.	Спецификация программы (описание задачи, описание входных и выходных данных, методы)	-	31.01.2025
3.	Тесты, контроль целостности данных	-	31.01.2025
B	Проектирование и реализация	55	
1.	Схемы проекта (схема архитектуры, логическая схема данных, физическая схема данных, функциональная и структурная схемы, диаграмма классов, схема тестирования, схема пользовательского интерфейса)	-	14.02.2025
2.	Реализация в инструментальной среде	-	28.02.2025
G	Технологическая часть проекта	5	
1.	Инструментальные средства разработки	-	14.03.2025
2.	Отладка программы	-	14.03.2025
3.	Защитное программирование	-	14.03.2025
4.	Характеристика программы	-	14.03.2025
D	Программная документация	10	
1.	ПРИЛОЖЕНИЕ А. Техническое задание	-	28.03.2025
2.	ПРИЛОЖЕНИЕ Б. Текст программы	-	28.03.2025
3.	ПРИЛОЖЕНИЕ В. Программа испытаний	-	28.03.2025
4.	ПРИЛОЖЕНИЕ Г. Руководство пользователя	-	28.03.2025
5.	ПРИЛОЖЕНИЕ Д. Скрипт базы данных	-	28.03.2025
E	Экспериментальная часть проекта	10	
1.	Программа на машинном носителе. Информация на носителе разбита на разделы: эксплуатационный пакет, тексты программы, документация.	-	14.04.2025

Руководитель курсового проекта (работы) Соколова Лариса Алексеевна, преподаватель

«21» января 2025 года _____ / Л.А. Соколова /

Дата выдачи курсового задания

Срок сдачи законченного проекта (работы)

«21» января 2025 года

«04» апреля 2025 года

Задание принял к исполнению

«21» января 2025 года / К.М. Кириллин /

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ федеральное государственное бюджетное образовательное
учреждение высшего образования
«Российский экономический университет им. Г.В. Плеханова»
Московский приборостроительный техникум

Курсовой проект

ПМ 11 Разработка, администрирование и защита баз данных

МДК 11.01 «Технология разработки и защиты баз данных»

Специальность 09.02.07 «Информационные системы и
программирование»

Квалификация: Программист

Тема: «Разработка информационной системы «Повышение
квалификации сотрудников (на примере ООО «Долфинс АЙ ТИ»)».

Пояснительная записка

Листов:

Руководитель

_____ /Л.А.Соколова

«____» _____ 2025 год

Исполнитель

_____ /К.М. Кириллин

«____» _____ 2025 год

2025

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Российский экономический университет имени Г.В. Плеханова»

МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ

«УТВЕРЖДАЮ»

Заместитель директора по учебной работе

_____ Л.А.Соколова

«____» _____ 2025 г.

ЗАДАНИЕ

на выполнение курсового проекта (курсовой работы)

Кириллину Кириллу Михайловичу

(фамилия, имя, отчество сотрудника — полностью)

сотруднику группы П50-1-21 специальности 09.02.07 «Информационные системы и
программирование» по МДК 02.01 «Технология разработки программного обеспечения»

1. Исходные данные к проекту (работе):

1.1. Тема: «Разработка информационной системы «Повышение квалификации сотрудников»
(на примере ООО «Долфинс АЙ ТИ»)

1.2. Состав курсового проекта:

1.2.1. Задание КП

1.2.2. Пояснительная записка

1.2.3. Программа (исходные данные) на электронном носителе

1.2.4. Презентация и инсталляционный пакет программы на электронном носителе

1.3. Содержание пояснительной записи:

ВВЕДЕНИЕ

1. ОБЩАЯ ЧАСТЬ

1.1. Цель разработки

1.2. Средства разработки

2. СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1. Постановка задачи

2.1.1. Входные данные

2.1.2. Выходные данные

2.1.3. Подробные требования к проекту

2.2. Внешняя спецификация

2.2.1. Описание задачи

2.2.2. Входные и выходные данные

2.2.3. Методы

2.2.4. Тесты

2.2.5. Контроль целостности данных

2.3. Проектирование

2.3.1. Схема архитектуры приложения

2.3.2. Логическая схема данных

2.3.3. Физическая схема данных

2.3.4. Структурная схема

2.3.5. Функциональная схема

2.3.6. Диаграмма классов

2.3.7. Схема тестирования

2.3.8. Схема пользовательского интерфейса

2.4. Результат работы программы

3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

3.1. Инструментальные средства

3.2. Отладка программы

3.3. Защитное программирование

3.4. Характеристики программы

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗУЕМЫХ МАТЕРИАЛОВ

ПРИЛОЖЕНИЕ А. Техническое задание

ПРИЛОЖЕНИЕ Б. Текст программы

ПРИЛОЖЕНИЕ В. Программа испытаний

ПРИЛОЖЕНИЕ Г. Руководство пользователя

ПРИЛОЖЕНИЕ Д. Скрипт базы данных

2. Содержание задания по проекту (работе) — перечень вопросов, подлежащих разработке

	Разрабатываемый вопрос	Объем от всего задания, %	Срок выполнения
A	Описательная часть проекта (введение, общее описание и т. д.)	5	
1.	Введение	-	24.01.2025
2.	Цель разработки	-	24.01.2025
3.	Средства разработки	-	24.01.2025
B	Анализ задачи и её постановка	15	
1.	Определение требований к программе	-	31.01.2025
2.	Спецификация программы (описание задачи, описание входных и выходных данных, методы)	-	31.01.2025
3.	Тесты, контроль целостности данных	-	31.01.2025
B	Проектирование и реализация	55	
1.	Схемы проекта (схема архитектуры, логическая схема данных, физическая схема данных, функциональная и структурная схемы, диаграмма классов, схема тестирования, схема пользовательского интерфейса)	-	14.02.2025
2.	Реализация в инструментальной среде	-	28.02.2025

	Разрабатываемый вопрос	Объем от всего задания, %	Срок выполнения
Г	Технологическая часть проекта	5	
1.	Инструментальные средства разработки	-	14.03.2025
2.	Отладка программы	-	14.03.2025
3.	Защитное программирование	-	14.03.2025
4.	Характеристика программы	-	14.03.2025
Д	Программная документация	10	
1.	ПРИЛОЖЕНИЕ А. Техническое задание	-	28.03.2025
2.	ПРИЛОЖЕНИЕ Б. Текст программы	-	28.03.2025
3.	ПРИЛОЖЕНИЕ В. Программа испытаний	-	28.03.2025
4.	ПРИЛОЖЕНИЕ Г. Руководство пользователя	-	28.03.2025
5.	ПРИЛОЖЕНИЕ Д. Скрипт базы данных	-	28.03.2025
Е	Экспериментальная часть проекта	10	
1.	Программа на машинном носителе. Информация на носителе разбита на разделы: эксплуатационный пакет, тексты программы, документация.	-	14.04.2025

Руководитель курсового проекта (работы) Соколова Лариса Алексеевна, преподаватель

«21» января 2025 года _____ / Л.А. Соколова /

Дата выдачи курсового задания

«21» января 2025 года

Срок сдачи законченного проекта (работы) «04» апреля 2025 года

Задание принял к исполнению

«21» января 2025 года _____ / К.М. Кириллин /

Содержание

ВВЕДЕНИЕ.....	6
1. ОБЩАЯ ЧАСТЬ.....	7
1.1. Цель разработки.....	7
1.2. Средства разработки.....	7
2. СПЕЦИАЛЬНАЯ ЧАСТЬ.....	9
2.1. Постановка задачи.....	9
2.2. Внешняя спецификации.....	11
2.3. Проектирование.....	25
3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ.....	49
3.1. Инструментальные средства.....	49
3.2. Отладка программы.....	50
3.3. Защитное программирование.....	50
3.4. Характеристики программы.....	51
ЗАКЛЮЧЕНИЕ.....	52

ВВЕДЕНИЕ

С развитием информационных технологий и потребностью в постоянном обновлении знаний, системы управления повышением квалификации сотрудников стали ключевым инструментом для современных ИТ-компаний. Регулярное обучение и развитие персонала является необходимым условием для поддержания конкурентоспособности компании на рынке информационных технологий. Однако для обеспечения эффективного процесса повышения квалификации важна качественная информационная система «BustCom», способная управлять учебными курсами, тестированием и отслеживанием прогресса сотрудников.

Проект информационной системы «BustCom» направлен на разработку современного решения для компании ООО "Долфинс АЙ ТИ". Основная цель курсовой работы — создание веб-приложения, которое объединяет административную панель для HR-отдела и руководства компании и удобный интерфейс для сотрудников. Приложение «BustCom» обеспечит эффективное управление учебными материалами и оптимизирует процесс профессионального развития персонала.

Существующие решения часто сталкиваются с проблемами: сложность в использовании, недостаток функционала для специфических потребностей ИТ-компаний или низкая интеграция с рабочими процессами. Разрабатываемая система «BustCom» создана с акцентом на гибкость, интуитивный интерфейс и безопасность данных. В приложении реализованы функции управления курсами, тестами, учёта сотрудников, а также аналитика эффективности обучения и формирование отчетов.

Курсовая работа включает в себя полный цикл разработки: от проектирования базы данных и программной логики до интеграции административной панели и пользовательского интерфейса. Приложение

«BustCom» станет универсальным инструментом для ООО "Долфинс АЙ ТИ", помогая решать задачи по развитию персонала и повышению общего уровня квалификации сотрудников компании.

1. ОБЩАЯ ЧАСТЬ

1.1. Цель разработки

Целью разработки является предоставление эффективной и современной системы управления образовательным процессом, которая позволит автоматизировать процесс создания и управления учебными курсами

1.2. Средства разработки

Для проектирования, разработки и тестирования веб-приложения, были использованы программные средства, указанные в Таблице 1.

Таблица 1 - Программные средства разработки ПО

№	Тип средства	Название средства	Назначение
1	2	3	4
1	Браузер	Google Chrome ver. 111.0.5563.65	Организация взаимодействия пользователя и программы.
2	Инструментальные средства разработки программного решения	Visual Studio Code ver. 1.95.3	Разработка программного обеспечения
3	СУБД	SQLite	Разработка базы данных

В качестве средств вычислительной техники при разработке ПО использовался ноутбук Lenovo Victum 16. Характеристики представлены в таблице 2.

Таблица 2 - Технические характеристики

№	Тип средства	Название средства
1	2	3
Ноутбук Lenovo Victum 16		
1	Размер экрана	14,0
2	Разрешение экрана	2560x1440
3	Линейка процессора	Ryzen 7 7700H
4	Количество ядер процессора	8
5	Оперативная память	16 ГБ
6	Тип видеокарты	встроенная
7	Видеокарта	Geforce RTX 3060
8	Конфигурация накопителей	SSD
9	Общий объем всех накопителей	1 ТБ
10	Операционная система	Windows 11 Pro

2. СПЕЦИАЛЬНАЯ ЧАСТЬ

2.1. Постановка задачи

Разработать веб-приложение BustCom, которое включает в себя следующие компоненты: просмотр, добавление и редактирование курсов, управление тестами, просмотр статистических графиков успеваемости и управление пользователями и их ролями.

2.1.1. Входные данные предметной области

Входные данные веб-приложения представлены в следующем виде:

- Аутентификационные данные пользователя (логин, пароль)
- Регистрационные данные пользователя (имя пользователя, электронная почта, пароль1, пароль2)
- Данные курса (название, описание, материалы, сроки)
- Данные теста (вопросы, варианты ответов, время выполнения)
- Данные ответов на тест (сотрудник, тест, ответы, время)
- Форма вопроса преподавателю (сотрудник, предмет, вопрос)
- Форма ответа преподавателя (сотрудник, предмет, вопрос, ответ)
- Данные пользователя (имя пользователя, электронная почта, пароль1, курсы, роль)
- Данные ролей (название)
- Данные группы (название, сотрудник)
- Управление курсами (курс, статус)

- Данные материалов курса (название, тип, содержимое)

2.1.2. Выходные данные предметной области

Выходные данные предметной области представлены в списке ниже:

- График успеваемости (сотрудник, средний балл)
- Данные курса (название, описание, материалы, сроки)
- Данные теста (вопросы, варианты ответов, время выполнения)
- Данные ответов на тест (сотрудник, тест, ответы, время)
- Данные пользователя (имя пользователя, электронная почта, пароль1, курсы, роль)
- Данные ролей (название)
- Данные группы (название, сотрудник)
- Данные материалов курса (название, тип, содержимое)
- Данные формы вопроса (сотрудник, предмет, вопрос)
- Данные формы ответа (сотрудник, предмет, вопрос, ответ)
- Статистика по курсам (количество сотрудников, средний балл)
- Статистика по тестам (процент успешных попыток, среднее время)

2.1.3. Подробные технические требования

Подробные требования к разработке веб-приложения представлены в Приложении А. Техническое задание.

2.2. Внешняя спецификации

2.2.1. Описание задачи

Разработать веб-приложение BustCom для управления курсами, тестами и пользователями.

Данное программное решение должно состоять из СУБД SQLite и веб-приложения.

Веб-приложение должно предоставлять функционал для отображения информации из базы данных с использованием ORM Django, а также предоставлять возможности добавления, изменения и удаления данных.

При запуске веб-приложения открывается главная страница с кратким описанием функционала приложения и кнопками для перехода на страницы авторизации и регистрации.

После авторизации или регистрации пользователь будет перенаправлен на главную страницу административной панели, где ему доступны следующие разделы:

- Управление пользователями — редактирование данных пользователей, управление их ролями
- Управление курсами — добавление, изменение и удаление курсов с их материалами
- Управление тестами — создание и редактирование тестов, просмотр результатов
- Просмотр аналитики — отображение статистических графиков по успеваемости и активности пользователей

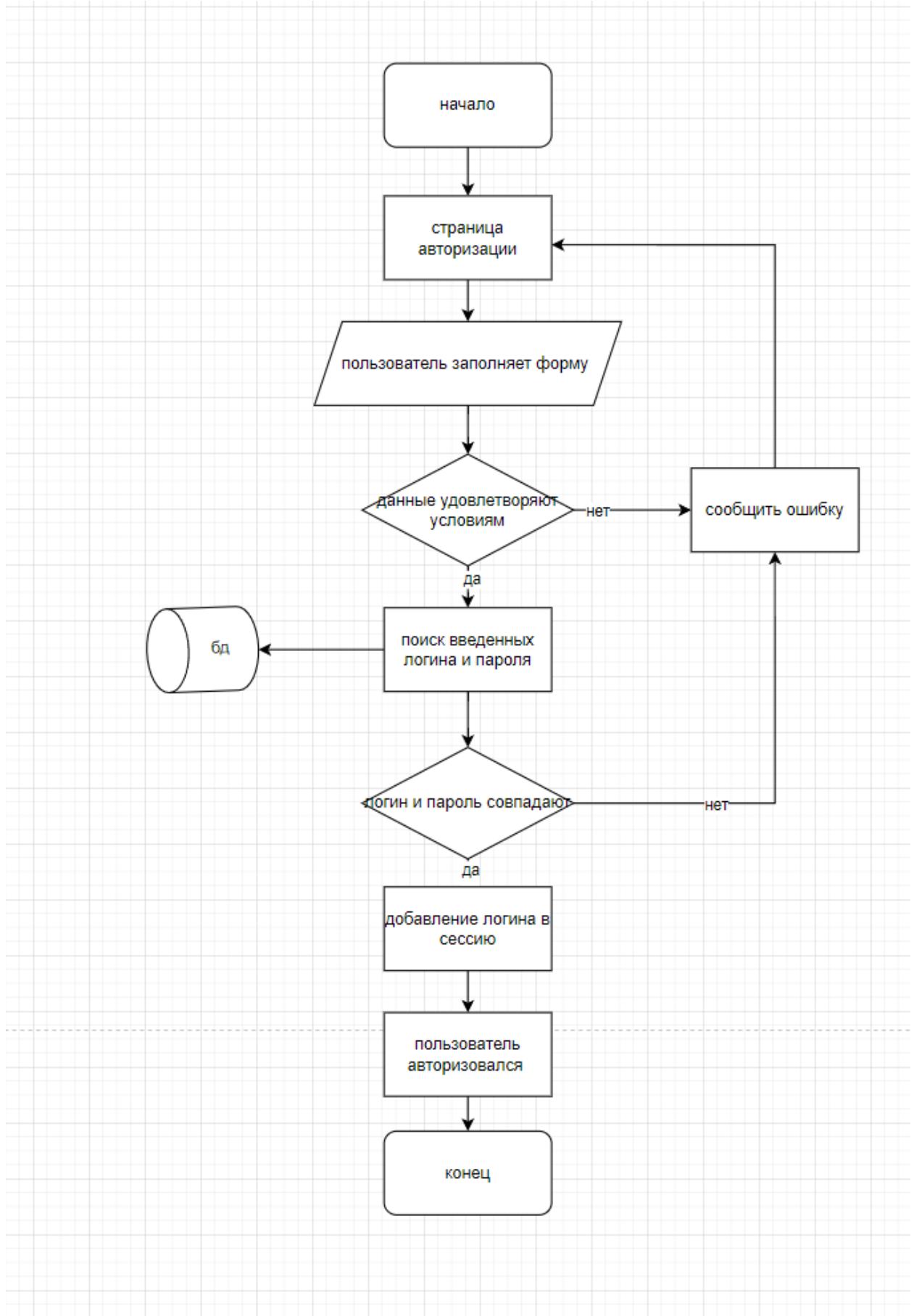


Рисунок 1 - авторизация

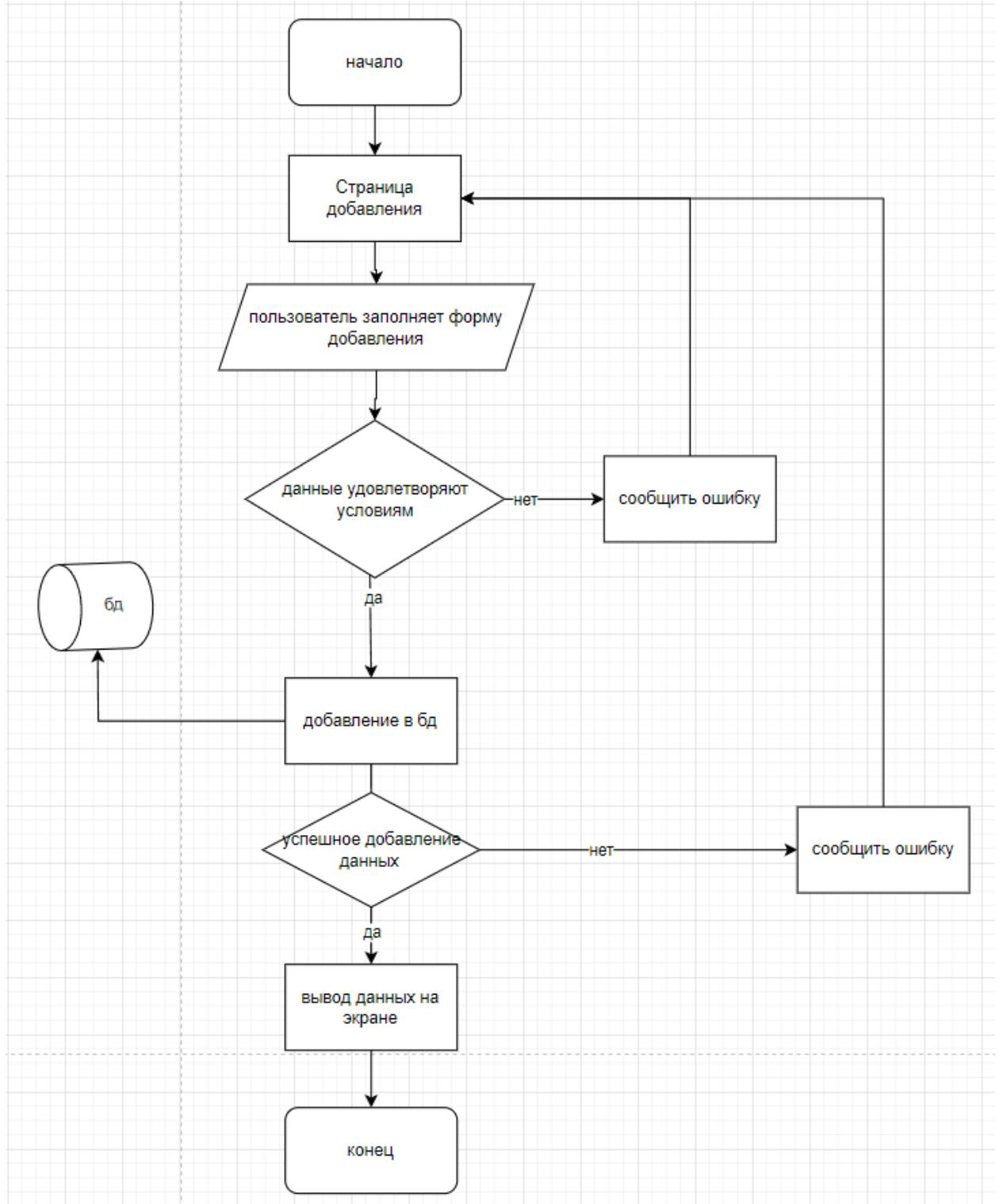


Рисунок 2 - добавление

Для более глубокого понимания образовательного процесса была разработана модель в формате IDEF0: to be. Модель представляет собой улучшенную версию системы, направленную на повышение эффективности образовательного процесса.

Модель to be представлена на рисунках 3-4.

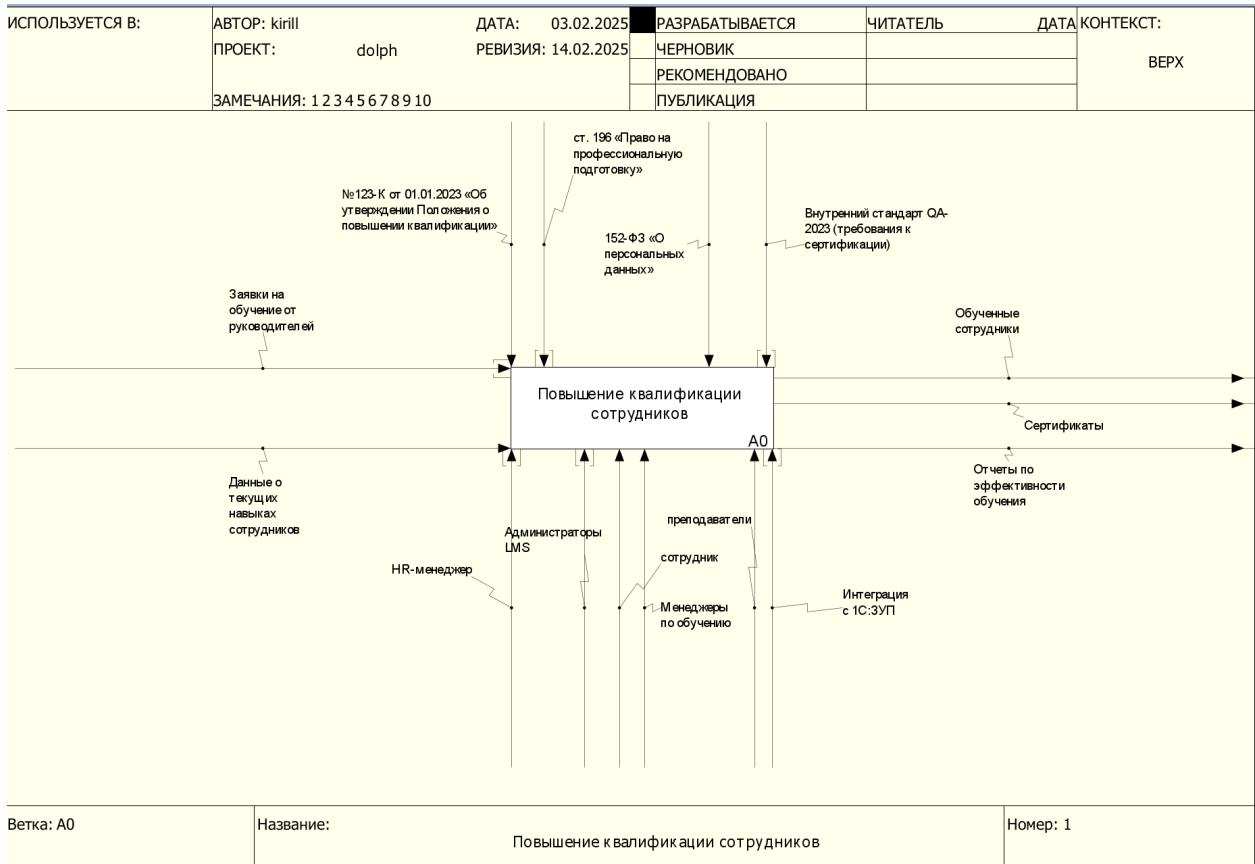


Рисунок 3 - Модель as is, часть первая

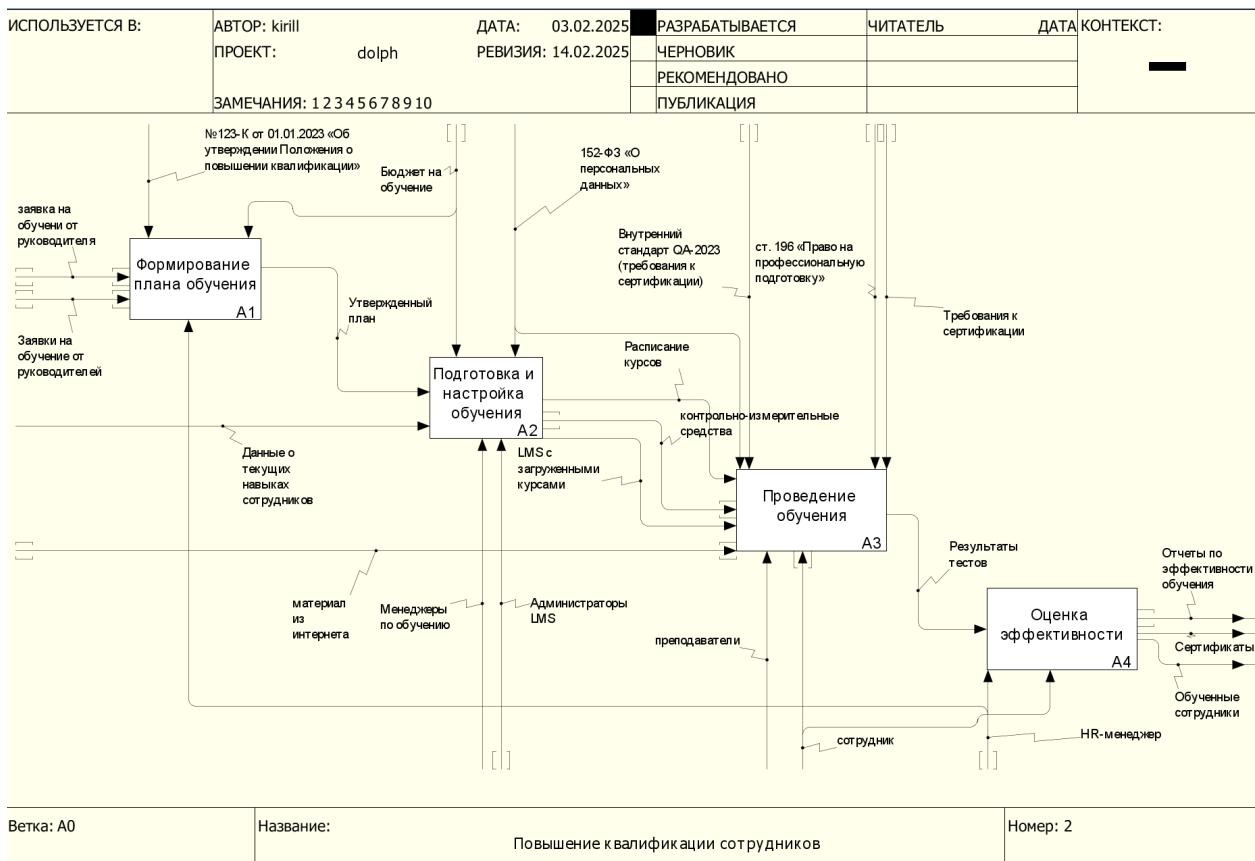


Рисунок 4 - Модель as is, декомпозиция

Также, для лучшего понимания работы процесса, была создана диаграмма прецедентов. Данную диаграмму можно увидеть на рисунке 5.

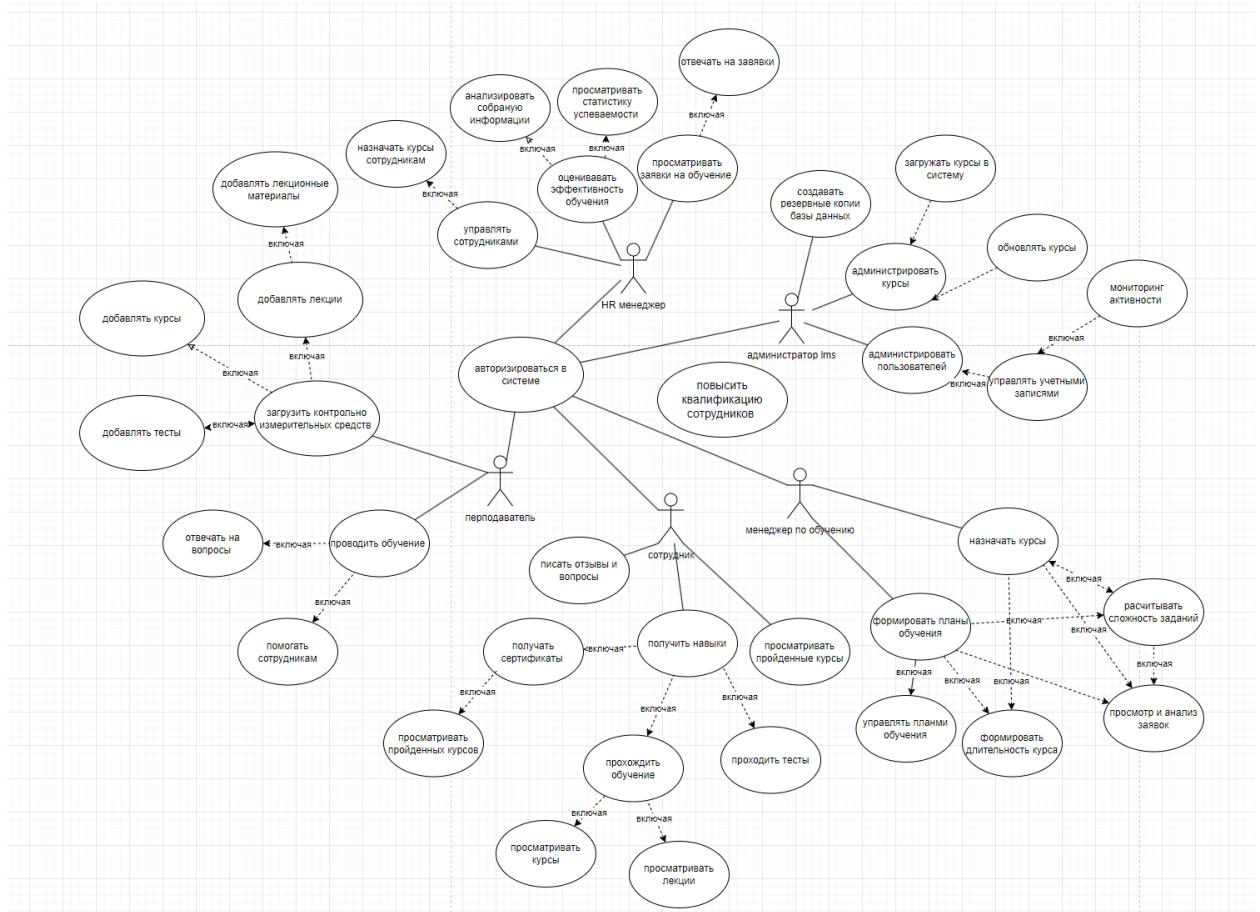


Рисунок 5 - Диаграмма прецедентов

В приложении присутствуют несколько ролей: сотрудник, преподаватель, администратор сайта, администратор базы данных и служба поддержки. Каждая из них имеет уникальные функции и взаимодействия, которые описаны ниже.

Возможности сотрудника:

Сотрудник взаимодействует с приложением для обучения и прохождения тестов.

Основные возможности:

- Регистрация: создание нового аккаунта с указанием уникального имени пользователя и пароля• Авторизация: вход в личный кабинет с использованием зарегистрированных учетных данных
- Просмотр курсов: доступ к списку доступных курсов, сортировка и фильтрация по категориям
- Прохождение тестов: выполнение тестовых заданий в отведенное время
- Изменять свой профиль
- Просмотр результатов: анализ оценок и прогресса обучения
- Задавание вопросов: создание сообщений с вопросами к преподавателям
- Настройка профиля: выбор темы оформления и настройка уведомлений

Возможности преподавателя:

Преподаватель управляет курсами и взаимодействует со сотрудниками.

Основные возможности:

- Создание курсов: разработка и публикация учебных материалов
- Управление тестами: создание и редактирование тестовых заданий
- Проверка работ: оценка выполненных заданий и тестов
- Ответы на вопросы: взаимодействие со сотрудниками через систему сообщений
- Просмотр статистики: анализ успеваемости сотрудников

- Формирование отчетов: создание отчетов по успеваемости
- Настройка курсов: управление материалами и сроками

Возможности администратора сайта:

Администратор управляет функциональностью приложения и его пользователями.

Основные возможности:

- Регистрация новых пользователей: создание учетных записей вручную
- Управление курсами: создание и редактирование курсов
- Администрирование сайта: управление контентом и настройками
- Просмотр данных пользователей: доступ к профилям и результатам
- Управление правами: настройка ролей и доступа
- Просмотр статистики: анализ активности пользователей
- Блокировка пользователей: управление доступом к системе
- Создание резервных копий: защита данных
- Создание резервных копий: регулярное сохранение данных
- Восстановление базы данных: загрузка сохраненных данных
- Настройка базы данных: контроль производительности
- Анализ данных: выполнение запросов для получения информации

Возможности менеджера по обучению:

Менеджер по обучению занимается организацией и контролем процесса повышения квалификации сотрудников.

Основные возможности:

- Создание и управление планами обучения
- Создание новых планов с указанием названия, описания, дат и статуса
- Редактирование существующих планов обучения
- Удаление планов, которые больше не актуальны

Возможности HR-менеджера:

HR-менеджер использует систему для управления персоналом и анализа потребностей в обучении.

Основные возможности:

- Отслеживание производительности обучения
- Просмотр средних, максимальных и минимальных баллов по тестам
- Анализ распределения результатов тестирования
- Доступ к статистике успеваемости сотрудников

2.2.2. Входные и выходные данные.

В таблицах 3 и 4 представлены входные и выходные данные приложения.

Таблица 3 - Входные данные

Имя	Тип	Ограничение	Формат ввода	Описание
Форма аутентификации				

Login	Строка	^[A-Za-z0-9!@#\$%^&()%]{5:20}\$	Поле ввода	Логин пользователя
Password	Строка	^[A-Za-z0-9!@#\$%^&()%]{5:20}\$	Поле ввода	Пароль пользователя
Форма регистрации				
Login	Строка	^[A-Za-z0-9!@#\$%^&()%]{5:20}\$	Поле ввода	Логин пользователя
Email	Строка	^[\w-\.]+@[(\w-]+\.)+[\w-]{2,4}\$	Поле ввода	Email пользователя
Password1	Строка	^[A-Za-z0-9!@#\$%^&()%]{5:20}\$	Поле ввода	Пароль пользователя
Password2	Строка	^[A-Za-z0-9!@#\$%^&()%]{5:20}\$	Поле ввода	Подтверждение пароля
Форма добавления курса				
Name	Строка	^[A-Za-z0-9!@#\$%^&()%]{5:50}\$	Поле ввода	Название курса
Description	Строка	Не более 1000 символов	Текстовое поле	Описание курса
Duration	Число	Не пустое	Поле ввода	Длительность курса
Category	Строка	^(?=.{3,20}\$)[а-яёА-ЯЁ]+(?:[-][а-яёА-ЯЁ]+)\$	Выпадающий список	Категория курса
Форма добавления теста				
Title	Строка	^[A-Za-z0-9!@#\$%^&()%]{5:50}\$	Поле ввода	Название теста
TimeLimit	Число	Не пустое	Поле ввода	Время на выполнение
Questions	Массив	Не пустой	Динамическая форма	Вопросы теста
Форма вопроса преподавателю				
Subject	Строка	^(?=.{3,50}\$)[а-яёА-ЯЁ]+(?:[-][а-яёА-ЯЁ])	Поле ввода	Тема вопроса

		+)\$		
Question	Строка	Не более 500 символов	Текстовое поле	Текст вопроса

Таблица 4 - Выходные данные

Имя	Тип	Ограничение	Описание
Графики			
PerformanceGraphic	График	Нет ограничений	График успеваемости по курсам
Данные курсов			
CourseID	Целочисленное число	Больше 1	ID курса
Name	Строка	Не пустое	Название курса
Description	Строка	Не пустое	Описание курса
Duration	Число	Не пустое	Длительность курса
Category	Строка	Не пустое	Категория курса
Данные тестов			
TestID	Целочисленное число	Больше 1	ID теста
Title	Строка	Не пустое	Название теста
TimeLimit	Число	Не пустое	Время на выполнение
Score	Число	Не пустое	Оценка за тест
Данные пользователей			
UserID	Целочисленное число	Больше 1	ID пользователя
Login	Строка	Не пустое	Логин пользователя
Email	Строка	Не пустое	Email пользователя
Role	Строка	Не пустое	Роль пользователя
Данные вопросов			
QuestionID	Целочисленное число	Больше 1	ID вопроса
UserID	Целочисленное число	Больше 1	ID пользователя

Subject	Строка	Не пустое	Тема вопроса
Question	Строка	Не пустое	Текст вопроса
Answer	Строка	Не пустое	Ответ преподавателя

2.2.3. Методы

Для работы с базой данных была использована ORM Django. С помощью данной ORM можно легко взаимодействовать с базой данных СУБД SQLite.

Необходимо использовать инкапсуляцию, которая позволяет защитить данные от нежелательного доступа и изменений, а также облегчить поддержку кода и его повторное использование с помощью скрытия определенных данных от других данных.

Также, при разработке нужно использовать рефакторинг и оптимизацию. Рефакторинг — это процесс улучшения структуры, качества и читаемости кода без изменения его функциональности. При рефакторинге программу не переписывают, не добавляют новых функций и не исправляют ошибки.

Оптимизация, с другой стороны, направлена на улучшение производительности приложения, т.е. на его скорость работы и эффективность использования ресурсов устройства. Оптимизация может включать изменение кода приложения, чтобы он работал быстрее, или использование оптимизированных библиотек и инструментов. Примеры оптимизации включают улучшение алгоритмов, оптимизацию использования памяти и ускорение процесса загрузки приложения.

```

def get_course_data(request):
    courses = Course.objects.all()
    for course in courses:
        students = Student.objects.filter(course=course)
        course.student_count = students.count()

        tests = Test.objects.filter(course=course)
        course.test_count = tests.count()

        for test in tests:
            results = TestResult.objects.filter(test=test)
            test.average_score = results.aggregate(Avg('score'))['score__avg']

    return render(request, 'courses.html', {'courses': courses})

```

Рисунок 6 - до оптимизации

```

def get_course_data(request):
    courses = Course.objects.annotate(
        student_count=Count('student'),
        test_count=Count('test')
    ).prefetch_related(
        'test_set',
        'test_set__testresult_set'
    )

    for course in courses:
        for test in course.test_set.all():
            test.average_score = test.testresult_set.aggregate(Avg('score'))['score__avg']

    return render(request, 'courses.html', {'courses': courses})

```

Рисунок 7 - после оптимизации

Также была проведена оптимизация. После оптимизации были устраниены:

- Повторные запросы к базе данных при загрузке курсов
- Избыточные вычисления при проверке тестов
- Неоптимальное кэширование данных пользователей
- Избыточные запросы при формировании отчетов

```
<!DOCTYPE html>
<html>
<head>
    <title>BustCom - Система управления обучением</title>
</head>
<body>
    <div class="header">
        <h1>Добро пожаловать в BustCom</h1>
        <nav>
            <a href="/courses">Курсы</a>
            <a href="/tests">Тесты</a>
            <a href="/profile">Профиль</a>
        </nav>
    </div>

    <div class="content">
        <div class="courses">
            <h2>Доступные курсы</h2>
            {% for course in courses %}
                <div class="course">
                    <h3>{{ course.name }}</h3>
                    <p>{{ course.description }}</p>
                </div>
            {% endfor %}
        </div>

        <div class="tests">
            <h2>Тесты</h2>
            {% for test in tests %}
                <div class="test">
                    <h3>{{ test.title }}</h3>
                    <p>{{ test.description }}</p>
                </div>
            {% endfor %}
        </div>

        <div class="footer">
            <p>© 2024 BustCom</p>
        </div>
    </body>
</html>
```

Рисунок 8 - код до рефакторинга

```

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>BustCom - Система управления обучением</title>
    <link rel="stylesheet" href="{% static 'css/style.css' %}">
</head>
<body>
    <header class="header">
        <h1>Добро пожаловать в BustCom</h1>
        <nav class="main-nav">
            <ul>
                <li><a href="{% url 'courses' %}">Курсы</a></li>
                <li><a href="{% url 'tests' %}">Тесты</a></li>
                <li><a href="{% url 'profile' %}">Профиль</a></li>
            </ul>
        </nav>
    </header>

    <main class="content">
        <section class="courses-section">
            <h2>Доступные курсы</h2>
            <div class="courses-grid">
                {% for course in courses %}
                    <article class="course-card">
                        <h3>{{ course.name }}</h3>
                        <p>{{ course.description }}</p>
                        <a href="{% url 'course_detail' course.id %}" class="btn">Подробнее</a>
                    </article>
                {% endfor %}
            </div>
        </section>

        <section class="tests-section">
            <h2>Тесты</h2>
            <div class="tests-grid">
                {% for test in tests %}
                    <article class="test-card">
                        <h3>{{ test.title }}</h3>
                        <p>{{ test.description }}</p>
                        <a href="{% url 'test_detail' test.id %}" class="btn">Начать тест</a>
                    </article>
                {% endfor %}
            </div>
        </section>
    </main>

```

Рисунок 9 - код после рефакторинга

2.2.4. Тесты

Подробные сценарий и результаты тестовых испытаний веб-приложения можно посмотреть в Приложении В. Сценарий и результаты тестовых испытаний.

2.2.5. Контроль целостности данных

В процессе разработки были реализованы методы валидации входных данных в информационной системе, которые обеспечивают целостность этих входных данных.

Контроль целостности, описывающих ситуации и реакции приложения на выполнения функций представлен в таблице 5.

Таблица 5 - Таблица контроля целостности

№	Ситуация	Аномалия	Реакция	Примечание
1	Регистрация	Введенные данные не соответствуют ограничениям	Приложение обработает ошибку и выведет пользователю сообщение с содержанием ошибки и выделит ошибочные поля	Регистрация возможна только с уникальным логином. Данные должны соответствовать ограничениям полей.
2	Добавление или изменение курса	Неправильно заполнены поля или введенные данные не соответствуют ограничениям	Приложение обработает ошибку и выведет пользователю сообщение с содержанием ошибки и выделит ошибочные поля	Все поля должны быть корректно заполнены. Материалы курса не являются обязательными, но поддерживаются.
3	Добавление или изменение теста	Неправильно заполнены	Приложение обработает	Тест должен содержать хотя бы

		поля или введенные данные не соответствуют ограничениям	ошибку и выведет пользователю сообщение с содержанием ошибки и выделит ошибочные поля	один вопрос. Время на выполнение должно быть положительным числом.
4	Создание или изменение ответа на тест	Неправильно заполнены поля или введенные данные не соответствуют ограничениям	Приложение обработает ошибку и выведет пользователю сообщение с содержанием ошибки и выделит ошибочные поля	Ответ должен быть дан в отведенное время. Все вопросы должны быть отвечены.
5	Добавление или изменение пользователя	Неправильно заполнены поля или введенные данные не соответствуют ограничениям	Приложение обработает ошибку и выведет пользователю сообщение с содержанием ошибки и выделит ошибочные поля	Поля должны быть корректно заполнены. Логин и email должны быть уникальными.
6	Аутентификация	Неправильно заполнены поля или введенные	Приложение обработает ошибку и выведет	Все поля обязательны к заполнению. Логин должен

		данные не соответствуют ограничениям	пользователю сообщение с содержанием ошибки и выделит ошибочные поля	существовать в базе данных.
		Пользователь с такими данными не существует	Приложение обработает ошибку и выведет пользователю сообщение о том, что такой пользователь не существует.	Убедитесь, что введённые данные соответствуют зарегистрированным пользователям.
7	Добавление вопроса преподавателю	Неправильно заполнены поля или введенные данные не соответствуют ограничениям	Приложение обработает ошибку и выведет пользователю сообщение с содержанием ошибки и выделит ошибочные поля	Тема вопроса и текст вопроса обязательны к заполнению.
8	Изменение статуса курса	Некорректный статус	Приложение обработает ошибку и выведет пользователю сообщение с содержанием	Статус может быть только "активный", "завершенный" или "черновик".

			ошибки	
--	--	--	--------	--

2.3. Проектирование

2.3.1. Схема архитектуры приложения.

Компонент пользовательский интерфейс виден пользователю и взаимодействует с ним. В зависимости от действия пользователя отправляется запрос на ORM Django, который обрабатывает этот запрос и возвращает HTML страницу в ответ на запрос.

Django взаимодействует с базой данных SQLite, на котором хранятся данные. Django позволяет совершать запросы на базу данных для получения или записи данных.

Схема архитектуры приложения представлена на рисунке 10.

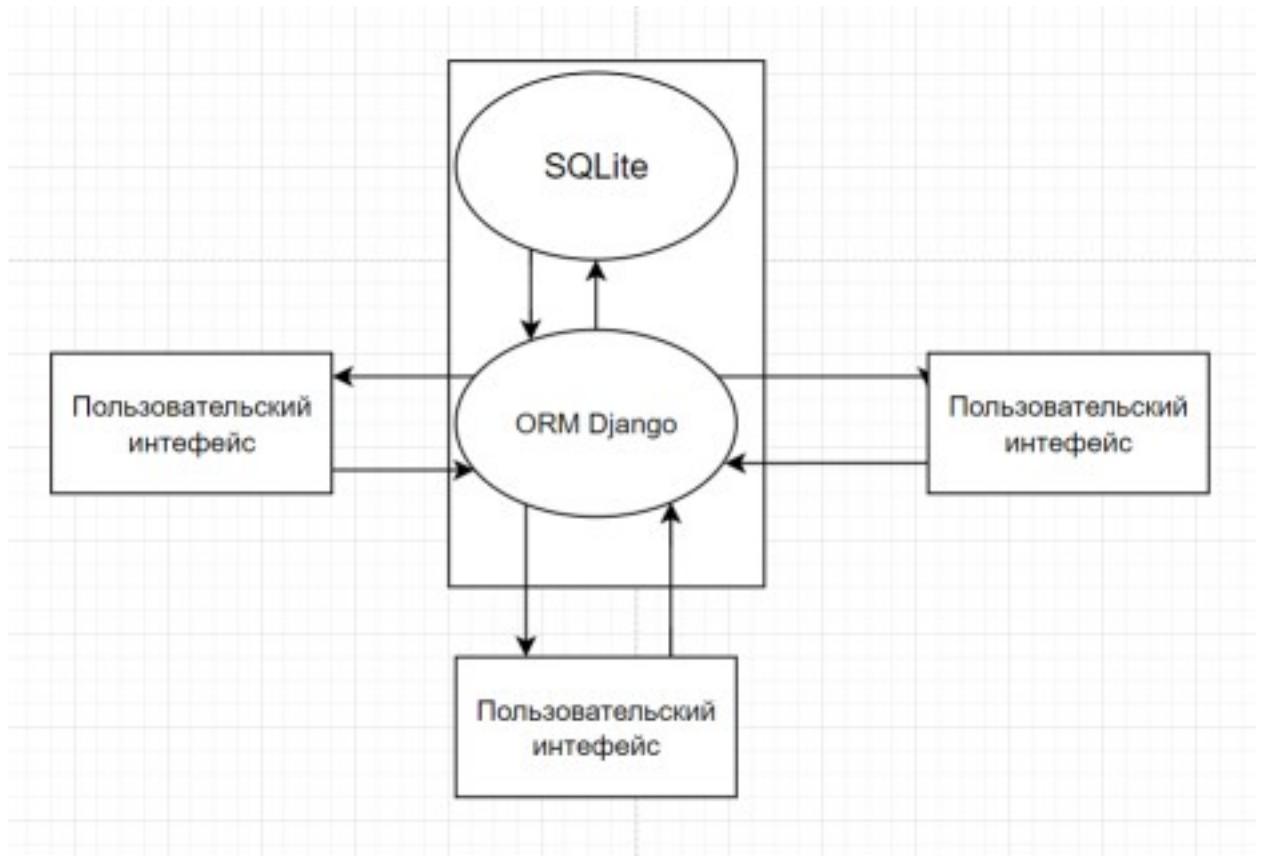


Рисунок 10 - Архитектура приложения

2.3.2. Логическая схема данных

Разработанная логическая схема базы данных, представленная на рисунке 11, послужила основой для реализации логики манипуляции данными в проектируемой базе данных.

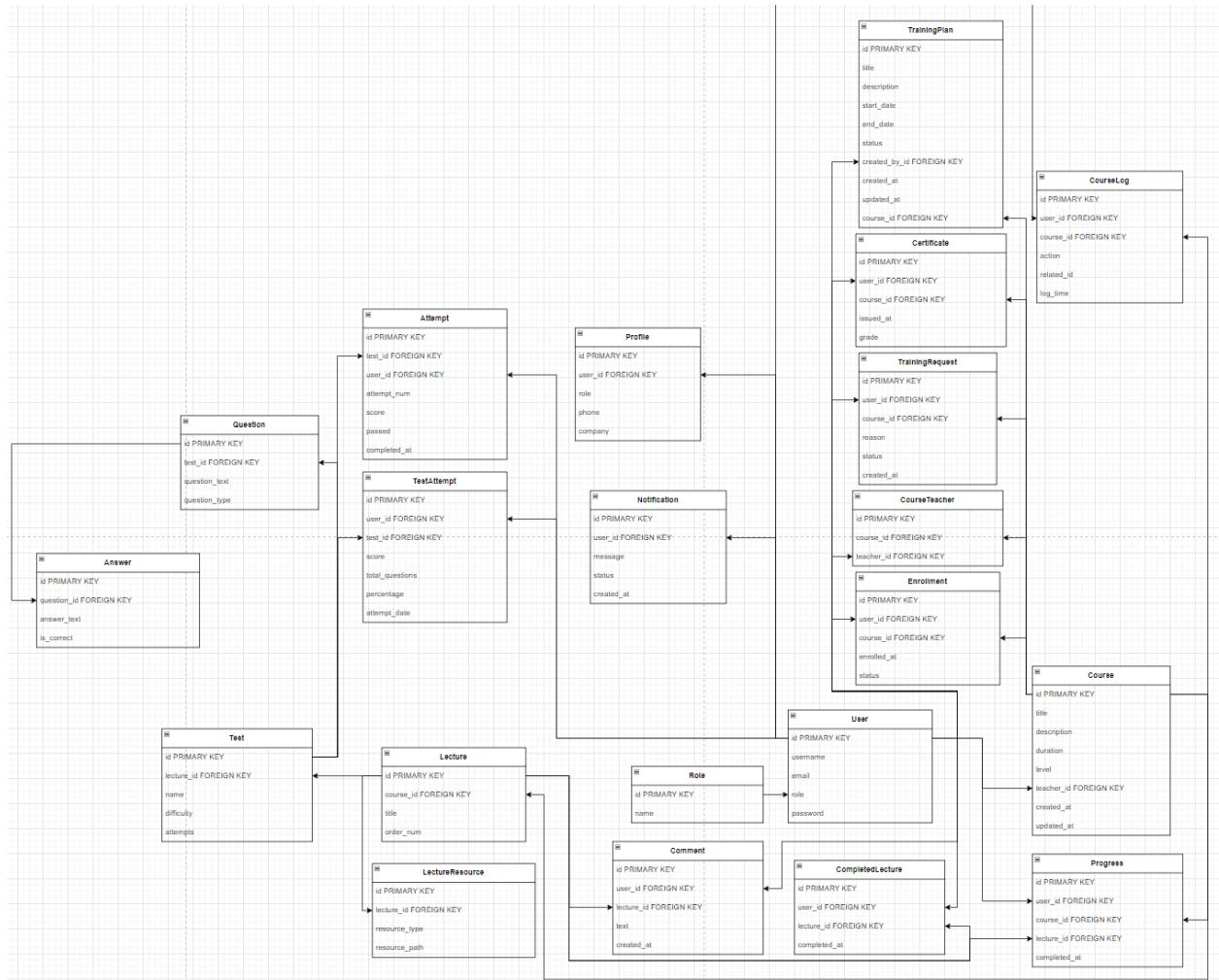


Рисунок 11 - логическая схема данных

2.3.3. Физическая схема данных

Разработанная физическая схема базы данных, представленная на рисунке 10, послужила основой для реализации логики хранения и защиты данных в проектируемой базе данных.

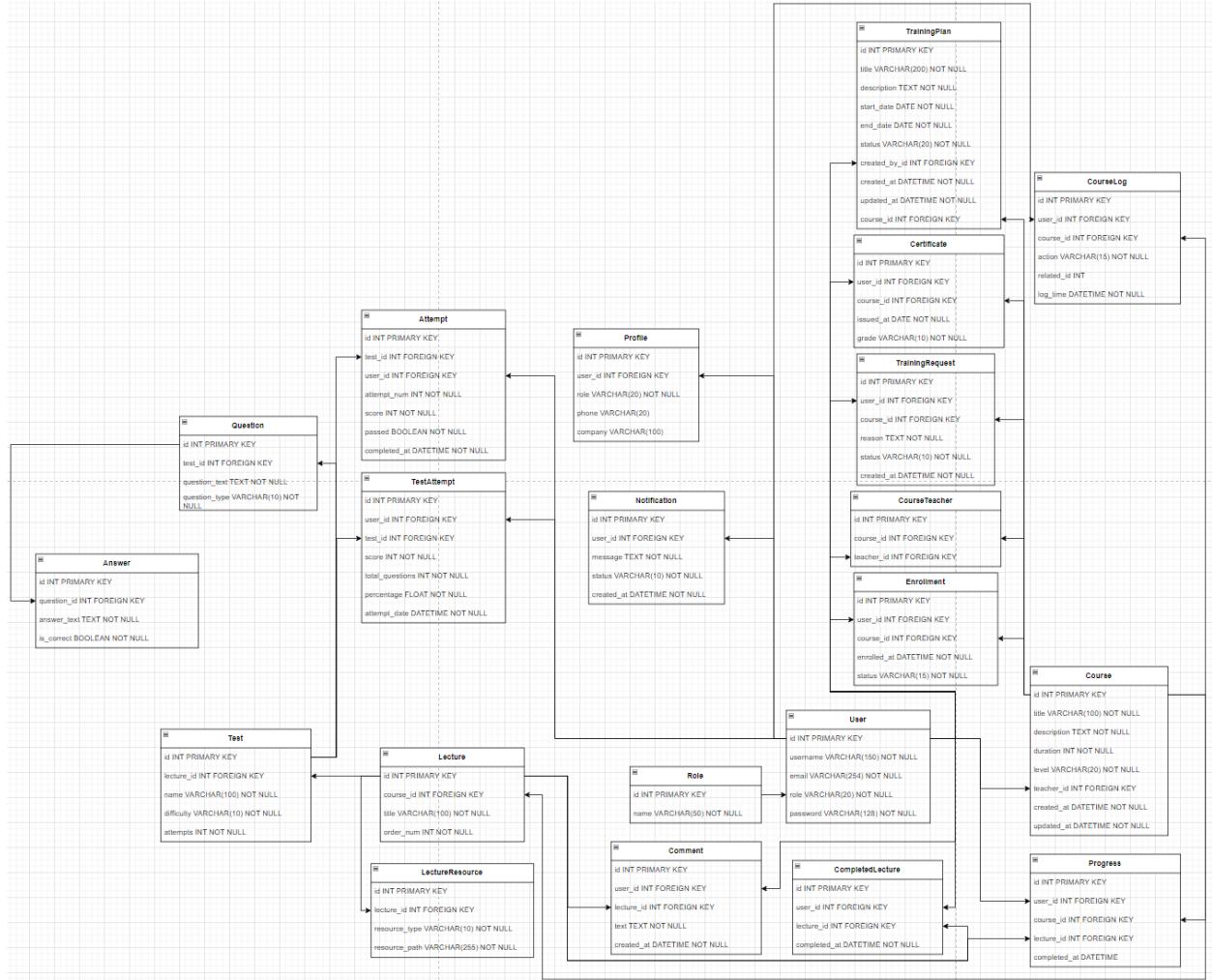


Рисунок 12 - физическая схема данных

В проектируемой базе данных выделено 12 сущностей (таблиц), каждая из которых содержит набор атрибутов (полей) для записи информации. Уникальное поле идентификатор, присутствующее в каждой таблице, используется для формирования связей между ними.

Таблица 6 - Таблица Role (Роли пользователей)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор роли	PRIMARY KEY	1
name	VARCHAR(50)	Название роли	NOT NULL, UNIQUE	'teacher'

Описание полей:

- **id**: Уникальный идентификатор роли, основной ключ таблицы.
- **name**: Название роли, обязательное для заполнения, уникальное.

Таблица 7 - Таблица User (Пользователи)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор пользователя	PRIMARY KEY	1
username	VARCHAR(150)	Имя пользователя	NOT NULL, UNIQUE	'ivan.petrov'
email	VARCHAR(254)	Электронная почта	NOT NULL, UNIQUE	'ivan@example.com'
role	VARCHAR(20)	Роль пользователя	NOT NULL	'teacher'
password	VARCHAR(128)	Пароль	NOT NULL	'hashed_password'

Описание полей:

- **id**: Основной ключ, уникальный идентификатор пользователя.
- **username**: Имя пользователя, используется для входа в систему и должно быть уникальным.
- **email**: Адрес электронной почты, также уникален для каждого пользователя.
- **role**: Роль пользователя, выбирается из предопределенных значений.
- **password**: Пароль пользователя.

Таблица 8 - Таблица Course (Курсы)

Поле	Тип данных	Описание	Ограничения	Пример данных

id	SERIAL	Уникальный идентификатор курса	PRIMARY KEY	1
title	VARCHAR(100)	Название курса	NOT NULL	'Основы Python'
description	TEXT	Описание курса		'Описание курса'
duration	INTEGER	Продолжительность курса (в часах)	NOT NULL	10
level	VARCHAR(20)	Уровень курса	NOT NULL	'beginner'
teacher_id	INTEGER	Преподаватель курса	FOREIGN KEY -> User.id	1
created_at	TIMESTAMP	Дата создания курса	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'
updated_at	TIMESTAMP	Дата обновления курса	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- **id**: Уникальный идентификатор курса.
- **title**: Название курса, обязательное для заполнения.
- **description**: Описание курса.
- **duration**: Продолжительность курса в часах.
- **level**: Уровень курса (например, 'beginner', 'intermediate', 'advanced').
- **teacher_id**: Ссылка на таблицу User, определяющая преподавателя курса.
- **created_at**: Дата и время создания курса.

- `updated_at`: Дата и время последнего обновления курса.

Таблица 9 - Таблица CourseTeacher (Преподаватели курсов)

Поле	Тип данных	Описание	Ограничения	Пример данных
<code>id</code>	SERIAL	Уникальный идентификатор записи	PRIMARY KEY	1
<code>course_id</code>	INTEGER	Курс	FOREIGN KEY -> Course.id	1
<code>teacher_id</code>	INTEGER	Преподаватель	FOREIGN KEY -> User.id	2

Описание полей:

- `id`: Уникальный идентификатор записи.
- `course_id`: Ссылка на таблицу Course, определяющая курс.
- `teacher_id`: Ссылка на таблицу User, определяющая преподавателя.

Таблица 10 - Таблица Lecture (Лекции)

Поле	Тип данных	Описание	Ограничения	Пример данных
<code>id</code>	SERIAL	Уникальный идентификатор лекции	PRIMARY KEY	1
<code>course_id</code>	INTEGER	Курс	FOREIGN KEY -> Course.id	1
<code>title</code>	VARCHAR(100)	Название лекции	NOT NULL	'Введение в Python'
<code>order_num</code>	INTEGER	Порядковый	NOT NULL	1

		номер лекции		
--	--	--------------	--	--

Описание полей:

- id: Уникальный идентификатор лекции.
- course_id: Ссылка на таблицу Course, определяющая курс.
- title: Название лекции, обязательное для заполнения.
- order_num: Порядковый номер лекции в курсе.

Таблица 11 - Таблица LectureResource (Ресурсы лекций)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор ресурса	PRIMARY KEY	1
lecture_id	INTEGER	Лекция	FOREIGN KEY -> Lecture.id	1
resource_type	VARCHAR(10)	Тип ресурса	NOT NULL	'file'
resource_path	VARCHAR(255)	Путь к ресурсу	NOT NULL	'/path/to/resource'

Описание полей:

- id: Уникальный идентификатор ресурса.
- lecture_id: Ссылка на таблицу Lecture, определяющая лекцию.
- resource_type: Тип ресурса (например, 'file', 'video').
- resource_path: Путь к ресурсу.

Таблица 12 - Таблица Enrollment (Записи на курсы)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор	PRIMARY KEY	1

		записи		
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
course_id	INTEGER	Курс	FOREIGN KEY -> Course.id	1
enrolled_at	TIMESTAMP	Дата записи на курс	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'
status	VARCHAR(15)	Статус записи	NOT NULL	'в процессе'

Описание полей:

- id: Уникальный идентификатор записи.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- course_id: Ссылка на таблицу Course, определяющая курс.
- enrolled_at: Дата и время записи на курс.
- status: Статус записи (например, 'в процессе', 'завершён', 'отчислен').

Таблица 13 - Таблица Progress (Прогресс)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор прогресса	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
course_id	INTEGER	Курс	FOREIGN KEY -> Course.id	1
lecture_id	INTEGER	Лекция	FOREIGN KEY -> Lecture.id	1

completed_at	TIMESTAMP	Дата завершения лекции		'2024-10-15 12:45:00'
--------------	-----------	------------------------	--	-----------------------

Описание полей:

- id: Уникальный идентификатор прогресса.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- course_id: Ссылка на таблицу Course, определяющая курс.
- lecture_id: Ссылка на таблицу Lecture, определяющая лекцию.
- completed_at: Дата и время завершения лекции.

Таблица 14 - Таблица Test (Тесты)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор теста	PRIMARY KEY	1
lecture_id	INTEGER	Лекция	FOREIGN KEY -> Lecture.id	1
name	VARCHAR(100)	Название теста	NOT NULL	'Тест 1'
difficulty	VARCHAR(10)	Сложность теста	NOT NULL	'низкая'
attempts	INTEGER	Количество попыток	DEFAULT 3	3

Описание полей:

- id: Уникальный идентификатор теста.
- lecture_id: Ссылка на таблицу Lecture, определяющая лекцию.

- name: Название теста, обязательное для заполнения.
- difficulty: Сложность теста (например, 'низкая', 'средняя', 'высокая').
- attempts: Количество попыток для прохождения теста.

Таблица 15 - Таблица Question (Вопросы)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор вопроса	PRIMARY KEY	1
test_id	INTEGER	Тест	FOREIGN KEY -> Test.id	1
question_text	TEXT	Текст вопроса	NOT NULL	'Что такое Python?'
question_type	VARCHAR(10)	Тип вопроса	NOT NULL	'один'

Описание полей:

- id: Уникальный идентификатор вопроса.
- test_id: Ссылка на таблицу Test, определяющая тест.
- question_text: Текст вопроса, обязательное для заполнения.
- question_type: Тип вопроса (например, 'один', 'несколько', 'ввод').

Таблица 16 - Таблица Answer (Ответы)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор ответа	PRIMARY KEY	1
question_id	INTEGER	Вопрос	FOREIGN KEY -> Question.id	1

answer_text	TEXT	Текст ответа	NOT NULL	'Python — это язык программирования.'
is_correct	BOOLEAN	Правильность ответа	NOT NULL	TRUE

Описание полей:

- id: Уникальный идентификатор ответа.
- question_id: Ссылка на таблицу Question, определяющая вопрос.
- answer_text: Текст ответа, обязательное для заполнения.
- is_correct: показывает, является ли ответ правильным.

Таблица 17 - Таблица Attempt (Попытки)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор попытки	PRIMARY KEY	1
test_id	INTEGER	Тест	FOREIGN KEY -> Test.id	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
attempt_num	INTEGER	Номер попытки	NOT NULL	1
score	INTEGER	Оценка	NOT NULL	80
passed	BOOLEAN	Показатель успешности	NOT NULL	TRUE
completed_at	TIMESTAMP	Дата завершения попытки	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор попытки.

- test_id: Ссылка на таблицу Test, определяющая тест.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- attempt_num: Номер попытки.
- score: Оценка за попытку.
- passed: Показывает, успешно ли пройден тест.
- completed_at: Дата и время завершения попытки.

Таблица 18 - Таблица Certificate (Сертификаты)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор сертификата	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
course_id	INTEGER	Курс	FOREIGN KEY -> Course.id	1
issued_at	DATE	Дата выдачи сертификата	DEFAULT CURRENT_DATE	'2024-10-15'
grade	VARCHAR(10)	Оценка	NOT NULL	'A'

Описание полей:

- id: Уникальный идентификатор сертификата.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- course_id: Ссылка на таблицу Course, определяющая курс.
- issued_at: Дата выдачи сертификата.
- grade: Оценка за курс.

Таблица 19 - Таблица Notification (Уведомления)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор уведомления	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
message	TEXT	Текст уведомления	NOT NULL	'Новое уведомление'
status	VARCHAR(10)	Статус уведомления	NOT NULL	'новое'
created_at	TIMESTAMP	Дата создания уведомления	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор уведомления.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- message: Текст уведомления, обязательное для заполнения.
- status: Статус уведомления (например, 'новое', 'прочитано').
- created_at: Дата и время создания уведомления.

Таблица 20 - Таблица CourseLog (Логи курсов)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор лога	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1

course_id	INTEGER	Курс	FOREIGN KEY -> Course.id	1
action	VARCHAR(15)	Действие	NOT NULL	'запись'
related_id	INTEGER	Связанный идентификатор		1
log_time	TIMESTAMP	Время записи лога	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор лога.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- course_id: Ссылка на таблицу Course, определяющая курс.
- action: Действие (например, 'запись', 'прогресс', 'завершение', 'сертификат').
- related_id: Связанный идентификатор (например, ID записи, прогресса и т.д.).
- log_time: Дата и время записи лога.

Таблица 21 - Таблица CompletedLecture (Завершенные лекции)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор завершенной лекции	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
lecture_id	INTEGER	Лекция	FOREIGN KEY -> Lecture.id	1
completed_at	TIMESTAMP	Дата	DEFAULT	'2024-10-15'

		завершения лекции	CURRENT_TIMESTAMP	12:45:00'
--	--	-------------------	-------------------	-----------

Описание полей:

- **id**: Уникальный идентификатор завершенной лекции.
- **user_id**: Ссылка на таблицу User, определяющая пользователя.
- **lecture_id**: Ссылка на таблицу Lecture, определяющая лекцию.
- **completed_at**: Дата и время завершения лекции.

Таблица 22 - Таблица Profile (Профили пользователей)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор профиля	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
role	VARCHAR(20)	Роль пользователя	NOT NULL	'teacher'
phone	VARCHAR(20)	Телефон		'+79991234567'
company	VARCHAR(100)	Компания		'Компания 1'

Описание полей:

- **id**: Уникальный идентификатор профиля.
- **user_id**: Ссылка на таблицу User, определяющая пользователя.
- **role**: Роль пользователя (например, 'teacher', 'student').
- **phone**: Телефон пользователя.

- company: Компания пользователя.

Таблица 23 - Таблица TestAttempt (Попытки тестов)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор попытки теста	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
test_id	INTEGER	Тест	FOREIGN KEY -> Test.id	1
score	INTEGER	Оценка	NOT NULL	80
total_questions	INTEGER	Общее количество вопросов	NOT NULL	10
percentage	FLOAT	Процент правильных ответов	NOT NULL	80.0
attempt_date	TIMESTAMP	Дата попытки	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор попытки теста.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- test_id: Ссылка на таблицу Test, определяющая тест.
- score: Оценка за попытку.
- total_questions: Общее количество вопросов в тесте.
- percentage: Процент правильных ответов.

- attempt_date: Дата и время попытки.

Таблица 24 - Таблица Comment (Комментарии)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор комментария	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
lecture_id	INTEGER	Лекция	FOREIGN KEY -> Lecture.id	1
text	TEXT	Текст комментария	NOT NULL	'Отличная лекция!'
created_at	TIMESTAMP	Дата создания комментария	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор комментария.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- lecture_id: Ссылка на таблицу Lecture, определяющая лекцию.
- text: Текст комментария, обязательное для заполнения.
- created_at: Дата и время создания комментария.

Таблица 25 - Таблица TrainingRequest (Запросы на обучение)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор запроса	PRIMARY KEY	1

user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
course_id	INTEGER	Курс	FOREIGN KEY -> Course.id	1
reason	TEXT	Причина запроса	NOT NULL	'Хочу изучить Python'
status	VARCHAR(10)	Статус запроса	NOT NULL	'pending'
created_at	TIMESTAMP	Дата создания запроса	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор запроса.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- course_id: Ссылка на таблицу Course, определяющая курс.
- reason: Причина запроса, обязательное для заполнения.
- status: Статус запроса (например, 'pending', 'approved', 'rejected').
- created_at: Дата и время создания запроса.

Таблица 26 - Таблица TrainingPlan (Планы обучения)

Поле	Тип данных	Описание	Ограничения	Пример данных
<u>id</u>	SERIAL	Уникальный идентификатор плана	PRIMARY KEY	1
<u>title</u>	VARCHAR(200)	Название плана	NOT NULL	'План обучения Python'
<u>description</u>	TEXT	Описание	NOT NULL	'Описание'

		плана		плана'
start_date	DATE	Дата начала плана	NOT NULL	'2024-10-15'
end_date	DATE	Дата окончания плана	NOT NULL	'2024-12-15'
status	VARCHAR(20)	Статус плана	NOT NULL	'draft'
created_by	INTEGER	Создатель плана	FOREIGN KEY -> User.id	1
created_at	TIMESTAMP	Дата создания плана	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'
updated_at	TIMESTAMP	Дата обновления плана	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор плана.
- title: Название плана, обязательное для заполнения.
- description: Описание плана.
- start_date: Дата начала плана.
- end_date: Дата окончания плана.
- status: Статус плана (например, 'draft', 'active', 'completed', 'cancelled').
- created_by: Ссылка на таблицу User, определяющая создателя плана.
- created_at: Дата и время создания плана.
- updated_at: Дата и время последнего обновления плана.

2.3.4. Структурная схема

На рисунке 13 представлен общий план структурной схемы веб приложения, в которой графически показано взаимодействие всех контроллеров, представлений и моделей.

Таблица 27 -Описание модулей

№	Название	Описание	Количество строк	Размер файла в Кб
1	2	3	4	5
1	manage.py	Основной скрипт управления Django проектом	23	0.7 КБ
2	db.sqlite3	База данных SQLite	126	400 КБ
3	views.py	Файл с представлениями (логика обработки запросов)	956	36 КБ
4	urls.py	Конфигурация URL-маршрутов	30	2.1 КБ
5	models.py	Модели данных проекта	320	12 КБ
6	admin.py	Настройки административной панели	39	1.3 КБ
7	forms.py	Формы для обработки данных	98	3.1 КБ
8	apps.py	Конфигурация приложения	9	0.3 КБ
9	signals.py	Обработчики сигналов Django	9	0.3 КБ
10	decorators.py	Декораторы для	8	0.4 КБ

		представлений		
11	tests.py	Модульные тесты	4	0.1 КБ
12	__init__.py	Инициализационный файл пакета	0	0 КБ
13	select_test.html	Файл разметки выбора теста	11	0.3 КБ
14	add_question.html	Файл разметки добавления вопроса	11	0.3 КБ
15	service.html	Файл разметки сервисной страницы	33	1.1 КБ
16	dashboard.html (admin)	Файл разметки панели администратора	187	5.3 КБ
17	restore_backup.html	Файл разметки восстановления резервной копии	162	4.2 КБ
18	manage_table.html	Файл разметки управления таблицами	204	5.2 КБ
19	edit_object.html	Файл разметки редактирования объекта	14	0.4 КБ
20	add_object.html	Файл разметки добавления объекта	14	0.4 КБ
21	test_difficulty.html	Файл разметки настройки сложности тестов	94	4.3 КБ
22	review_requests.html	Файл разметки просмотра заявок	100	3.6 КБ
23	create_training_plan.html	Файл разметки создания плана обучения	101	4.4 КБ

24	dashboard.html (training)	Файл разметки панели менеджера обучения	38	1.8 КБ
25	styles.css	Основные стили проекта	585	14 КБ
26	track_performance.html	Файл разметки отслеживания производительности	95	3.4 КБ
27	assign_courses.html	Файл разметки назначения курсов	80	3.4 КБ
28	dashboard.html (hr)	Файл разметки панели HR	29	1.3 КБ
29	training_evaluation.html	Файл разметки оценки обучения	9	0.3 КБ
30	assign_course.html	Файл разметки назначения курса	27	0.8 КБ
31	add_lecture_material.html	Файл разметки добавления материалов лекции	252	7.0 КБ
32	create_course.html	Файл разметки создания курса	207	5.2 КБ
33	edit_course.html	Файл разметки редактирования курса	15	0.4 КБ
34	my_certificates.html	Файл разметки просмотра личных сертификатов	322	9.1 КБ
35	view_certificate.html	Файл разметки просмотра сертификата	162	4.3 КБ

2.3.5. Функциональная схема

На рисунке 14 изображена функциональная схема веб-приложения

«BustCom». На данной схеме изображен возможности пользователя в процессе эксплуатации.

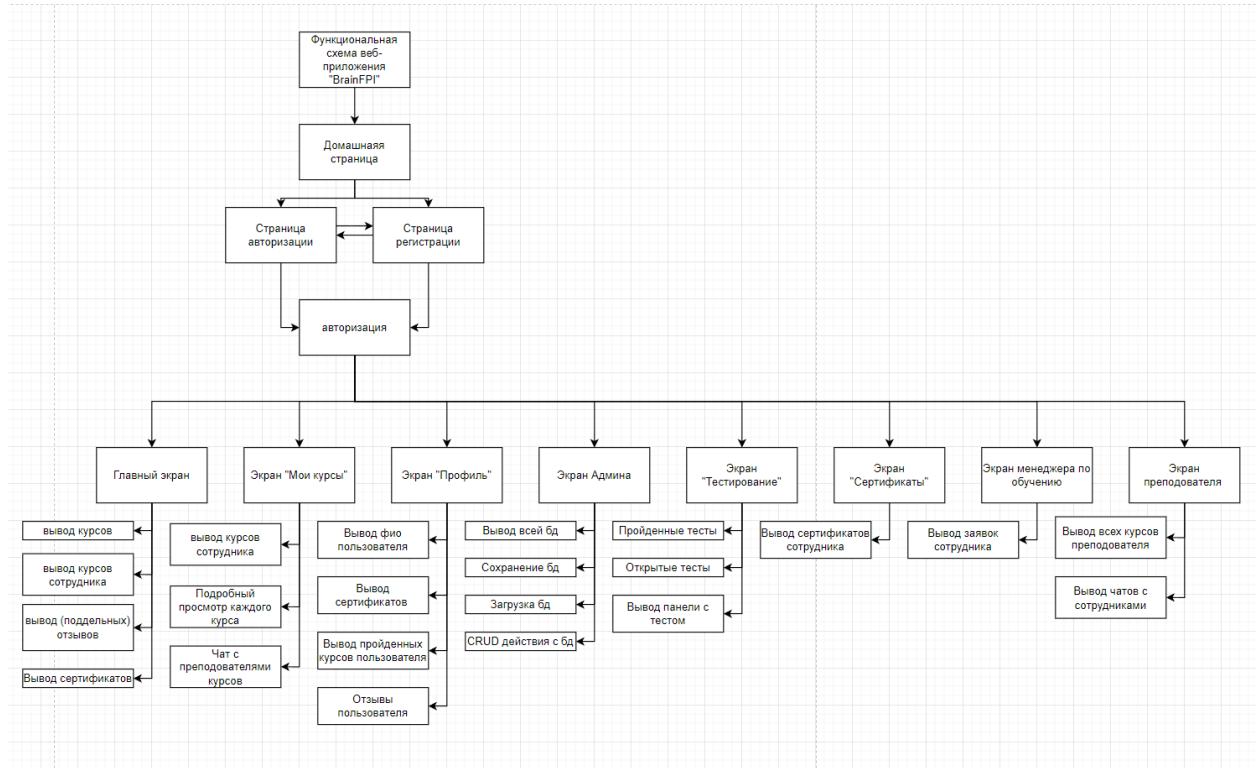


Рисунок 14 - Функциональная схема приложения

2.3.6. Схема тестирования

На рисунке 15 представлена схема тестирования веб-приложения «BustCom».

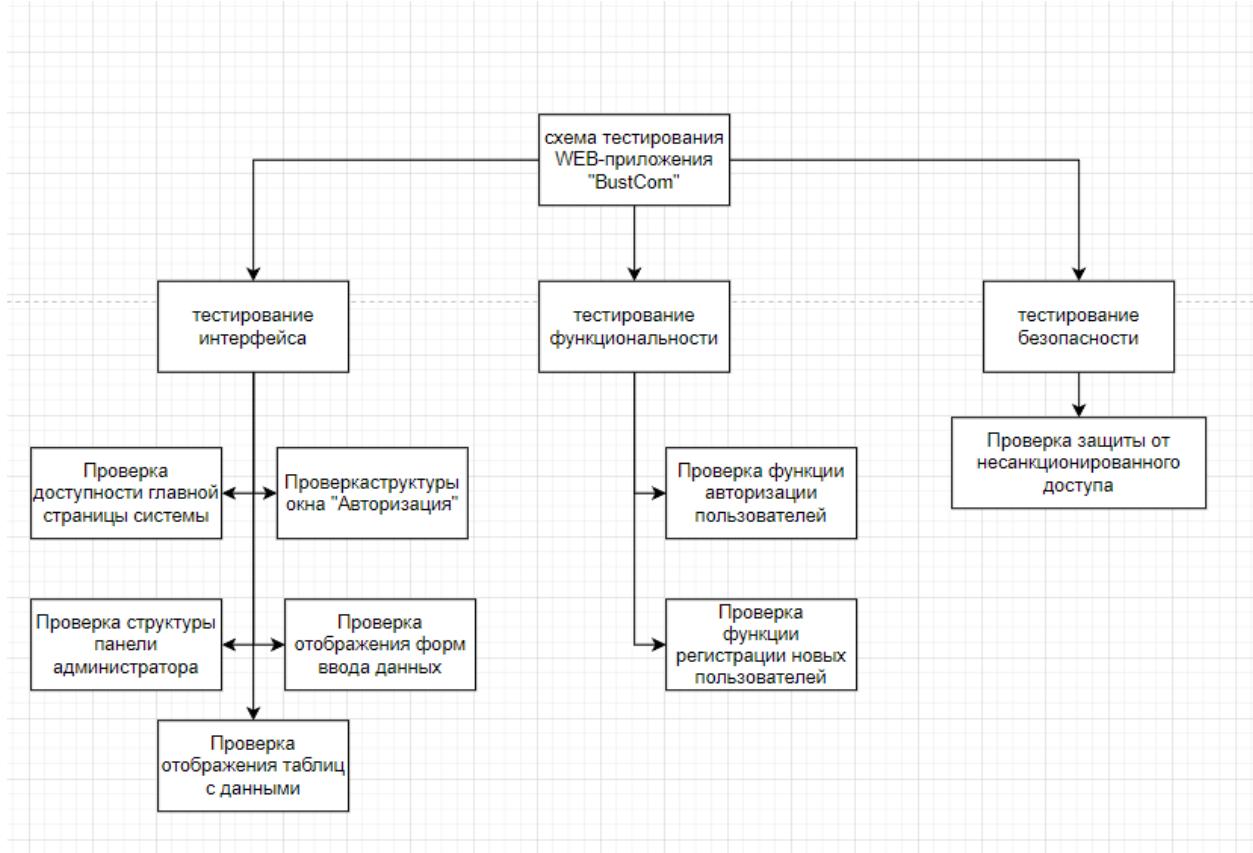


Рисунок 15 - Схема тестирования

2.3.7. Схема пользовательского интерфейса.

Согласно логике работы приложения, неавторизованный пользователь при входе на сайт попадает на главную страницу, где он может выбрать переход на страницу авторизации или регистрации.

После успешной регистрации или авторизации пользователь попадает на страницу "Панель управления". Из данной страницы в зависимости от своей роли он может перейти на следующие разделы:

1. Административная панель — доступна только пользователям с ролью администратора. Здесь предоставляются возможности управления всеми аспектами приложения:

- Управление пользователями
- Управление курсами

- Управление тестами
 - Управление категориями
 - Управление материалами
 - Просмотр отчетов и статистики
2. Профиль — личная страница пользователя, где отображается:
- Личная информация
 - Список записанных курсов
 - Результаты тестов
 - История обучения
 - Настройки профиля
3. Курсы — раздел для просмотра доступных курсов:
- Каталог курсов
 - Фильтрация по категориям
 - Поиск курсов
 - Запись на курсы
 - Отслеживание прогресса
4. Тесты — раздел для работы с тестами:
- Список доступных тестов
 - Прохождение тестов

- Просмотр результатов
- История попыток
- Статистика успеваемости

5. Вопросы преподавателю — раздел для взаимодействия с преподавателями:

- Задавание вопросов
- Просмотр ответов
- История обращений
- Статусы вопросов

6. Панель преподавателя — доступна пользователям с ролью преподавателя:

- Управление курсами
- Создание и редактирование тестов
- Ответы на вопросы сотрудников
- Просмотр статистики обучения
- Управление материалами курсов

7. Панель HR — доступна пользователям с ролью модератора:

- Модерация контента
- Проверка вопросов
- Управление отзывами

- Контроль качества материалов

8. Панель поддержки — предназначена для сотрудников службы поддержки:

- Обработка обращений пользователей
- Ответы на вопросы
- Управление статусами обращений
- Работа с жалобами

На всех страницах приложения присутствует "шапка" — верхняя панель с быстрыми ссылками для перехода между основными разделами. Это обеспечивает пользователям удобную и быструю навигацию по приложению.

Приложение адаптировано под разные роли, что делает его гибким и подходящим для выполнения задач как сотрудников, так и преподавателей с расширенными правами.

Результат работы программы:

В результате поставленной задачи было разработано требуемое программное решение в виде веб-приложения для управления образовательным процессом. На рисунках представлен результат некоторых из возможностей веб-приложения. Более подробный результат описан в Приложении Г "Руководство пользователя".

3. ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

3.1. Инструментальные средства

Для разработки веб-приложения BustCom был использован редактор кода Visual Studio Code. Этот редактор предоставляет удобные возможности для написания, отладки и тестирования кода. VSCode отличается лёгкостью, высокой производительностью и возможностью настройки под нужды разработчика благодаря множеству доступных расширений. Его широкая поддержка в сообществе разработчиков и простота интеграции с различными инструментами делают его отличным выбором для работы с Django и React.

Другие популярные редакторы, такие как PyCharm или Sublime Text, также могли бы быть использованы, но VSCode обладает преимуществами в виде лёгкости, универсальности и доступности множества плагинов, таких как встроенная поддержка Django, React и расширения для работы с Python и JavaScript. В качестве языков программирования для разработки были выбраны:

- Python (серверная часть)
- JavaScript/TypeScript (клиентская часть)

Python отличается простотой и лаконичностью синтаксиса, что делает его удобным для создания приложений любого уровня сложности. Python идеально подходит для веб-разработки благодаря своей объектно-ориентированной природе, поддержке модульности и большому количеству доступных библиотек.

JavaScript/TypeScript обеспечивает создание интерактивного пользовательского интерфейса и эффективную работу с данными на стороне клиента.

Для работы с базой данных в проекте использовалась PostgreSQL, которая обеспечивает надежное хранение данных и высокую производительность при работе с большими объемами информации. PostgreSQL отлично подходит для образовательных систем, так как позволяет эффективно хранить и обрабатывать данные о курсах, тестах и результатах обучения.

Взаимодействие с базой данных обеспечивается:

- Django ORM для серверной части
- Prisma ORM для клиентской части

Это позволяет работать с данными через объекты и устраниет необходимость написания сложных SQL-запросов.

3.2. Отладка программы

Отладка веб-приложения BustCom осуществлялась с использованием:

- Встроенного режима отладки в Django
- Отладчика в Visual Studio Code
- React Developer Tools для отладки клиентской части
- Chrome DevTools для отладки сетевых запросов

Django предоставляет подробные сообщения об ошибках в режиме разработки, включая трассировку стека, что помогает быстро находить и устранять баги. VSCode предлагает удобный интерфейс для пошаговой отладки, просмотра переменных и выполнения кода.

3.3. Защитное программирование

В программном коде предусмотрены защиты от ошибок при помощи:

- Конструкций `if...else` для проверки условий
- Блоков `try...catch` для обработки исключений
- Встроенных валидаторов данных Django
- Валидации форм на стороне клиента
- Проверки прав доступа
- Защиты от SQL-инъекций
- Защиты от XSS-атак
- CSRF-защиты

Пароли пользователей хранятся в хешированном виде с использованием алгоритма `bcrypt`.

3.4. Характеристики программы

Разработанное веб-приложение может запускаться на любых компьютерах с современными веб-браузерами (Chrome, Firefox, Safari, Edge) и операционными системами (Windows, macOS, Linux).

Характеристики программы представлены в приложении Б «Текст программы» в таблице 1 «Модули». Основные требования к системе:

- Современный веб-браузер
- Минимум 4 ГБ оперативной памяти
- Стабильное интернет-соединение

- Разрешение экрана не менее 1024x768

ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта было разработано веб-приложение для управления образовательным процессом под названием BustCom. Разработанное приложение предоставляет удобный интерфейс для управления курсами, тестами и взаимодействия между преподавателями и сотрудниками. Оно включает в себя весь функционал, заданный в техническом задании, и успешно выполняет все необходимые функции.

В ходе проекта была проведена аналитическая работа по постановке и описанию задачи, а также детальная проработка структуры приложения, включая проектирование моделей данных, определение пользовательских сценариев и схем навигации.

Перед началом разработки был проведён подготовительный этап, который включал настройку среды разработки на основе Visual Studio Code и установку необходимых библиотек, таких как Django и React. Эти инструменты позволили эффективно организовать процесс разработки.

Для хранения данных использовалась база данных PostgreSQL, что обеспечило надежное хранение и высокую производительность при работе с большими объемами данных. Благодаря использованию ORM Django и Prisma стало возможно управлять базой данных через объектно-ориентированные модели, что сократило сложность написания SQL-запросов.

На этапе разработки были реализованы ключевые модули приложения:

- Управление курсами (создание, редактирование, удаление)
- Создание и проведение тестов

- Функционал для регистрации, авторизации и управления профилем пользователя
- Интерфейс для взаимодействия с базой данных через административную панель
- Система вопросов и ответов между сотрудниками и преподавателями

Благодаря использованию языков программирования Python и JavaScript/TypeScript, приложение получилось гибким, легко масштабируемым и интуитивно понятным для разработки. Django и React предоставили встроенные инструменты для создания безопасного и производительного веб-приложения.

На этапе тестирования было тщательно проверено выполнение всех функций приложения, включая обработку ошибок, правильность работы с базой данных, корректное отображение интерфейса и безопасность пользовательских данных. Тестирование проводилось как в среде разработки Django, так и с использованием инструментов отладки в Visual Studio Code. Выявленные ошибки устранились оперативно, что позволило добиться стабильной работы приложения.

BustCom отвечает всем требованиям, предъявленным к проекту, включая удобство использования и функциональность. Однако проект имеет потенциал для дальнейшего развития. Возможные улучшения включают:

1. Реализация системы поиска курсов и материалов: добавить возможность поиска по названию, категории или уровню сложности.
2. Интеграция с системами видеоконференций, такими как Zoom или Google Meet, для проведения онлайн-занятий.

3. Создание мобильной версии интерфейса, чтобы пользователи могли работать с приложением на смартфонах.

4. Расширение аналитики обучения: добавление графиков и статистики успеваемости для преподавателей и администраторов.

5. Реализация API: предоставить сторонним приложениям возможность взаимодействовать с системой, например, для интеграции с образовательными платформами или системами управления обучением.

6. Добавление системы геймификации для повышения мотивации сотрудников.

7. Реализация системы автоматической проверки заданий с использованием искусственного интеллекта.

Таким образом, разработанное веб-приложение является удобным, функциональным и готовым к эксплуатации. Тем не менее, существует множество возможностей для его усовершенствования, что позволит сделать BustCom ещё более эффективным инструментом для управления образовательным процессом.

ПРИЛОЖЕНИЕ А. Техническое задание

АННОТАЦИЯ

В данном программном документе приведено техническое задание на тему «Разработка информационной системы «Повышение квалификации сотрудников (на примере ООО «Долфинс АЙ ТИ»)».

Содержание

1. ОБЩИЕ СВЕДЕНИЯ.....	4
1.1. Полное наименование АС и ее условное обозначение.....	4
1.2. Наименование организации — заказчика АС, наименование организаций-разработчика.....	4
1.3. Перечень документов, на основании которых создается АС.....	4
1.4. Плановые сроки начала и окончания работ по созданию АС.....	4
1.5. Общие сведения об источниках и порядке финансирования работ.....	5
2. ЦЕЛИ И НАЗНАЧЕНИЕ СОЗДАНИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ.....	6
2.1. Цели создания АС.....	6
2.2. Назначение АСАвтоматическая система имеет следующие назначения: 6	
3. ХАРАКТЕРИСТИКА ОБЪЕКТОВ АВТОМАТИЗАЦИИ.....	8
3.1. Основные сведения об объекте автоматизации.....	8
3.2. Сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды.....	8
4. ТРЕБОВАНИЯ К АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ.....	10
4.1. Требования к структуре АС.....	10
4.2. Требования к видам обеспечения АС.....	12
4.3. Общие технические требования к АС.....	13

5. СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ.....	16
5.1. Перечень этапов разработки.....	16
5.2. Сроки выполнения каждого этапа.....	16
6. ПОРЯДОК РАЗРАБОТКИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ.....	17
6.1. Организация разработки.....	17
6.2. Исходные данные.....	17
6.3. Документы на каждом этапе.....	17
6.4. Проведение экспертизы документации.....	18
6.5. Порядок утверждения и согласования.....	18
7. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ.....	19
7.1. Виды испытаний.....	19
7.2. Порядок приёмки работ Порядок приёмки:.....	19
7.3. Состав приёмочной комиссии.....	19
8. ТРЕБОВАНИЯ К СОСТАВУ И СОДЕРЖАНИЮ РАБОТ ПО ПОДГОТОВКЕ ОБЪЕКТА АВТОМАТИЗАЦИИ К ВВОДУ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ В ДЕЙСТВИЕ.....	20
8.1. Условия функционирования объекта.....	20
8.2. Организационно-штатные мероприятия.....	20
8.3. Обучение персонала.....	20
9. ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ.....	21

9.1.	Перечень подлежащих разработке документов.....	21
9.2.	Формат и количество документов.....	21
9.3.	Использование ЕСКД и ЕСПД.....	21
10.	ИСТОЧНИКИ РАЗРАБОТКИ.....	22
10.1.	Технико-экономическое обоснование.....	22
10.2.	Научные исследования и отчёты.....	22
10.3.	Аналоги систем.....	22

1. ОБЩИЕ СВЕДЕНИЯ

1.1. Полное наименование АС и ее условное обозначение

Полное наименование – «Система для повышение квалификации сотрудников» Условное обозначение – «Дольфинс АйТи»

1.2. Наименование организации — заказчика АС, наименование организации-разработчика

Организация-заказчик: «Дольфинс АйТи»

Организация разработчика: Московский Приборостроительный Техникум

1.3. Перечень документов, на основании которых создается АС

Федеральный государственный стандарт по 09.02.07 «Информационные технологии и программирования».

Распоряжение об утверждении тем курсовых работ (проектов) по дисциплине «МДК 02.01 Технология разработки программного обеспечения».

1.4. Плановые сроки начала и окончания работ по созданию АС

- Постановка задачи: 21.01.2025 – 24.01.2025
- Проектирование: 24.01.2025 – 31.01.2025
- Разработка ПО: 31.01.2025 – 28.02.2025
- Тестирование: 28.02.2025 – 14.03.2025
- Внедрение: 14.03.2025 – 04.04.2025

1.5. Общие сведения об источниках и порядке финансирования работ.

Не финансируется.

2. ЦЕЛИ И НАЗНАЧЕНИЕ СОЗДАНИЯ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ

2.1. Цели создания АС

Целями создания автоматической системы являются:

- Автоматизация процесса повышения квалификации сотрудников ООО "Долфинс АЙ ТИ"
- Сокращение времени обработки запросов на обучение
- Уменьшение количества ошибок в организации обучающих мероприятий
- Обеспечение доступа к актуальным учебным материалам в режиме реального времени
- Оптимизация процессов тестирования и оценки знаний сотрудников
- Повышение эффективности управления процессом обучения персонала

2.2. Назначение АС

Автоматическая система имеет следующие назначения:

- Автоматизация бизнес-процесса "Повышение квалификации сотрудников"
- Управление каталогом учебных курсов и программ
- Формирование индивидуальных планов обучения сотрудников
- Сбор статистики по результатам обучения

- Формирование и выдача сертификатов о прохождении курсов
- Оценка эффективности учебных программ

3. ХАРАКТЕРИСТИКА ОБЪЕКТОВ АВТОМАТИЗАЦИИ

3.1. Основные сведения об объекте автоматизации

Объект автоматизации:

- Бизнес-процесс "Повышение квалификации сотрудников"
- Не более 1000 пользователей (сотрудников)
- Управление курсами повышения квалификации
- Проведение тестирования знаний
- Отслеживание прогресса обучения
- Формирование отчетов по результатам
- Уведомление о новых курсах и тестах
- Управление расписанием занятий
- Ведение статистики успеваемости
- Анализ эффективности обучения

3.2. Сведения об условиях эксплуатации объекта автоматизации и характеристиках окружающей среды.

Условия эксплуатации:

- Доступ через веб-интерфейс для удаленного обучения
- Доступ в интернет
- Возможность работы как в офисе, так и удаленно

- Поддержка различных устройств (компьютеры, планшеты, смартфоны)
- Работа в защищенном режиме

Характеристики окружающей среды:

- Температура эксплуатации: от +10°C до +35°C
- Относительная влажность: не более 70%
- Постоянное подключение к интернету со скоростью не меньше 100 Мбит/с
- Наличие достаточного дискового пространства для хранения учебных материалов
- Система резервного копирования данных
- Система контроля доступа к учебным материалам
- Система логирования действий пользователей
- Система оповещения о новых курсах и тестах
- Система восстановления после сбоев
- Система обновления учебных материалов

3.3. Разграничение функций между пользователями системы

1. Сотрудник

Функции:

- Просмотр доступных курсов повышения квалификации
- Запись на курсы (самостоятельно или по назначению)
- Доступ к учебным материалам курсов
- Прохождение тестов и проверочных заданий
- Просмотр собственных результатов и прогресса обучения
- Получение сертификатов о прохождении курсов
- Предоставление обратной связи по пройденным курсам
- Отправка вопросов преподавателям

2. Преподаватель

Функции:

- Создание и редактирование курсов по своей специализации
- Разработка учебных материалов различных форматов
- Создание тестов и проверочных заданий
- Проверка практических заданий сотрудников
- Проведение онлайн-консультаций
- Ответы на вопросы сотрудников

3. HR-менеджер

Функции:

- Управление каталогом курсов
- Формирование плана обучения сотрудников
- Назначение курсов сотрудникам
- Контроль прохождения обучения
- Создание индивидуальных планов развития
- Анализ потребностей в обучении
- Формирование отчетов для руководства
- Мониторинг эффективности обучения

4. Трейнинг-менеджер

- Создание плана обучения
- Работа с заявками на курс
- Мониторинг активности

5. Администратор системы

Функции:

- Управление учетными записями пользователей
- Назначение ролей и прав доступа
- Управление структурой организации в системе
- Настройка параметров системы

- Мониторинг производительности системы
- Резервное копирование и восстановление данных
- Техническая поддержка пользователей

4. ТРЕБОВАНИЯ К АВТОМАТИЗИРОВАННОЙ СИСТЕМЕ

4.1. Требования к структуре АС.

4.1.1. Перечень подсистем, их назначение и основные характеристики.

Автоматизированная система имеет следующие подсистемы:

- Подсистема управления пользователями;
- Подсистема управления курсами;
- Подсистема тестирования;
- Подсистема управления лекциями;

4.1.1.1. Временной регламент реализации каждой функции

Каждая из функций должна выполняться максимум до 2 минут.

4.1.1.2. Требования к реализации каждой функции

Подсистема управления пользователями:

- Регистрация и авторизация пользователей
- Управление профилями пользователей
- Управление правами доступа

- Отслеживание активности пользователей

Подсистема управления курсами:

- Создание и редактирование курсов
- Управление материалами курсов
- Формирование расписания занятий
- Отслеживание прогресса обучения

Подсистема тестирования:

- Создание и редактирование тестов
- Проведение тестирования
- Оценка результатов
- Формирование отчетов по результатам

Управление материалами курсов

- Загрузка различных типов контента
- Структурирование материалов по темам

Управление правами доступа

- Настройка прав для различных ролей
- Разграничение доступа к разделам системы
- Назначение и изменение ролей

Подсистема отчетности:

- Статистика успеваемости
- Анализ эффективности курсов

4.1.1.3. Перечень и критерии отказов для каждой функции

Критерии отсутствуют.

4.1.2. Требования к способам и средствам обеспечения информационного взаимодействия компонентов АС

Интеграция с базой данных реализована через SQLite.

4.1.3. Требования к характеристикам взаимосвязей создаваемой АС со смежными АС

Корпоративная система управления персоналом (HRM)

- Синхронизация данных о сотрудниках
- Автоматическое обновление всех данных сотрудников

4.1.4. Требования к режимам функционирования АС

Система поддерживает следующие режимы:

- Нормальный режим работы
- Режим обслуживания
- Аварийный режим
- Режим тестирования

4.1.5. Перспективы развития, модернизации АСВ системе должна быть возможность:

- Добавления новых курсов и тестов

- Расширения функционала отчетности
- Интеграции с системами видеоконференций
- Добавления новых типов тестов
- Расширения аналитических возможностей

4.2. Требования к видам обеспечения АС

4.2.1. Требования к математическому виду обеспечения АСВ АС должны присутствовать:

- Методы расчета статистики
- Алгоритмы оценки знаний
- Методы анализа эффективности обучения

4.2.2. Требования к информационному виду обеспечения АС

Структура базы данных

- Реляционная модель данных с нормализацией до ЗНФ
- Обеспечение целостности данных через систему ограничений и внешних ключей
- Индексирование полей, участвующих в частых запросах

Информационные массивы

- Учебные материалы различных форматов
- База тестовых вопросов и ответов
- Архив результатов обучения

4.2.3. Требования к лингвистическому виду обеспечения АС

Автоматическая система должна поддерживать русский язык.

4.2.4. Требования к программному виду обеспечения АС

АС должна быть реализована в виде WEB-приложения.

4.2.5. Требования к техническому виду обеспечения АС

Серверное оборудование

- Процессор: не ниже Intel Xeon E5 или аналог
- Оперативная память: не менее 16 ГБ
- Дисковое пространство: не менее 500 ГБ с возможностью расширения
- Резервирование ключевых компонентов

Клиентские устройства

- Минимальные требования: процессор 2 ГГц, 4 ГБ ОЗУ
- Поддержка разрешения экрана от 1366x768
- Подключение к сети со скоростью не менее 1 Мбит/с
- Наличие аудиовыхода для мультимедийных материалов

4.2.6. Требования к метрологическому виду обеспечения АС

Требования отсутствуют.

4.2.7. Требования к организационному виду обеспечения АС

Структура управления системой

- Распределение ролей и ответственности между участниками
- Процедуры принятия решений по изменениям
- Регламенты взаимодействия администраторов и пользователей
- Схема эскалации проблем

Методы и средства организации работы персонала

- Инструкции по использованию системы для разных категорий пользователей
- Процедуры обработки запросов пользователей
- Регламенты создания и публикации учебного контента
- Порядок проведения обучающих мероприятий

4.2.8. Требования к методическому виду обеспечения АС

Требования отсутствуют.

4.2.9. Требования к другим видам обеспечения АС

Требования отсутствуют.

4.3. Общие технические требования к АС.

4.3.1. Требования к численности и квалификации персонала и пользователей АС

- Администратор системы: 1 человек
- Преподаватели: в зависимости от количества курсов
- Специалист технической поддержки: 1 человек
- Модераторы контента: 1-2 человека

4.3.2. Требования к показателям назначения

- Загрузка страниц не должна занимать более 2 минут
- Время ответа системы не более 1 секунды
- Поддержка одновременной работы не менее 1000 пользователей

4.3.3. Требования к надежности

- Время восстановления после сбоя: не более 10 минут
- Доступность системы: 99.9%
- Автоматическое резервное копирование данных

4.3.4. Требования по безопасности

- Аутентификация пользователей через почту и пароль
- Пароль должен быть хэширован
- Двухфакторная аутентификация для администраторов
- Шифрование данных при передаче

4.3.5. Требования к эргономике и технической эстетике

- Минималистичный дизайн интерфейса
- Адаптивный дизайн для различных устройств
- Интуитивно понятная навигация
- Поддержка темной темы

4.3.6. Требования к транспортабельности для подвижных АС

Адаптивность для мобильных устройств

- Поддержка любых размеров экрана от смартфонов до настольных компьютеров

- Жесты для управления на сенсорных устройствах

Автономный режим работы

- Синхронизация при восстановлении соединения

4.3.7. Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов АС

- АС должна располагаться на сервере с размером свободного дискового пространства не менее 25 ГБ

- Регулярное обновление системы безопасности

- Мониторинг производительности системы

4.3.8. Требования к защите информации от несанкционированного доступа

- Разграничение прав доступа

- Логирование действий пользователей

- Защита от SQL-инъекций

- Защита от XSS-атак

4.3.9. Требования по сохранности информации при авариях

- Регулярное резервное копирование

- Система восстановления данных

- Дублирование критических компонентов

4.3.10. Требования к защите от влияния внешних воздействий

Устойчивость к сбоям электропитания

- Размещение в data-центре с резервным электропитанием
- Корректное завершение работы при отключении электричества
- Автоматический запуск после восстановления питания

4.3.11. Требования к патентной чистоте и патентоспособности

Использование открытого ПО и стандартов

- Все компоненты системы основаны на открытом ПО с совместимыми лицензиями
 - Соблюдение общепринятых стандартов веб-разработки
 - Документирование всех сторонних компонентов и их лицензий

Оригинальность технических решений

- Уникальные алгоритмы оценки эффективности обучения
- Собственные методики построения индивидуальных планов обучения
 - Потенциально патентоспособные элементы системы адаптивного обучения

4.3.12. Требования по стандартизации и унификации

Соответствие стандартам

- Соблюдение стандартов ГОСТ Р ИСО/МЭК 12207-2010 (процессы жизненного цикла ПО)

- Соответствие требованиям W3C по доступности веб-контента (WCAG 2.1)

Унификация интерфейсов

- Единая система обозначений и терминов
- Единый подход к организации пользовательского интерфейса

4.3.13. Дополнительные требования

Требования отсутствуют.

5. СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ

5.1. Перечень этапов разработки.

Перечень этапов:

1. Постановка задачи и формирование требований.
 2. Разработка технического задания.
 3. Проектирование структуры базы данных и пользовательского интерфейса.
 4. Разработка программного обеспечения.
 5. Тестирование системы.
 6. Внедрение системы в учебный процесс.
 7. Сопровождение и обучение пользователей.
- 5.2. Сроки выполнения каждого этапа.

Постановка задачи: 21.01.2025 – 24.01.2025

Проектирование: 24.01.2025 – 31.01.2025

Разработка ПО: 31.01.2025 – 28.02.2025

Тестирование: 28.02.2025 – 14.03.2025

Внедрение: 14.03.2025 – 04.04.2025

6. ПОРЯДОК РАЗРАБОТКИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ

6.1. Организация разработки

Организация разработки:

- Менеджер проекта: контроль выполнения этапов и координация с заказчиком
- Системный аналитик: формирование требований и разработка ТЗ
- Проектировщик: проектирование базы данных и интерфейса
- Программист: реализация функциональности
- Тестировщик: проверка системы на соответствие требованиям
- Методист: разработка учебных материалов и методических рекомендаций
- Дизайнер: разработка пользовательского интерфейса
- Специалист по безопасности: обеспечение защиты данных

6.2. Исходные данные

Исходные данные:

- Приказ о назначении темы курсовой работы Кириллину Кириллу Михайловичу

6.3. Документы на каждом этапе

Документы на каждом этапе:

- Постановка задачи: техническое задание.

- Проектирование: эскизный проект, технический проект.
- Разработка: спецификация программы, текст программы.
- Тестирование: программа и методика испытаний, протокол испытаний.
- Внедрение: руководство пользователя.

6.4. Проведение экспертизы документации

Экспертиза документации:

- Экспертиза проводится руководителем учебной практики.
- Проверяются соответствие ТЗ требованиям ГОСТ, полнота проектной документации.

6.5. Порядок утверждения и согласования

Порядок утверждения:

- Техническое задание утверждается директором учебного заведения.
- Проектная документация согласовывается с методическим отделом.
- Программы испытаний утверждаются руководителем проекта.

7. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ.

7.1. Виды испытаний

Виды испытаний:

- Предварительные испытания: проверка функциональности системы на уровне разработчика.
- Приёмочные испытания: проверка системы заказчиком на соответствие ТЗ.

7.2. Порядок приёмки работ

Порядок приёмки:

- Проведение приёмочных испытаний.
- Подготовка отчёта о результатах испытаний.
- Подписание акта приёмки-передачи системы.

7.3. Состав приёмочной комиссии

Состав приёмочной комиссии:

- Руководитель проекта (заказчик).
- Главный разработчик.

8. ТРЕБОВАНИЯ К СОСТАВУ И СОДЕРЖАНИЮ РАБОТ ПО ПОДГОТОВКЕ ОБЪЕКТА АВТОМАТИЗАЦИИ К ВВОДУ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ В ДЕЙСТВИЕ.

8.1. Условия функционирования объекта

Условия функционирования:

- Подготовка серверной комнаты с климат-контролем.
- Обеспечение сетевого подключения 100 Мбит/с.

8.2. Организационно-штатные мероприятия

Организационно-штатные мероприятия:

- Назначение администратора системы из числа ИТ-персонала.
- Проведение инструктажа для сотрудников.

8.3. Обучение персонала

Обучение персонала:

- Проведение двухдневного тренинга для персонала.

9. ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ.

9.1. Перечень подлежащих разработке документов

Перечень документов:

- Техническое задание.
- Руководство пользователя.
- Программа и методика испытаний.
- Текст программы
- Скрипт базы данных
- Входные и выходные данные

9.2. Формат и количество документов

Формат и количество документов:

- Техническое задание: 2 экземпляра, формат А4.
- Руководство пользователя: 10 экземпляров, электронный формат PDF.

9.3. Использование ЕСКД и ЕСПД

Использование стандартов:

- Разработка документов в соответствии с ГОСТ 19.106-78 (ЕСПД).
- Применение графических схем согласно ГОСТ 2.701-84 (ЕСКД).

10. ИСТОЧНИКИ РАЗРАБОТКИ.

10.1. Технико-экономическое обоснование

- Отсутствует.

10.2. Научные исследования и отчёты

- Отсутствуют.

10.3. Аналоги систем

Аналоги систем:

- W3SSchhools: <https://profile.w3schools.com/>

ПРИЛОЖЕНИЕ Б. Текст программы

АННОТАЦИЯ

В данном программном документе «Текст программы» приведен исходный текст программного изделия.

В документе представлен раздел «Текст программы» с подразделами:

«Наименование программы»

«Область применения программы»

«Модули»

В подразделе «Наименование программы» содержится наименование самой программы.

В подразделе «Область применения» содержится описание области, в которой применяется программа.

В подразделе «Модули» содержится перечисление всех фалов программы с детальной информацией.

Содержание

1.	ТЕКСТ ПРОГРАММЫ.....	3
1.1.	Наименование объекта.....	3
1.2.	Область применения объекта.....	3
1.3.	Модули.....	3
1.4.	Код программы.....	6

1. ТЕКСТ ПРОГРАММЫ

1.1. Наименование объекта

Наименование – «Разработка информационной системы «Повышение квалификации сотрудников (на примере ООО «Долфинс АЙ ТИ»)».

1.2. Область применения объекта

Приложение предназначено для автоматизации процесса повышения квалификации сотрудников. Оно включает в себя учет данных о пройденных курсах, регистрацию на новые программы обучения, доступ к учебным материалам, мониторинг прогресса сотрудников, обратную связь по курсам, а также анализ эффективности обучения.

1.3. Модули

Таблица 1. Модули

№	Название	Описание	Количество строк	Размер файла в КБ
1	2	3	4	5
1	manage.py	Основной скрипт управления Django проектом	23	0.7 КБ
2	db.sqlite3	База данных SQLite	126	400 КБ
3	views.py	Файл с представлениями (логика обработки запросов)	956	36 КБ
4	urls.py	Конфигурация URL-маршрутов	30	2.1 КБ
5	models.py	Модели данных проекта	320	12 КБ
6	admin.py	Настройки административной панели	39	1.3 КБ

7	forms.py	Формы для обработки данных	98	3.1 КБ
8	apps.py	Конфигурация приложения	9	0.3 КБ
9	signals.py	Обработчики сигналов Django	9	0.3 КБ
10	decorators.py	Декораторы для представлений	8	0.4 КБ
11	tests.py	Модульные тесты	4	0.1 КБ
12	__init__.py	Инициализационный файл пакета	0	0 КБ
13	select_test.html	Файл разметки выбора теста	11	0.3 КБ
14	add_question.html	Файл разметки добавления вопроса	11	0.3 КБ
15	service.html	Файл разметки сервисной страницы	33	1.1 КБ
16	dashboard.html (admin)	Файл разметки панели администратора	187	5.3 КБ
17	restore_backup.html	Файл разметки восстановления резервной копии	162	4.2 КБ
18	manage_table.html	Файл разметки управления таблицами	204	5.2 КБ
19	edit_object.html	Файл разметки редактирования объекта	14	0.4 КБ
20	add_object.html	Файл разметки добавления объекта	14	0.4 КБ
21	test_difficulty.html	Файл разметки настройки сложности тестов	94	4.3 КБ
22	review_requests.html	Файл разметки просмотра заявок	100	3.6 КБ
23	create_training_plan.html	Файл разметки создания плана обучения	101	4.4 КБ

24	dashboard.html (training)	Файл разметки панели менеджера обучения	38	1.8 КБ
25	styles.css	Основные стили проекта	585	14 КБ
26	track_performance.html	Файл разметки отслеживания производительности	95	3.4 КБ
27	assign_courses.html	Файл разметки назначения курсов	80	3.4 КБ
28	dashboard.html (hr)	Файл разметки панели HR	29	1.3 КБ
29	training_evaluation.html	Файл разметки оценки обучения	9	0.3 КБ
30	assign_course.html	Файл разметки назначения курса	27	0.8 КБ
31	add_lecture_material.html	Файл разметки добавления материалов лекции	252	7.0 КБ
32	create_course.html	Файл разметки создания курса	207	5.2 КБ
33	edit_course.html	Файл разметки редактирования курса	15	0.4 КБ
34	my_certificates.html	Файл разметки просмотра личных сертификатов	322	9.1 КБ
35	view_certificate.html	Файл разметки просмотра сертификата	162	4.3 КБ

1.4. Код программы

manage.py

```
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'bustCom.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
        )
```

```

    "available on your PYTHONPATH environment variable? Did you "
    "forget to activate a virtual environment?""
 )from exc
execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()

```

views.py

```

from django.contrib.auth import authenticate, login, logout, get_user_model
from django.contrib.auth.decorators import login_required, user_passes_test
from django.contrib import messages
from django.shortcuts import render, redirect, get_object_or_404
from django.db.models import Avg, Max, Min, Count
from django.http import JsonResponse, HttpResponseRedirect, HttpResponseForbidden, FileResponse
from django.forms import modelform_factory, Form, FileField
import os
import datetime
import sqlite3
import tempfile
import shutil

from .models import (
    User, Role, Course, CourseTeacher, Lecture, LectureResource, Enrollment, Progress,
    Test, Question, Answer, Attempt, Certificate, Notification, CourseLog, CompletedLecture,
    Profile, Comment, TrainingRequest, TestAttempt, TrainingPlan
)
from .forms import (
    CustomUserCreationForm, ProfileForm, CommentForm, LectureForm, TestForm, QuestionForm,
    AnswerForm, CourseForm
)
from .decorators import role_required

def role_required(role):
    """Декоратор для проверки роли пользователя"""
    def check_role(user):
        return user.is_authenticated and getattr(user.profile, 'role', '').lower() == role.lower()
    return user_passes_test(check_role)

def roles_required(*roles):
    """Декоратор для проверки нескольких ролей"""
    def check_role(user):
        return user.is_authenticated and getattr(user.profile, 'role', None) in roles
    return user_passes_test(check_role)

MODEL_MAP = {
    'users': get_user_model(),
    'roles': Role,
    'courses': Course,
    'course_teachers': CourseTeacher,
    'lectures': Lecture,
    'lecture_resources': LectureResource,
    'enrollments': Enrollment,
    'progress': Progress,
    'tests': Test,
    'questions': Question,
    'answers': Answer,
    'attempts': Attempt,
    'certificates': Certificate,
    'notifications': Notification,
    'course_logs': CourseLog,
}

```

```

    'completed_lectures': CompletedLecture,
}

TABLES = {
    'users': 'Пользователи',
    'roles': 'Роли',
    'courses': 'Курсы',
    'lectures': 'Лекции',
    'enrollments': 'Записи на курсы',
    'tests': 'Тесты',
    'questions': 'Вопросы',
    'answers': 'Ответы',
    'attempts': 'Попытки тестов',
    'certificates': 'Сертификаты',
}
}

ROLE_DASHBOARD = {
    'admin': 'admin_dashboard',
    'teacher': 'teacher_dashboard',
    'hr': 'hr_dashboard',
    'training_manager': 'training_manager_dashboard',
    'employee': 'employee_dashboard',
}
}

@role_required('admin')
def admin_dashboard(request):
    return render(request, 'admin_panel/dashboard.html', {'tables': TABLES})

@role_required('admin')
def manage_table(request, table_name):
    model = MODEL_MAP.get(table_name)
    if not model:
        return JsonResponse({'error': 'Неверное имя таблицы'}, status=400)

    objects = model._default_manager.all()
    fields = [field.name for field in model._meta.get_fields()]
    object_values = objects.values_list()

    return render(request, 'admin_panel/manage_table.html', {
        'objects': objects,
        'object_values': object_values,
        'table_name': table_name,
        'fields': fields
    })

@role_required('admin')
def add_object(request, table_name):
    model = MODEL_MAP.get(table_name)
    if not model:
        return JsonResponse({'error': 'Неверное имя таблицы'}, status=400)

    form_class = modelform_factory(model, fields="__all__")
    form = form_class()

    if request.method == 'POST':
        form = form_class(request.POST)
        if form.is_valid():
            form.save()
            return redirect('manage_table', table_name=table_name)

    return render(request, 'admin_panel/add_object.html', {'form': form, 'table_name': table_name})

```

```

@role_required('admin')
def edit_object(request, table_name, object_id):
    model = MODEL_MAP.get(table_name)
    if not model:
        return JsonResponse({'error': 'Неверное имя таблицы'}, status=400)

    obj = get_object_or_404(model, pk=object_id)
    form_class = modelform_factory(model, fields="__all__")
    form = form_class(instance=obj)

    if request.method == 'POST':
        form = form_class(request.POST, instance=obj)
        if form.is_valid():
            form.save()
            return redirect('manage_table', table_name=table_name)

    return render(request, 'admin_panel/edit_object.html', {'form': form, 'table_name': table_name})

@role_required('admin')
def delete_object(request, table_name, object_id):
    model = MODEL_MAP.get(table_name)
    if not model:
        return JsonResponse({'error': 'Неверное имя таблицы'}, status=400)

    obj = get_object_or_404(model, id=object_id)
    obj.delete()
    return redirect('manage_table', table_name=table_name)

@login_required
@role_required('teacher')
def add_test(request, lecture_id):
    lecture = get_object_or_404(Lecture, id=lecture_id)

    if request.method == 'POST':
        form = TestForm(request.POST)
        if form.is_valid():
            test = form.save(commit=False)
            test.lecture = lecture
            test.save()
            return redirect('lecture_detail', lecture.id)
    else:
        form = TestForm()

    return render(request, 'bustComProj/add_test.html', {'form': form, 'lecture': lecture})

@login_required
@role_required('teacher')
def add_question(request, test_id):
    test = get_object_or_404(Test, id=test_id)

    if request.method == "POST":
        form = QuestionForm(request.POST)
        if form.is_valid():
            question = form.save(commit=False)
            question.test = test
            question.save()
            return redirect('add_answer', question.id)
    else:
        form = QuestionForm(initial={'test': test})

    return render(request, 'bustComProj/add_question.html', {'form': form, 'test': test})

@login_required

```

```

@role_required('teacher')
def add_answer(request, question_id):
    question = get_object_or_404(Question, id=question_id)

    if request.method == "POST":
        form = AnswerForm(request.POST)
        if form.is_valid():
            answer = form.save(commit=False)
            answer.question = question
            answer.save()
            return redirect('add_answer', question.id)
    else:
        form = AnswerForm(initial={'question': question})

    return render(request, 'bustComProj/add_answer.html', {'form': form, 'question': question})

@login_required
def lecture_test(request, lecture_id, test_id):
    lecture = get_object_or_404(Lecture, id=lecture_id)
    test = get_object_or_404(Test, id=test_id, lecture=lecture)
    questions = Question.objects.filter(test=test).prefetch_related('answer_set')

    if request.method == "POST":
        score = 0
        total_questions = questions.count()

        for question in questions:
            selected_answer_id = request.POST.get(f'question_{question.id}')
            if selected_answer_id:
                try:
                    selected_answer = Answer.objects.get(id=selected_answer_id)
                    if selected_answer.is_correct:
                        score += 1
                except Answer.DoesNotExist:
                    pass

        percentage = (score / total_questions) * 100 if total_questions > 0 else 0

        TestAttempt.objects.create(
            user=request.user,
            test=test,
            score=score,
            total_questions=total_questions,
            percentage=percentage
        )

    return render(request, 'bustComProj/test_result.html', {
        'lecture': lecture, 'test': test, 'score': score, 'total': total_questions, 'percentage': percentage
    })

attempts = TestAttempt.objects.filter(user=request.user, test=test).order_by('-attempt_date')
return render(request, 'bustComProj/lecture_test.html', {
    'lecture': lecture, 'test': test, 'questions': questions, 'attempts': attempts
})

@login_required
def courses_list(request):
    courses = Course.objects.all()
    user_courses = Course.objects.filter(enrollment_user=request.user)

    return render(request, 'bustComProj/courses_list.html', {
        'courses': courses,
        'user_courses': user_courses
    })

```

```

    })

@login_required
def course_detail(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    lectures = Lecture.objects.filter(course=course).order_by('order_num')

    is_enrolled = Enrollment.objects.filter(user=request.user, course=course).exists()

    return render(request, 'bustComProj/course_detail.html', {
        'course': course,
        'lectures': lectures,
        'is_enrolled': is_enrolled
    })

@login_required
def course_lectures(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    lectures = Lecture.objects.filter(course=course).order_by('order_num')
    return render(request, 'bustComProj/course_lectures.html', {'course': course, 'lectures': lectures})

@login_required
def lecture_detail(request, lecture_id):
    lecture = get_object_or_404(Lecture, id=lecture_id)
    resources = LectureResource.objects.filter(lecture=lecture)
    comments = Comment.objects.filter(lecture=lecture).order_by('-created_at')

    if request.method == "POST":
        form = CommentForm(request.POST)
        if form.is_valid():
            comment = form.save(commit=False)
            comment.user = request.user
            comment.lecture = lecture
            comment.save()
            return redirect('lecture_detail', lecture_id=lecture.id)
    else:
        form = CommentForm()

    return render(request, 'bustComProj/lecture_detail.html', {
        'lecture': lecture, 'resources': resources, 'comments': comments, 'form': form
    })

@login_required
@role_required('teacher')
def add_lecture(request, course_id):
    course = get_object_or_404(Course, id=course_id)

    if request.method == 'POST':
        form = LectureForm(request.POST)
        if form.is_valid():
            lecture = form.save(commit=False)
            lecture.course = course
            lecture.save()
            return redirect('course_detail', course_id=course.id)
    else:
        form = LectureForm()

    return render(request, 'bustComProj/add_lecture.html', {'form': form, 'course': course})

@login_required
@role_required('teacher')
def edit_lecture(request, lecture_id):
    lecture = get_object_or_404(Lecture, id=lecture_id)

```

```

if request.method == 'POST':
    form = LectureForm(request.POST, instance=lecture)
    if form.is_valid():
        form.save()
        return redirect('lecture_detail', lecture.id)
    else:
        form = LectureForm(instance=lecture)

    return render(request, 'bustComProj/edit_lecture.html', {'form': form, 'lecture': lecture})

@login_required
@role_required('teacher')
def delete_lecture(request, lecture_id):
    lecture = get_object_or_404(Lecture, id=lecture_id)

    if request.method == 'POST':
        course_id = lecture.course.id
        lecture.delete()
        messages.success(request, "Лекция успешно удалена.")
        return redirect('course_lectures', course_id=course_id)

    return render(request, 'bustComProj/delete_lecture.html', {'lecture': lecture})

@login_required
def profile(request):
    user = request.user
    enrollments = Enrollment.objects.filter(user=user).select_related('course')

    progress_data = []
    for enrollment in enrollments:
        course = enrollment.course
        total_lectures = Lecture.objects.filter(course=course).count()
        completed_lectures = CompletedLecture.objects.filter(user=user, lecture_course=course).count()
        progress = int((completed_lectures / total_lectures) * 100) if total_lectures > 0 else 0

        progress_data.append({
            'course': course.title,
            'course_id': course.id,
            'progress': progress,
            'total_lectures': total_lectures,
            'completed_lectures': completed_lectures
        })

    return render(request, 'bustComProj/profile.html', {
        'progress_data': progress_data
    })

@login_required
def edit_profile(request):
    profile, created = Profile.objects.get_or_create(user=request.user)

    if request.method == 'POST':
        form = ProfileForm(request.POST, instance=profile)
        if form.is_valid():
            form.save()
            return redirect('profile')
    else:
        form = ProfileForm(instance=profile)

    return render(request, 'bustComProj/edit_profile.html', {'form': form})

def home(request):

```

```

if request.user.is_authenticated:
    role = request.user.profile.role
    if role == 'hr':
        return redirect('hr_dashboard')
    elif role == 'training_manager':
        return redirect('training_manager_dashboard')
    elif role == 'teacher':
        return redirect('teacher_dashboard')
    elif role == 'employee':
        return redirect('employee_dashboard')
    elif role == 'admin':
        return redirect('admin_dashboard')
return render(request, 'bustComProj/home.html')

def register(request):
    if request.method == 'POST':
        form = CustomUserCreationForm(request.POST)
        if form.is_valid():
            form.save()
            messages.success(request, 'Регистрация успешна! Теперь войдите в систему.')
            return redirect('login')
    else:
        form = CustomUserCreationForm()
    return render(request, 'bustComProj/register.html', {'form': form})

def login_view(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username, password=password)

        if user:
            login(request, user)
            user_role = getattr(user, 'role', 'employee').lower()
            return redirect(ROLE_DASHBOARD.get(user_role, 'employee_dashboard'))
        else:
            messages.error(request, 'Неверные данные!')
    return render(request, 'bustComProj/login.html')

def logout_view(request):
    logout(request)
    return redirect('home')

@login_required
def enroll_course(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    enrollment, created = Enrollment.objects.get_or_create(user=request.user, course=course)

    if created:
        messages.success(request, f"Вы успешно записались на курс: {course.name}")
    else:
        messages.info(request, f"Вы уже записаны на этот курс!")

    return redirect('course_detail', course_id=course.id)

@login_required
def mark_lecture_complete(request, lecture_id):
    lecture = get_object_or_404(Lecture, id=lecture_id)
    CompletedLecture.objects.get_or_create(user=request.user, lecture=lecture)
    check_course_completion(request.user, lecture.course)
    return redirect('lecture_detail', lecture_id=lecture.id)

```

```

def check_course_completion(user, course):
    total_lectures = Lecture.objects.filter(course=course).count()
    completed_lectures = CompletedLecture.objects.filter(user=user, lecture_course=course).count()

    if total_lectures > 0 and total_lectures == completed_lectures:
        Certificate.objects.get_or_create(user=user, course=course)

@login_required
def dashboard_view(request):
    role = request.user.profile.role
    role_to_template = {
        "admin": "dashboards/admin_dashboard.html",
        "teacher": "dashboards/teacher_dashboard.html",
        "hr": "dashboards/hr_dashboard.html",
        "training_manager": "dashboards/training_manager_dashboard.html",
        "employee": "dashboards/employee_dashboard.html",
    }
    template = role_to_template.get(role, "dashboards/employee_dashboard.html")
    return render(request, template)

@login_required
@role_required('employee')
def employee_dashboard(request):
    return render(request, 'dashboards/employee_dashboard.html')

@login_required
@role_required('teacher')
def teacher_dashboard(request):
    return render(request, 'dashboards/teacher_dashboard.html')

@login_required
def hr_dashboard(request):
    if request.user.profile.role != 'hr':
        return redirect('home')
    return render(request, 'hr/dashboard.html')

@login_required
def training_manager_dashboard(request):
    if request.user.profile.role != 'training_manager':
        return redirect('home')
    return render(request, 'training_manager/dashboard.html')

@login_required
@role_required('employee')
def employee_dashboard(request):
    user_courses = Course.objects.filter(enrollment_user=request.user)
    return render(request, 'dashboards/employee_dashboard.html', {
        'courses': user_courses
    })

@login_required
@roles_required('teacher', 'admin')
def create_course(request):
    if request.method == 'POST':
        form = CourseForm(request.POST)
        if form.is_valid():
            course = form.save(commit=False)
            course.teacher = request.user
            course.save()
            messages.success(request, 'Курс успешно создан!')
            return redirect('course_detail', course_id=course.id)
    else:
        form = CourseForm()

```

```

return render(request, 'courses/create_course.html', {
    'form': form,
    'title': 'Создать новый курс'
})

@login_required
@roles_required('teacher', 'admin')
def edit_course(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    if request.method == 'POST':
        form = CourseForm(request.POST, instance=course)
        if form.is_valid():
            form.save()
            messages.success(request, 'Курс обновлён!')
            return redirect('course_detail', course_id=course.id)
    else:
        form = CourseForm(instance=course)
    return render(request, 'courses/edit_course.html', {'form': form, 'course': course})

@login_required
@roles_required('teacher', 'admin')
def delete_course(request, course_id):
    course = get_object_or_404(Course, id=course_id)
    course.delete()
    messages.success(request, 'Курс удалён!')
    return redirect('courses_list')

@login_required
def test_difficulty(request):
    if request.user.profile.role != 'training_manager':
        return redirect('home')
    tests = Test.objects.all()
    return render(request, "training/test_difficulty.html", {"tests": tests})

@login_required
def review_requests(request):
    if request.user.profile.role != 'training_manager':
        return redirect('home')
    if request.method == 'POST':
        request_id = request.POST.get('request_id')
        action = request.POST.get('action')

        training_request = TrainingRequest.objects.get(id=request_id)

        if action == 'approve':
            training_request.status = 'approved'
        elif action == 'reject':
            training_request.status = 'rejected'

        training_request.save()

        messages.success(request, f"Заявка {training_request.user.username} на
{training_request.course.name} обновлена!")

    requests = TrainingRequest.objects.all()
    return render(request, 'training/review_requests.html', {'requests': requests})

@login_required
def create_training_plan(request):
    if request.user.profile.role != 'training_manager':
        return redirect('home')

```

```

if request.method == 'POST':
    action = request.POST.get('action')

    if action == 'delete':
        plan_id = request.POST.get('plan_id')
        try:
            plan = TrainingPlan.objects.get(id=plan_id)
            plan.delete()
            messages.success(request, 'План обучения успешно удален')
        except TrainingPlan.DoesNotExist:
            messages.error(request, 'План обучения не найден')
    else:
        title = request.POST.get('title')
        description = request.POST.get('description')
        start_date = request.POST.get('start_date')
        end_date = request.POST.get('end_date')
        courses = request.POST.getlist('courses')
        employees = request.POST.getlist('employees')

        try:
            plan = TrainingPlan.objects.create(
                title=title,
                description=description,
                start_date=start_date,
                end_date=end_date,
                created_by=request.user
            )

            plan.courses.set(courses)
            plan.employees.set(employees)

            messages.success(request, 'План обучения успешно создан')
        except Exception as e:
            messages.error(request, f'Ошибка при создании плана: {str(e)}')

courses = Course.objects.all()
employees = get_user_model().objects.filter(profile_role='employee')
training_plans = TrainingPlan.objects.all().order_by('-created_at')

return render(request, 'training_manager/create_training_plan.html', {
    'courses': courses,
    'employees': employees,
    'training_plans': training_plans
})

def conduct_training(request):
    return render(request, 'training/conduct_training.html')

@login_required
def dashboard_redirect(request):
    role_dashboard = {
        'admin': 'admin_dashboard',
        'teacher': 'teacher_dashboard',
        'hr': 'hr_dashboard',
        'training_manager': 'training_manager_dashboard',
        'employee': 'employee_dashboard',
        'employee': 'employee_dashboard',
    }

    user_role = getattr(request.user, 'role', 'employee').lower()

    return redirect(role_dashboard.get(user_role, 'employee_dashboard'))

```

```

@login_required
@role_required('teacher')
def answer_questions(request):
    questions = Question.objects.all()

    if request.method == 'POST':
        question_id = request.POST.get('question_id')
        answer_text = request.POST.get('answer_text')

        question = get_object_or_404(Question, id=question_id)
        Answer.objects.create(question=question, answer_text=answer_text, is_correct=False)

        messages.success(request, 'Ответ отправлен!')
        return redirect('answer_questions')

    return render(request, 'questions/answer_questions.html', {'questions': questions})

@login_required
@role_required('teacher')
def view_feedback(request):
    feedbacks = Comment.objects.all()
    return render(request, 'teacher/view_feedback.html', {'feedbacks': feedbacks})

@login_required
def track_performance(request):
    if request.user.profile.role != 'hr':
        return redirect('home')

    avg_score = TestAttempt.objects.aggregate(Avg('score'))['score__avg'] or 0
    max_score = TestAttempt.objects.aggregate(Max('score'))['score__max'] or 0
    min_score = TestAttempt.objects.aggregate(Min('score'))['score__min'] or 0

    score_distribution = (
        TestAttempt.objects.values('score')
        .annotate(count=Count('score'))
        .order_by('score')
    )

    scores = [entry['score'] for entry in score_distribution]
    counts = [entry['count'] for entry in score_distribution]

    context = {
        'avg_score': round(avg_score, 2),
        'max_score': max_score,
        'min_score': min_score,
        'scores': scores,
        'counts': counts
    }
    return render(request, 'bustComProj/track_performance.html', context)

@login_required
@role_required('hr')
def evaluate_training(request):
    return render(request, 'hr/evaluate_training.html')

@login_required
@role_required('hr')
def training_evaluation(request):
    return render(request, 'hr/training_evaluation.html')

@login_required
@roles_required('training_manager')
def employee_progress(request):

```

```

employees = User.objects.filter(profile__role='employee')
progress_data = []

for employee in employees:
    enrollments = Enrollment.objects.filter(user=employee)
    data = []
    for enrollment in enrollments:
        total = Lecture.objects.filter(course=enrollment.course).count()
        completed = CompletedLecture.objects.filter(user=employee,
lecture_course=enrollment.course).count()
        progress = (completed / total) * 100 if total > 0 else 0
        data.append({'course': enrollment.course, 'progress': progress})

    progress_data.append({'employee': employee, 'courses': data})

return render(request, 'training_manager/progress.html', {'progress_data': progress_data})

def service(request):
    return render(request, 'service.html')

def manage_users(request):
    return render(request, 'manage_users.html')

@login_required
def start_training(request):
    return render(request, 'training/start_training.html')

@login_required
def my_certificates(request):
    certificates = Certificate.objects.filter(user=request.user).select_related('course')
    return render(request, 'certificates/my_certificates.html', {'certificates': certificates})

@login_required
@role_required('hr')
def view_certificates(request):
    certificates = Certificate.objects.all().select_related('user', 'course')
    return render(request, 'certificates/view_certificates.html', {'certificates': certificates})

@login_required
def download_certificate(request, certificate_id):
    certificate = get_object_or_404(Certificate, id=certificate_id, user=request.user)
    return HttpResponseRedirect("Функция скачивания сертификата в разработке")

@login_required
def view_certificate(request, certificate_id):
    certificate = get_object_or_404(Certificate, id=certificate_id, user=request.user)
    return render(request, 'certificates/view_certificate.html', {'certificate': certificate})

@login_required
@role_required('teacher')
def add_lecture_material(request):
    if request.method == 'POST':
        lecture_id = request.POST.get('lecture')
        resource_type = request.POST.get('resource_type')
        resource_path = request.POST.get('resource_path')

        if not all([lecture_id, resource_type, resource_path]):
            messages.error(request, 'Пожалуйста, заполните все поля')
            return redirect('add_lecture_material')

    try:
        lecture = Lecture.objects.get(id=lecture_id)
        if lecture.course.teacher != request.user:

```

```

messages.error(request, 'У вас нет прав для добавления материалов к этой лекции')
return redirect('add_lecture_material')

LectureResource.objects.create(
    lecture=lecture,
    resource_type=resource_type,
    resource_path=resource_path
)
messages.success(request, 'Материал успешно добавлен')
return redirect('lecture_detail', lecture_id=lecture.id)

except Lecture.DoesNotExist:
    messages.error(request, 'Лекция не найдена')
    return redirect('add_lecture_material')
else:
    courses = Course.objects.filter(teacher=request.user)
    lectures = Lecture.objects.filter(course__in=courses).order_by('course__title', 'order_num')

    return render(request, 'bustComProj/courses/add_lecture_material.html', {
        'lectures': lectures,
        'title': 'Добавить материал к лекции'
    })

@login_required
def assign_courses(request):
    if request.user.profile.role != 'hr':
        return redirect('home')

    if request.method == 'POST':
        action = request.POST.get('action')

        if action == 'delete':
            enrollment_id = request.POST.get('enrollment_id')
            try:
                enrollment = Enrollment.objects.get(id=enrollment_id)
                enrollment.delete()
                messages.success(request, 'Назначение успешно удалено')
            except Enrollment.DoesNotExist:
                messages.error(request, 'Назначение не найдено')
        else:
            employee_id = request.POST.get('employee')
            course_id = request.POST.get('course')

            try:
                employee = get_user_model().objects.get(id=employee_id)
                course = Course.objects.get(id=course_id)

                if not Enrollment.objects.filter(user=employee, course=course).exists():
                    Enrollment.objects.create(user=employee, course=course)
                    messages.success(request, f'Курс "{course.title}" успешно назначен сотруднику {employee.get_full_name()}')
                else:
                    messages.warning(request, 'Этот курс уже назначен данному сотруднику')
            except (get_user_model().DoesNotExist, Course.DoesNotExist):
                messages.error(request, 'Сотрудник или курс не найден')

    employees = get_user_model().objects.filter(profile_role='employee')
    courses = Course.objects.all()
    enrollments = Enrollment.objects.select_related('user', 'course').all()

    return render(request, 'hr/assign_courses.html', {
        'employees': employees,
        'courses': courses,
    })

```

```

        'enrollments': enrollments
    })

@login_required
def review_requests(request):
    if request.user.profile.role != 'training_manager':
        return redirect('home')
    return render(request, 'training_manager/review_requests.html')

@login_required
def test_difficulty(request):
    if request.user.profile.role != 'training_manager':
        return redirect('home')
    return render(request, 'training_manager/test_difficulty.html')

@login_required
@role_required('admin')
def download_backup(request):
    """Создает и отправляет резервную копию базы данных"""
    db_path = os.path.join(os.path.dirname(os.path.dirname(__file__)), 'db.sqlite3')

    timestamp = datetime.datetime.now().strftime("%Y%m%d_%H%M%S")
    temp_dir = tempfile.mkdtemp()
    backup_path = os.path.join(temp_dir, f'backup_{timestamp}.sqlite3')

    try:
        shutil.copy2(db_path, backup_path)

        response = FileResponse(
            open(backup_path, 'rb'),
            as_attachment=True,
            filename=f'backup_{timestamp}.sqlite3'
        )
        return response
    finally:
        try:
            os.remove(backup_path)
            os.rmdir(temp_dir)
        except:
            pass

class RestoreBackupForm(Form):
    """Форма для загрузки файла бэкапа"""
    backup_file = FileField(label='Файл бэкапа')

@login_required
@role_required('admin')
def restore_backup(request):
    """Восстанавливает базу данных из загруженного бэкапа"""
    if request.method == 'POST':
        form = RestoreBackupForm(request.POST, request.FILES)
        if form.is_valid():
            backup_file = request.FILES['backup_file']

            if not backup_file.name.endswith('.sqlite3'):
                messages.error(request, 'Неверный формат файла. Пожалуйста, загрузите файл .sqlite3')
                return redirect('admin_dashboard')

            temp_dir = tempfile.mkdtemp()
            temp_path = os.path.join(temp_dir, 'temp_backup.sqlite3')

            try:
                with open(temp_path, 'wb+') as destination:

```

```

        for chunk in backup_file.chunks():
            destination.write(chunk)

    try:
        conn = sqlite3.connect(temp_path)
        conn.close()
    except sqlite3.Error:
        messages.error(request, 'Загруженный файл не является корректной базой данных SQLite')
        return redirect('admin_dashboard')

    db_path = os.path.join(os.path.dirname(os.path.dirname(__file__)), 'db.sqlite3')

    backup_timestamp = datetime.datetime.now().strftime("%Y%m%d_%H%M%S")
    current_backup = f'{db_path}.backup_{backup_timestamp}'
    shutil.copy2(db_path, current_backup)

    shutil.copy2(temp_path, db_path)

    messages.success(request, 'База данных успешно восстановлена из бэкапа')

except Exception as e:
    messages.error(request, f'Ошибка при восстановлении базы данных: {str(e)}')
finally:
    try:
        os.remove(temp_path)
        os.rmdir(temp_dir)
    except:
        pass

    return redirect('admin_dashboard')
else:
    form = RestoreBackupForm()

return render(request, 'admin_panel/restore_backup.html', {'form': form})

```

urls.py

```

from django.urls import path
from django.contrib import admin
from bustComProj.views import *

urlpatterns = [
    path("", home, name='home'),
    path('courses/', courses_list, name='courses_list'),
    path('course/create/', create_course, name='add_course'),
    path('course/<int:course_id>', course_detail, name='course_detail'),
    path('course/<int:course_id>/enroll/', enroll_course, name='enroll_course'),
    path('course/<int:course_id>/lectures/', course_lectures, name='course_lectures'),
    path('course/<int:course_id>/edit/', edit_course, name='edit_course'),
    path('course/<int:course_id>/delete/', delete_course, name='delete_course'),

    path('lecture/<int:lecture_id>', lecture_detail, name='lecture_detail'),
    path('lecture/<int:lecture_id>/test/<int:test_id>', lecture_test, name='lecture_test'),

    path('lecture/<int:lecture_id>/complete/', mark_lecture_complete, name='mark_lecture_complete'),

    path('register/', register, name='register'),
    path('login/', login_view, name='login'),
    path('logout/', logout_view, name='logout'),
    path('course/<int:course_id>/add-lecture/', add_lecture, name='add_lecture'),

    path('admin/', admin.site.urls),
]

```

```

path('admin-panel/', admin_dashboard, name='admin_dashboard'),
path('admin-panel/backup/download/', download_backup, name='download_backup'),
path('admin-panel/backup/restore/', restore_backup, name='restore_backup'),

path('profile/', profile, name='profile'),
path('profile/edit/', edit_profile, name='edit_profile'),

path('training/test-difficulty/', test_difficulty, name='test_difficulty'),
path('training/review-requests/', review_requests, name='review_requests'),
path('training/create-plan/', create_training_plan, name='create_training_plan'),
path('training/conduct/', conduct_training, name='conduct_training'),

path('dashboard/', dashboard_redirect, name='dashboard_redirect'),
path('dashboard/employee/', employee_dashboard, name='employee_dashboard'),
path('dashboard/employee/', employee_dashboard, name='employee_dashboard'),
path('dashboard/teacher/', teacher_dashboard, name='teacher_dashboard'),
path('dashboard/hr/', hr_dashboard, name='hr_dashboard'),
path('dashboard/training-manager/', training_manager_dashboard, name='training_manager_dashboard'),

path('questions/answer/', answer_questions, name='answer_questions'),
path('teacher/feedback/', view_feedback, name='view_feedback'),

path('hr/assign-courses/', assign_courses, name='assign_courses'),
path('hr/track-performance/', track_performance, name='track_performance'),
path('hr/evaluate-training/', evaluate_training, name='evaluate_training'),
path('hr/training-evaluation/', training_evaluation, name='training_evaluation'),

path('training-manager/employee-progress/', employee_progress, name='employee_progress'),

path('service/', service, name='service'),

path('lecture/<int:lecture_id>/add-test', add_test, name='add_test'),
path('lecture/<int:lecture_id>/edit', edit_lecture, name='edit_lecture'),
path('lecture/<int:lecture_id>/delete', delete_lecture, name='delete_lecture'),

path('manage/<str:table_name>', manage_table, name='manage_table'),
path('manage/<str:table_name>/add', add_object, name='add_object'),
path('manage/<str:table_name>/edit/<int:object_id>', edit_object, name='edit_object'),
path('manage/<str:table_name>/delete/<int:object_id>', delete_object, name='delete_object'),

path('test/<int:test_id>/add-question', add_question, name='add_question'),
path('question/<int:question_id>/add-answer', add_answer, name='add_answer'),
path('manage-users/', manage_users, name='manage_users'),
path('training/start/', start_training, name='start_training'),
path('my-certificates/', my_certificates, name='my_certificates'),
path('view-certificates/', view_certificates, name='view_certificates'),
path('certificate/<int:certificate_id>/view', view_certificate, name='view_certificate'),
path('certificate/<int:certificate_id>/download', download_certificate, name='download_certificate'),

path('add-lecture-material/', add_lecture_material, name='add_lecture_material'),
]

```

models.py

```

from django.db import models
from django.conf import settings
from django.contrib.auth.models import AbstractUser, BaseUserManager
from django.db import models

class Role(models.Model):
    name = models.CharField(max_length=50, unique=True)

```

```

def __str__(self):
    return self.name

class UserManager(BaseUserManager):
    def create_user(self, username, email, password=None, **extra_fields):
        if not email:
            raise ValueError("Email обязательен!")
        email = self.normalize_email(email)
        user = self.model(username=username, email=email, **extra_fields)
        user.set_password(password)
        user.save(using=self._db)
        return user

    def create_superuser(self, username, email, password=None, **extra_fields):
        extra_fields.setdefault('is_staff', True)
        extra_fields.setdefault('is_superuser', True)

        return self.create_user(username, email, password, **extra_fields)

class User(AbstractUser):
    ROLE_CHOICES = [
        ('admin', 'Администратор'),
        ('hr', 'HR-менеджер'),
        ('teacher', 'Преподаватель'),
        ('training_manager', 'Менеджер по обучению'),
        ('employee', 'Сотрудник'),
    ]

    email = models.EmailField(unique=True)
    role = models.CharField(max_length=20, choices=ROLE_CHOICES, default='employee')

    objects = UserManager()

    groups = models.ManyToManyField(
        'auth.Group',
        related_name="bustcomproj_user_groups",
        blank=True
    )
    user_permissions = models.ManyToManyField(
        'auth.Permission',
        related_name="bustcomproj_user_permissions",
        blank=True
    )

    def is_hr(self):
        return self.role == 'hr'

    def is_teacher(self):
        return self.role == 'teacher'

    def is_training_manager(self):
        return self.role == 'training_manager'

    def is_employee(self):
        return self.role == 'employee'

    def __str__(self):
        return f'{self.username} - {self.get_role_display()}'
```

```

class Course(models.Model):
    LEVEL_CHOICES = [
        ('beginner', 'Начальный'),
        ('intermediate', 'Средний'),
        ('advanced', 'Продвинутый')
    ]

    title = models.CharField(max_length=100)
    description = models.TextField()
    duration = models.IntegerField(help_text='Продолжительность в часах')
    level = models.CharField(max_length=20, choices=LEVEL_CHOICES, default='beginner')
    teacher = models.ForeignKey(User, on_delete=models.CASCADE, related_name='courses')
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.title


class CourseTeacher(models.Model):
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    teacher = models.ForeignKey(User, on_delete=models.CASCADE)

    class Meta:
        unique_together = ('course', 'teacher')


class Lecture(models.Model):
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    title = models.CharField(max_length=100)
    order_num = models.IntegerField()

    def __str__(self):
        return self.title


class LectureResource(models.Model):
    lecture = models.ForeignKey(Lecture, on_delete=models.CASCADE)
    resource_type = models.CharField(max_length=10, choices=[('file', 'Файл'), ('video', 'Видео')])
    resource_path = models.CharField(max_length=255)

    def __str__(self):
        return f'{self.lecture.title} - {self.resource_type}'


class Enrollment(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    enrolled_at = models.DateTimeField(auto_now_add=True)
    status_choices = [('в процессе', 'В процессе'), ('завершён', 'Завершён'), ('отчислен', 'Отчислен')]
    status = models.CharField(max_length=15, choices=status_choices, default='в процессе')

    class Meta:
        unique_together = ('user', 'course')


class Progress(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    lecture = models.ForeignKey(Lecture, on_delete=models.CASCADE)
    completed_at = models.DateTimeField(null=True, blank=True)

    class Meta:
        unique_together = ('user', 'course', 'lecture')


class Test(models.Model):
    lecture = models.ForeignKey(Lecture, on_delete=models.CASCADE)
    name = models.CharField(max_length=100)

```

```

difficulty_choices = [('низкая', 'Низкая'), ('средняя', 'Средняя'), ('высокая', 'Высокая')]
difficulty = models.CharField(max_length=10, choices=difficulty_choices)
attempts = models.IntegerField(default=3)

def __str__(self):
    return self.name


class Question(models.Model):
    test = models.ForeignKey(Test, on_delete=models.CASCADE)
    question_text = models.TextField()
    question_type_choices = [('один', 'Один ответ'), ('несколько', 'Несколько ответов'), ('ввод', 'Ввод текста')]
    question_type = models.CharField(max_length=10, choices=question_type_choices)


class Answer(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    answer_text = models.TextField()
    is_correct = models.BooleanField()


class Attempt(models.Model):
    test = models.ForeignKey(Test, on_delete=models.CASCADE)
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    attempt_num = models.IntegerField()
    score = models.IntegerField()
    passed = models.BooleanField()
    completed_at = models.DateTimeField(auto_now_add=True)


class Certificate(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    issued_at = models.DateTimeField(auto_now_add=True)
    grade = models.CharField(max_length=10)


class Notification(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    message = models.TextField()
    status_choices = [('новое', 'Новое'), ('прочитано', 'Прочитано')]
    status = models.CharField(max_length=10, choices=status_choices, default='новое')
    created_at = models.DateTimeField(auto_now_add=True)


class CourseLog(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    action_choices = [('запись', 'Запись'), ('прогресс', 'Прогресс'), ('завершение', 'Завершение'), ('сертификат', 'Сертификат')]
    action = models.CharField(max_length=15, choices=action_choices)
    related_id = models.IntegerField(null=True, blank=True)
    log_time = models.DateTimeField(auto_now_add=True)


class CompletedLecture(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    lecture = models.ForeignKey(Lecture, on_delete=models.CASCADE)
    completed_at = models.DateTimeField(auto_now_add=True)


class Meta:
    unique_together = ('user', 'lecture')

def __str__(self):
    return f'{self.user.username} - {self.lecture.title}'

```

```

class Profile(models.Model):
    ROLE_CHOICES = [
        ('admin', 'Администратор'),
        ('hr', 'HR-менеджер'),
        ('teacher', 'Преподаватель'),
        ('training_manager', 'Менеджер по обучению'),
        ('employee', 'Сотрудник'),
    ]
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    role = models.CharField(max_length=20, choices=ROLE_CHOICES, default='employee')
    phone = models.CharField(max_length=20, blank=True, null=True)
    company = models.CharField(max_length=100, blank=True, null=True)

    def __str__(self):
        return f'{self.user.username} - {self.get_role_display()}'


from django.db import models
from django.contrib.auth.models import User

from django.conf import settings

class TestAttempt(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    test = models.ForeignKey(Test, on_delete=models.CASCADE)
    score = models.IntegerField()
    total_questions = models.IntegerField()
    percentage = models.FloatField()
    attempt_date = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f'{self.user.username} - {self.test.lecture.title} - {self.percentage}%'


class Comment(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    lecture = models.ForeignKey(Lecture, on_delete=models.CASCADE, related_name='comments')
    text = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f'{self.user.username} - {self.lecture.title} - {self.created_at}'


from django.db import models
from django.conf import settings

class TrainingRequest(models.Model):
    STATUS_CHOICES = [
        ('pending', 'Ожидание'),
        ('approved', 'Одобрено'),
        ('rejected', 'Отклонено')
    ]
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    course = models.ForeignKey('Course', on_delete=models.CASCADE)
    reason = models.TextField()
    status = models.CharField(max_length=10, choices=STATUS_CHOICES, default='pending')
    created_at = models.DateTimeField(auto_now_add=True)

```

```

def __str__(self):
    return f'{self.user.username} - {self.course.title} ({self.get_status_display()})'

from django.db.models.signals import post_save
from django.dispatch import receiver
from .models import Profile, User

@receiver(post_save, sender=User)
def create_user_profile(sender, instance, created, **kwargs):
    if created:
        Profile.objects.get_or_create(user=instance, defaults={'role': instance.role})

class TrainingPlan(models.Model):
    STATUS_CHOICES = [
        ('draft', 'Черновик'),
        ('active', 'Активный'),
        ('completed', 'Завершен'),
        ('cancelled', 'Отменен')
    ]

    title = models.CharField(max_length=200)
    description = models.TextField()
    start_date = models.DateField()
    end_date = models.DateField()
    status = models.CharField(max_length=20, choices=STATUS_CHOICES, default='draft')
    created_by = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    courses = models.ManyToManyField('Course', related_name='training_plans')
    employees = models.ManyToManyField(settings.AUTH_USER_MODEL,
                                       related_name='assigned_training_plans')

    def __str__(self):
        return self.title

```

admin.py

```

from django.contrib import admin
from .models import (
    Role, User, Course, CourseTeacher, Lecture, LectureResource,
    Enrollment, Progress, Test, Question, Answer, Attempt,
    Certificate, Notification, CourseLog, TrainingRequest, TrainingPlan
)

admin.site.register(Role)

@admin.register(User)
class UserAdmin(admin.ModelAdmin):
    list_display = ('username', 'email', 'role', 'is_staff', 'is_active')
    search_fields = ('username', 'email')
    list_filter = ('role', 'is_staff', 'is_active')

@admin.register(Course)
class CourseAdmin(admin.ModelAdmin):
    list_display = ('title', 'teacher', 'duration', 'level', 'created_at')
    list_filter = ('level', 'teacher')
    search_fields = ('title', 'description')
    ordering = ('-created_at',)

@admin.register(Lecture)
class LectureAdmin(admin.ModelAdmin):

```

```

list_display = ('title', 'course', 'order_num')
search_fields = ('title',)
list_filter = ('course',)

admin.site.register([
    LectureResource, CourseTeacher, Enrollment, Progress, Test,
    Question, Answer, Attempt, Certificate, Notification,
    CourseLog, TrainingRequest, TrainingPlan
])

```

forms.py

```

from django import forms
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth import get_user_model
from .models import User, Role, Profile, Course, Comment, Lecture, Test, Question, Answer
from django.forms.widgets import DateInput
User = get_user_model()

class CustomUserCreationForm(UserCreationForm):
    role = forms.ModelChoiceField(queryset=Role.objects.all(), required=True, label="Роль")

    class Meta:
        model = User
        fields = ['username', 'email', 'password1', 'password2', 'role']

    def save(self, commit=True):
        user = super().save(commit=False)
        user.role = self.cleaned_data['role']

        if commit:
            user.save()
            Profile.objects.get_or_create(user=user, defaults={'role': user.role.name})
        return user

class CourseForm(forms.ModelForm):
    class Meta:
        model = Course
        fields = ['title', 'description', 'duration', 'level']
        widgets = {
            'title': forms.TextInput(attrs={
                'class': 'form-control',
                'placeholder': 'Введите название курса'
            }),
            'description': forms.Textarea(attrs={
                'class': 'form-control',
                'placeholder': 'Введите описание курса',
                'rows': 4
            }),
            'duration': forms.NumberInput(attrs={
                'class': 'form-control',
                'placeholder': 'Продолжительность в часах'
            }),
            'level': forms.Select(attrs={
                'class': 'form-control'
            })
        }
        labels = {
            'title': 'Название курса',
            'description': 'Описание',
            'duration': 'Продолжительность (часов)',
            'level': 'Уровень сложности'
        }

```

```

        }
    help_texts = {
        'duration': 'Укажите примерную продолжительность курса в часах',
        'level': 'Выберите уровень сложности курса'
    }

class ProfileForm(forms.ModelForm):
    class Meta:
        model = Profile
        fields = ['role', 'phone', 'company']

class CommentForm(forms.ModelForm):
    class Meta:
        model = Comment
        fields = ['text']


class LectureForm(forms.ModelForm):
    class Meta:
        model = Lecture
        fields = ['title', 'order_num']

class TestForm(forms.ModelForm):
    class Meta:
        model = Test
        fields = ['name', 'difficulty', 'attempts']


class QuestionForm(forms.ModelForm):
    class Meta:
        model = Question
        fields = ['test', 'question_text', 'question_type']

class AnswerForm(forms.ModelForm):
    class Meta:
        model = Answer
        fields = ['question', 'answer_text', 'is_correct']

```

apps.py

```

from django.apps import AppConfig

class BustComProjConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'bustComProj'

    def ready(self):
        import bustComProj.signals

```

signals.py

```

from django.db.models.signals import post_save
from django.dispatch import receiver
from .models import User, Profile

@receiver(post_save, sender=User)
def create_user_profile(sender, instance, created, **kwargs):
    if created:

```

```
Profile.objects.create(user=instance)
```

decorators.py

```
from django.contrib.auth.decorators import user_passes_test

def role_required(role):
    """Декоратор для проверки роли пользователя"""
    def check_role(user):
        return user.is_authenticated and getattr(user.profile, 'role', None) == role
    return user_passes_test(check_role)
```

tests.py

```
from django.test import TestCase

# Create your tests here.
```

select_test.html

```
{% extends "base.html" %}

{% block content %}
<h2>Выберите тест для лекции "{{ lecture.title }}"</h2>
<ul>
    {% for test in tests %}
        <li><a href="{% url 'lecture_test' lecture.id test.id %}">{{ test.name }}</a></li>
    {% endfor %}
</ul>
{% endblock %}
```

add_question.html

```
{% extends "base.html" %}

{% block content %}
<h2>Добавить вопрос к тесту "{{ test.name }}"</h2>
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Добавить</button>
</form>
{% endblock %}
```

service.html

```
{% extends "bustComProj/base.html" %}

{% block title %}Наш сервис{% endblock %}

{% block content %}
<h2>⚙️ Наша услуга</h2>

<div class="services">
    <div class="service-card">
        <h3>💻 Онлайн-курсы</h3>
```

```

<p>Обучение по различным направлениям с сертификацией.</p>
</div>

<div class="service-card">
  <h3>sertifikaty Сертификаты</h3>
  <p>Получите сертификат после успешного завершения курсов.</p>
</div>

<div class="service-card">
  <h3>podderzhka Поддержка</h3>
  <p>Общение с преподавателями и ответы на вопросы.</p>
</div>

<div class="service-card">
  <h3>progress Прогресс</h3>
  <p>Отслеживайте свой прогресс и статистику обучения.</p>
</div>
</div>

<a class="btn" href="{% url 'courses_list' %}">Перейти к курсам</a>

{% endblock %}

```

admin_dashboard.html

```

{% extends "bustComProj/base.html" %}

{% block title %}Панель администратора{% endblock %}

{% block content %}
  <h1>Панель администратора</h1>
  <ul>
    <li><a href="{% url 'manage_users' %}">👤 Управление пользователями</a></li>
    <li><a href="{% url 'upload_courses' %}">📁 Загрузка курсов</a></li>
    <li><a href="{% url 'monitor_activity' %}">📊 Мониторинг активности</a></li>
  </ul>

  <h2>📊 Статистика</h2>
  <canvas id="dashboardChart"></canvas>

  <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
  <script>
    fetch("{% url 'get_dashboard_data' %}")
      .then(response => response.json())
      .then(data => {
        var ctx = document.getElementById('dashboardChart').getContext('2d');
        new Chart(ctx, {
          type: 'bar',
          data: {
            labels: ["Курсы", "Тесты", "Пройдено лекций", "Пользователи"],
            datasets: [
              {
                label: "Общая активность",
                data: [data.courses, data.tests, data.lectures_completed, data.users],
                backgroundColor: ['#ff00ff', '#00ffcc', '#ffcc00', '#ff6600'],
              }
            ],
            options: { responsive: true, scales: { y: { beginAtZero: true } } }
          });
      });
  </script>
{% endblock %}

```

dashboard_base.html

```
{% extends 'bustComProj/base.html' %}

{% block title %}Личный кабинет{% endblock %}

{% block content %}
<div class="dashboard-container">
    <aside class="sidebar">
        <nav class="nav-menu">
            <div class="nav-section">
                <h3>Основное меню</h3>
                <ul>
                    <li><a href="{% url 'home' %}"><i class="fas fa-home"></i> Главная</a></li>
                    <li><a href="{% url 'dashboard_redirect' %}"><i class="fas fa-tachometer-alt"></i> Панель
                        управления</a></li>
                    {% block main_menu %}{% endblock %}
                </ul>
            </div>

            <div class="nav-section">
                <h3>Дополнительно</h3>
                <ul>
                    {% block extra_menu %}{% endblock %}
                    <li><a href="{% url 'logout' %}"><i class="fas fa-sign-out-alt"></i> Выход</a></li>
                </ul>
            </div>
        </nav>
    </aside>

    <main class="main-content">
        <header class="dashboard-header">
            <h1>{% block dashboard_title %}Панель управления{% endblock %}</h1>
        </header>

        <div class="dashboard-content">
            {% block dashboard_content %}{% endblock %}
        </div>
    </main>
</div>

<style>
.dashboard-container {
    display: grid;
    grid-template-columns: 250px 1fr;
    min-height: 100vh;
    background: #1a1a1a;
}

.sidebar {
    background: rgba(22, 22, 22, 0.95);
    border-right: 1px solid rgba(0, 255, 255, 0.1);
    padding: 2rem 0;
}

.nav-menu {
    position: sticky;
    top: 2rem;
}

```

```
.nav-section {  
    margin-bottom: 2rem;  
    padding: 0 1.5rem;  
}  
  
.nav-section h3 {  
    color: #00ffff;  
    font-size: 0.9rem;  
    text-transform: uppercase;  
    letter-spacing: 0.1em;  
    margin-bottom: 1rem;  
    padding-bottom: 0.5rem;  
    border-bottom: 1px solid rgba(0, 255, 255, 0.2);  
}  
  
.nav-section ul {  
    list-style: none;  
    padding: 0;  
    margin: 0;  
}  
  
.nav-section li {  
    margin-bottom: 0.5rem;  
}  
  
.nav-section a {  
    display: flex;  
    align-items: center;  
    padding: 0.75rem 1rem;  
    color: rgba(255, 255, 255, 0.8);  
    text-decoration: none;  
    border-radius: 6px;  
    transition: all 0.2s ease;  
}  
  
.nav-section a:hover {  
    background: rgba(0, 255, 255, 0.1);  
    color: #00ffff;  
}  
  
.nav-section a i {  
    margin-right: 0.75rem;  
    font-size: 1.1rem;  
    width: 20px;  
    text-align: center;  
}  
  
.main-content {  
    padding: 2rem;  
}  
  
.dashboard-header {  
    margin-bottom: 2rem;  
}  
  
.dashboard-header h1 {  
    font-size: clamp(1.5rem, 4vw, 2rem);  
    color: #00ffff;  
    margin: 0;  
    text-shadow: 0 0 10px rgba(0, 255, 255, 0.3);  
}  
  
.dashboard-content {
```

```

background: rgba(22, 22, 22, 0.95);
border: 1px solid rgba(0, 255, 255, 0.1);
border-radius: 12px;
padding: 2rem;
}

@media (max-width: 768px) {
  .dashboard-container {
    grid-template-columns: 1fr;
  }

  .sidebar {
    display: none;
  }

  .main-content {
    padding: 1rem;
  }

  .dashboard-content {
    padding: 1rem;
  }
}
</style>
{% endblock %}

```

hr_dashboard.html

```

{% extends "dashboards/dashboard_base.html" %}

{% block title %}Панель HR-менеджера{% endblock %}

{% block extra_links %}
<li><a href="{% url 'assign_courses' %}">> Назначить курсы</a></li>
<li><a href="{% url 'track_performance' %}">> Мониторинг успеваемости</a></li>
<li><a href="{% url 'training_evaluation' %}">> Оценка эффективности</a></li>

```

{% endblock %}

employee_dashboard.html

```

{% extends "dashboards/dashboard_base.html" %}

{% load static %}

{% block title %}Личный кабинет сотрудника{% endblock %}

{% block content %}
<div class="dashboard-container">
  <div class="welcome-section">
    <h1 class="dashboard-title">Личный кабинет</h1>
    <p class="dashboard-subtitle">Добро пожаловать, {{ user.get_full_name|default:user.username }}!</p>
  </div>

  <div class="courses-section">
    <div class="section-header">
      <h2 class="section-title">Мои курсы</h2>

```

```

<a href="{% url 'courses_list' %}" class="btn-action btn-outline">
    <i class="fas fa-search"></i>
    <span>Найти курсы</span>
</a>
</div>

{% if user_courses %}
<div class="courses-grid">
    {% for course in user_courses %}
        <div class="course-card">
            <div class="course-header">
                <div class="course-icon">
                    <i class="fas fa-graduation-cap"></i>
                </div>
                <h3>{{ course.name }}</h3>
            </div>
            <div class="course-info">
                <div class="info-row">
                    <i class="fas fa-clock"></i>
                    <span>{{ course.duration }} часов</span>
                </div>
                <div class="info-row">
                    <i class="fas fa-chalkboard-teacher"></i>
                    <span>{{ course.teacher.get_full_name }}</span>
                </div>
                <div class="course-progress">
                    <div class="progress-header">
                        <span class="progress-label">Прогресс курса</span>
                        <span class="progress-value">{{ course.progress }}%</span>
                    </div>
                    <div class="progress-bar">
                        <div class="progress" style="width: {{ course.progress }}%"></div>
                    </div>
                </div>
            </div>
            <div class="course-actions">
                <a href="{% url 'course_detail' course.id %}" class="btn-action btn-continue">
                    <i class="fas fa-play"></i>
                    <span>Продолжить обучение</span>
                </a>
                <a href="{% url 'course_detail' course.id %}" class="btn-action btn-view">
                    <i class="fas fa-eye"></i>
                    <span>Подробнее</span>
                </a>
            </div>
        {% endfor %}
    </div>
    {% else %}
        <div class="no-courses">
            <div class="no-courses-icon">
                <i class="fas fa-book"></i>
            </div>
            <p>У вас пока нет активных курсов</p>
            <a href="{% url 'courses_list' %}" class="btn-action btn-enroll">
                <i class="fas fa-plus"></i>
                <span>Записаться на курс</span>
            </a>
        </div>
    {% endif %}
</div>
</div>

```

```
<style>
  .dashboard-container {
    padding: clamp(1rem, 3vw, 2rem);
    max-width: 1400px;
    margin: 0 auto;
  }

  .welcome-section {
    text-align: center;
    margin-bottom: clamp(2rem, 5vw, 3rem);
  }

  .dashboard-title {
    font-size: clamp(1.75rem, 5vw, 2.5rem);
    color: #fff;
    margin-bottom: 0.5rem;
    font-weight: 600;
  }

  .dashboard-subtitle {
    color: rgba(255, 255, 255, 0.7);
    font-size: clamp(1rem, 3vw, 1.2rem);
  }

  .section-header {
    display: flex;
    align-items: center;
    justify-content: space-between;
    gap: 1rem;
    margin-bottom: clamp(1.5rem, 4vw, 2rem);
    flex-wrap: wrap;
  }

  .section-title {
    font-size: clamp(1.25rem, 4vw, 1.75rem);
    color: #fff;
    font-weight: 500;
    margin: 0;
  }

  .courses-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
    gap: clamp(0.75rem, 2vw, 1.5rem);
  }

  .course-card {
    background: rgba(22, 22, 22, 0.95);
    border: 1px solid rgba(0, 255, 255, 0.1);
    border-radius: 12px;
    padding: clamp(0.75rem, 2vw, 1.25rem);
    transition: all 0.3s ease;
    height: 100%;
    display: flex;
    flex-direction: column;
  }

  .course-card:hover {
    transform: translateY(-2px);
    box-shadow: 0 8px 30px rgba(0, 255, 255, 0.15);
  }

  .course-header {
```

```
display: flex;
align-items: center;
gap: 0.75rem;
margin-bottom: 1rem;
}

.course-icon {
width: clamp(32px, 5vw, 40px);
height: clamp(32px, 5vw, 40px);
display: flex;
align-items: center;
justify-content: center;
background: linear-gradient(45deg, #00ffff20, #ff00ff20);
border-radius: 8px;
flex-shrink: 0;
}

.course-icon i {
font-size: clamp(0.9rem, 2vw, 1.1rem);
color: #00ffff;
}

.course-header h3 {
color: #fff;
font-size: clamp(0.9rem, 2vw, 1.1rem);
margin: 0;
font-weight: 500;
line-height: 1.3;
}

.course-info {
flex-grow: 1;
margin-bottom: 1rem;
}

.info-row {
display: flex;
align-items: center;
gap: 0.5rem;
padding: 0.25rem 0;
color: rgba(255, 255, 255, 0.7);
font-size: clamp(0.8rem, 1.5vw, 0.9rem);
}

.info-row i {
color: #00ffff;
font-size: 0.9rem;
width: 16px;
text-align: center;
flex-shrink: 0;
}

.course-progress {
margin-top: 1rem;
}

.progress-header {
display: flex;
align-items: center;
justify-content: space-between;
margin-bottom: 0.5rem;
}
```

```
.progress-label {
  color: rgba(255, 255, 255, 0.7);
  font-size: clamp(0.8rem, 1.5vw, 0.9rem);
}

.progress-value {
  color: #00ffff;
  font-size: clamp(0.8rem, 1.5vw, 0.9rem);
  font-weight: 500;
}

.progress-bar {
  height: 6px;
  background: rgba(255, 255, 255, 0.1);
  border-radius: 3px;
  overflow: hidden;
}

.progress {
  height: 100%;
  background: linear-gradient(90deg, #00ffff, #ff00ff);
  transition: width 0.3s ease;
}

.course-actions {
  display: flex;
  flex-direction: column;
  gap: 0.5rem;
  margin-top: auto;
}

.btn-action {
  display: inline-flex;
  align-items: center;
  justify-content: center;
  gap: 0.5rem;
  padding: 0.75rem;
  border-radius: 8px;
  font-size: clamp(0.8rem, 1.5vw, 0.9rem);
  font-weight: 500;
  text-decoration: none;
  transition: all 0.2s ease;
}

.btn-action i {
  font-size: 0.9em;
  flex-shrink: 0;
}

.btn-outline {
  color: #00ffff;
  background: transparent;
  border: 1px solid rgba(0, 255, 255, 0.3);
}

.btn-outline:hover {
  background: rgba(0, 255, 255, 0.1);
  border-color: rgba(0, 255, 255, 0.4);
}

.btn-continue {
  color: #00ff95;
  background: rgba(0, 255, 149, 0.1);
```

```
        border: 1px solid rgba(0, 255, 149, 0.2);
    }

    .btn-continue:hover {
        background: rgba(0, 255, 149, 0.15);
        border-color: rgba(0, 255, 149, 0.3);
    }

    .btn-view {
        color: #ff00ff;
        background: rgba(255, 0, 255, 0.1);
        border: 1px solid rgba(255, 0, 255, 0.2);
    }

    .btn-view:hover {
        background: rgba(255, 0, 255, 0.15);
        border-color: rgba(255, 0, 255, 0.3);
    }

    .btn-enroll {
        color: #00ffff;
        background: rgba(0, 255, 255, 0.1);
        border: 1px solid rgba(0, 255, 255, 0.2);
    }

    .btn-enroll:hover {
        background: rgba(0, 255, 255, 0.15);
        border-color: rgba(0, 255, 255, 0.3);
    }

    .no-courses {
        text-align: center;
        padding: clamp(2rem, 5vw, 3rem);
        background: rgba(22, 22, 22, 0.95);
        border-radius: 12px;
        border: 1px solid rgba(0, 255, 255, 0.1);
    }

    .no-courses-icon {
        width: clamp(48px, 8vw, 64px);
        height: clamp(48px, 8vw, 64px);
        margin: 0 auto 1rem;
        display: flex;
        align-items: center;
        justify-content: center;
        background: linear-gradient(45deg, #00ffff20, #ff00ff20);
        border-radius: 12px;
    }

    .no-courses-icon i {
        font-size: clamp(1.25rem, 3vw, 1.75rem);
        color: #00ffff;
    }

    .no-courses p {
        color: rgba(255, 255, 255, 0.7);
        font-size: clamp(0.9rem, 2vw, 1rem);
        margin-bottom: 1.5rem;
    }

    @media (max-width: 768px) {
        .courses-grid {
            grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));
        }
    }
}
```

```

        }
    }

    @media (max-width: 480px) {
        .dashboard-container {
            padding: 1rem;
        }

        .section-header {
            flex-direction: column;
            align-items: flex-start;
            gap: 0.75rem;
        }

        .courses-grid {
            gap: 1rem;
        }

        .course-card {
            padding: 1rem;
        }
    }

```

</style>

{% endblock %}

teacher_dashboard.html

```

{% extends "dashboards/dashboard_base.html" %}

{% block dashboard_title %}Панель преподавателя{% endblock %}

{% block dashboard_content %}


<h3><i class="fas fa-graduation-cap"></i> Мои курсы</h3>
<p>Управляйте своими курсами и создавайте новые</p>
<div class="card-actions">
    <a href="{% url 'add_course' %}" class="btn btn-primary">
        <i class="fas fa-plus"></i> Создать курс
    </a>
    <a href="{% url 'courses_list' %}" class="btn btn-secondary">
        <i class="fas fa-list"></i> Все курсы
    </a>
</div>


```

```
</a>
</div>
</div>

<div class="dashboard-card">
  <h3><i class="fas fa-chart-line"></i> Статистика</h3>
  <p>Просматривайте статистику по курсам</p>
  <div class="card-actions">
    <a href="#" class="btn btn-secondary">
      <i class="fas fa-chart-bar"></i> Просмотреть отчеты
    </a>
  </div>
</div>
</div>

<style>
  .dashboard-grid {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 1.5rem;
  }

  .dashboard-card {
    background: rgba(255, 255, 255, 0.05);
    border: 1px solid rgba(0, 255, 255, 0.1);
    border-radius: 10px;
    padding: 1.5rem;
    transition: all 0.3s ease;
  }

  .dashboard-card:hover {
    transform: translateY(-5px);
    box-shadow: 0 5px 15px rgba(0, 255, 255, 0.1);
  }

  .dashboard-card h3 {
    color: #00ffff;
    font-size: 1.25rem;
    margin-bottom: 1rem;
    display: flex;
    align-items: center;
    gap: 0.75rem;
  }

  .dashboard-card p {
    color: rgba(255, 255, 255, 0.7);
    margin-bottom: 1.5rem;
  }

  .card-actions {
    display: flex;
    gap: 1rem;
    flex-wrap: wrap;
  }

  .btn {
    display: inline-flex;
    align-items: center;
    gap: 0.5rem;
    padding: 0.75rem 1.25rem;
    border-radius: 6px;
    font-size: 0.9rem;
    font-weight: 500;
  }
</style>
```

```

        text-decoration: none;
        transition: all 0.2s ease;
    }

    .btn i {
        font-size: 1rem;
    }

    .btn-primary {
        background: rgba(0, 255, 255, 0.1);
        color: #00ffff;
        border: 1px solid rgba(0, 255, 255, 0.2);
    }

    .btn-primary:hover {
        background: rgba(0, 255, 255, 0.15);
        border-color: rgba(0, 255, 255, 0.3);
    }

    .btn-secondary {
        background: rgba(255, 255, 255, 0.1);
        color: #fff;
        border: 1px solid rgba(255, 255, 255, 0.2);
    }

    .btn-secondary:hover {
        background: rgba(255, 255, 255, 0.15);
        border-color: rgba(255, 255, 255, 0.3);
    }

    @media (max-width: 768px) {
        .dashboard-grid {
            grid-template-columns: 1fr;
        }
    }

    .card-actions {
        flex-direction: column;
    }

    .btn {
        width: 100%;
        justify-content: center;
    }

```

</style>

{% endblock %}

training_manager_dashboard.html

{% extends "dashboards/dashboard_base.html" %}

{% block title %}Панель менеджера по обучению{% endblock %}

{% block extra_links %}

-  Создать план обучения
-  Рассмотреть заявки
-  Настроить сложность тестов

{% endblock %}

restore_backup.html

```

{% extends "bustComProj/base.html" %}

{% block content %}
<div class="restore-backup">
    <h1 class="page-title">Восстановление базы данных из бэкапа</h1>

    <div class="backup-form">
        <div class="warning-message">
            <i class="fas fa-exclamation-triangle"></i>
            <p>
                <strong>Внимание!</strong> Восстановление базы данных из бэкапа заменит все текущие
                данные.
                Перед продолжением убедитесь, что у вас есть актуальная резервная копия.
            </p>
        </div>

        <form method="post" enctype="multipart/form-data" class="upload-form">
            {% csrf_token %}
            <div class="form-group">
                <label for="{{ form.backup_file.id_for_label }}>Выберите файл бэкапа (.sqlite3):</label>
                {{ form.backup_file }}
            </div>
            <div class="form-actions">
                <button type="submit" class="btn btn-danger">
                    <i class="fas fa-upload"></i>
                    Восстановить из бэкапа
                </button>
                <a href="{% url 'admin_dashboard' %}" class="btn btn-secondary">
                    <i class="fas fa-times"></i>
                    Отмена
                </a>
            </div>
        </form>
    </div>
</div>

<style>
.restore-backup {
    max-width: 800px;
    margin: 2rem auto;
    padding: 2rem;
}

.page-title {
    font-size: clamp(1.5rem, 4vw, 2rem);
    color: var(--primary-color);
    margin-bottom: 2rem;
    text-align: center;
    text-shadow: 0 0 10px rgba(0, 255, 255, 0.3);
}

.backup-form {
    background: rgba(22, 22, 22, 0.95);
    border: 1px solid rgba(0, 255, 255, 0.1);
    border-radius: 12px;
    padding: 2rem;
}

.warning-message {
    display: flex;
    align-items: flex-start;
    gap: 1rem;
    background: rgba(255, 87, 34, 0.1);
}

```

```
border: 1px solid rgba(255, 87, 34, 0.3);
border-radius: 8px;
padding: 1rem;
margin-bottom: 2rem;
}

.warning-message i {
  color: #FF5722;
  font-size: 1.5rem;
}

.warning-message p {
  margin: 0;
  color: var(--text-primary);
}

.form-group {
  margin-bottom: 1.5rem;
}

.form-group label {
  display: block;
  margin-bottom: 0.5rem;
  color: var(--text-primary);
}

.form-group input[type="file"] {
  width: 100%;
  padding: 0.5rem;
  background: rgba(255, 255, 255, 0.05);
  border: 1px solid rgba(0, 255, 255, 0.1);
  border-radius: 6px;
  color: var(--text-primary);
}

.form-actions {
  display: flex;
  gap: 1rem;
  justify-content: flex-start;
  align-items: center;
}

.btn {
  display: inline-flex;
  align-items: center;
  gap: 0.5rem;
  padding: 0.75rem 1.5rem;
  border: none;
  border-radius: 6px;
  font-weight: 500;
  cursor: pointer;
  transition: all 0.3s ease;
  text-decoration: none;
}

.btn i {
  font-size: 1.1rem;
}

.btn-danger {
  background: #FF5722;
  color: white;
}
```

```

.btn-danger:hover {
    background: #F4511E;
    transform: translateY(-2px);
}

.btn-secondary {
    background: rgba(255, 255, 255, 0.1);
    color: var(--text-primary);
}

.btn-secondary:hover {
    background: rgba(255, 255, 255, 0.2);
    transform: translateY(-2px);
}

@media (max-width: 768px) {
    .restore-backup {
        padding: 1rem;
    }

    .backup-form {
        padding: 1rem;
    }

    .form-actions {
        flex-direction: column;
        width: 100%;
    }

    .btn {
        width: 100%;
        justify-content: center;
    }
}
</style>
{%- endblock %}

```

manage_table.html

```

{% extends "bustComProj/base.html" %}
{% load custom_filters %}

{% block content %}
<div class="admin-panel">
    <div class="panel-header">
        <h1 class="panel-title">{{ table_name }}</h1>
        <a href="{% url 'add_object' table_name %}" class="btn btn-primary">
            <i class="fas fa-plus"></i>
            Добавить запись
        </a>
    </div>

    <div class="table-container">
        <table>
            <thead>
                <tr>
                    {% for field in fields %}
                        <th>{{ field }}</th>
                    {% endfor %}
                    <th class="actions-column">Действия</th>
                </tr>
            <tbody>
                {% for row in table_rows %}
                    <tr>
                        <td>{{ row.id }}</td>
                        <td>{{ row.name }}</td>
                        <td>{{ row.description }}</td>
                        <td>{{ row.date }}</td>
                        <td>{{ row.status }}</td>
                        <td>
                            <a href="{% url 'edit_object' table_name row.id %}">Изменить</a>
                            <a href="#">Удалить</a>
                        </td>
                    </tr>
                {% endfor %}
            </tbody>
        </table>
    </div>
</div>

```

```

</thead>
<tbody>
    {% for obj in objects %}
        <tr>
            {% for field in fields %}
                <td data-label="{{ field }}>{{ obj|getattr_custom:field }}</td>
            {% endfor %}
            <td class="actions-cell" data-label="Действия">
                <div class="actions-wrapper">
                    <a href="{% url 'edit_object' table_name obj.id %}" class="btn btn-secondary btn-sm">
                        <i class="fas fa-edit"></i>
                        <span class="btn-text">Редактировать</span>
                    </a>
                    <a href="{% url 'delete_object' table_name obj.id %}" class="btn btn-danger btn-sm">
                        <i class="fas fa-trash"></i>
                        <span class="btn-text">Удалить</span>
                    </a>
                </div>
            </td>
        </tr>
    {% endfor %}
</tbody>
</table>
</div>
</div>

<style>
    .admin-panel {
        background: var(--background-light);
        border-radius: var(--border-radius);
        padding: var(--spacing-md);
        margin-bottom: var(--spacing-lg);
    }

    .panel-header {
        display: flex;
        justify-content: space-between;
        align-items: center;
        margin-bottom: var(--spacing-lg);
        flex-wrap: wrap;
        gap: var(--spacing-md);
    }

    .panel-title {
        font-size: 1.5rem;
        color: var(--primary-color);
        margin: 0;
    }

    .table-container {
        overflow-x: auto;
        background: var(--background-dark);
        border-radius: var(--border-radius);
        box-shadow: 0 4px 6px rgba(0, 0, 0, 0.1);
    }

    table {
        width: 100%;
        border-collapse: collapse;
    }

    th {

```

```
background: rgba(0, 0, 0, 0.2);
padding: var(--spacing-sm);
text-align: left;
color: var(--primary-color);
font-weight: 500;
white-space: nowrap;
}

td {
  padding: var(--spacing-sm);
  border-bottom: 1px solid rgba(255, 255, 255, 0.1);
  max-width: 200px;
  overflow: hidden;
  text-overflow: ellipsis;
  white-space: nowrap;
}

.actions-column {
  width: 200px;
}

.actions-cell {
  width: 200px;
}

.actions-wrapper {
  display: flex;
  gap: var(--spacing-xs);
  flex-wrap: wrap;
}

.btn-sm {
  padding: var(--spacing-xs) var(--spacing-sm);
  font-size: 0.8rem;
}

.btn-danger {
  background-color: #ff4444;
  color: white;
}

.btn-danger:hover {
  background-color: #cc0000;
}

@media (max-width: 768px) {
  .panel-header {
    flex-direction: column;
    align-items: stretch;
  }

  .panel-title {
    text-align: center;
  }

  table {
    display: block;
  }

  thead {
    display: none;
  }
}
```

```

tbody {
  display: block;
}

tr {
  display: block;
  margin-bottom: var(--spacing-md);
  background: var(--background-light);
  border-radius: var(--border-radius);
  padding: var(--spacing-sm);
}

td {
  display: block;
  text-align: right;
  padding: var(--spacing-xs) var(--spacing-sm);
  border: none;
  position: relative;
  padding-left: 50%;
}

td::before {
  content: attr(data-label);
  position: absolute;
  left: var(--spacing-sm);
  width: 45%;
  text-align: left;
  color: var(--text-secondary);
}

.actions-cell {
  width: 100%;
  text-align: center;
  padding-left: var(--spacing-sm);
}

.actions-wrapper {
  justify-content: center;
}

@media (max-width: 480px) {
  .btn-text {
    display: none;
  }

  .btn-sm {
    padding: var(--spacing-xs);
  }

  .actions-wrapper {
    gap: var(--spacing-md);
  }
}

</style>
{% endblock %}

```

edit_object.html

```
{% extends "bustComProj/base.html" %}
```

```

{% block title %} Редактировать {{ table_name }} {% endblock %}

{% block content %}
<h1>Редактировать {{ table_name }}</h1>
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Сохранить</button>
</form>
<a href="{% url 'manage_table' table_name=table_name %}">← Назад</a>
{% endblock %}

```

add_object.html

```

{% extends "bustComProj/base.html" %}

{% block title %} Добавить в {{ table_name }} {% endblock %}

{% block content %}
<h1>Добавить в {{ table_name }}</h1>
<form method="post">
    {% csrf_token %}
    {{ form.as_p }}
    <button type="submit">Сохранить</button>
</form>
<a href="{% url 'manage_table' table_name=table_name %}">← Назад</a>
{% endblock %}

```

test_difficulty.html

```

{% extends "bustComProj/base.html" %}

{% block title %} 📈 Настройка сложности тестов {% endblock %}

{% block content %}
<h2>📈 Настроить сложность тестов</h2>

<table>
    <tr>
        <th>Название теста</th>
        <th>Текущая сложность</th>
        <th>Изменить</th>
    </tr>
    {% for test in tests %}
    <tr>
        <td>{{ test.name }}</td>
        <td>{{ test.get_difficulty_display }}</td>
        <td>
            <form method="post">
                {% csrf_token %}
                <select name="difficulty">
                    <option value="низкая">Низкая</option>
                    <option value="средняя">Средняя</option>
                    <option value="высокая">Высокая</option>
                </select>
                <button type="submit">📝 Сохранить</button>
            </form>
        </td>
    </tr>
    {% endfor %}

```

```
</table>

{% endblock %}
```

review_requests.html

```
{% extends 'bustComProj/base.html' %}

{% block title %}Просмотр заявок на обучение{% endblock %}

{% block content %}


# Просмотр заявок на обучение



| Сотрудник                        | Курс                       | Дата заявки                           | Статус                                                                                           | Действия                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------------------|----------------------------|---------------------------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {{ request.user.get_full_name }} | {{ request.course.title }} | {{ request.created_at date:"d.m.Y" }} | <span class="status-badge status-{{ request.status }}"> {{ request.get_status_display }} </span> | {% if request.status == 'pending' %} <form method="post" style="display: inline;">     {% csrf_token %}     <input type="hidden" name="request_id" value="{{ request.id }}"/>     <input type="hidden" name="action" value="approve">     <button type="submit" class="btn btn-success btn-sm">         <i class="fas fa-check"></i>     </button> </form> <form method="post" style="display: inline;">     {% csrf_token %}     <input type="hidden" name="request_id" value="{{ request.id }}"/>     <input type="hidden" name="action" value="reject">     <button type="submit" class="btn btn-danger btn-sm">         <i class="fas fa-times"></i>     </button> </form> {% endif %} |
|                                  |                            |                                       |                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |


```

```

        <td colspan="5" class="text-center">Нет активных заявок на обучение</td>
    </tr>
    {% endfor %}
    </tbody>
    </table>
    </div>
</div>

<style>
.status-badge {
    padding: 4px 8px;
    border-radius: 12px;
    font-size: 0.875rem;
    font-weight: 500;
}

.status-pending {
    background-color: rgba(255, 193, 7, 0.1);
    color: #ffc107;
}

.status-approved {
    background-color: rgba(0, 255, 0, 0.1);
    color: #00ff00;
}

.status-rejected {
    background-color: rgba(255, 0, 0, 0.1);
    color: #ff0000;
}

.btn-sm {
    padding: 4px 8px;
    font-size: 0.875rem;
}

.btn-success {
    background: linear-gradient(135deg, var(--success-color), var(--success-dark));
}

.btn-danger {
    background: linear-gradient(135deg, var(--danger-color), var(--danger-dark));
}
</style>
{% endblock %}

```

create_training_plan.html

```
{% extends "bustComProj/base.html" %}
```

```
{% block title %}Создание плана обучения{% endblock %}
```

```
{% block content %}
```

```
<h2>📝 Создание плана обучения</h2>
```

```
<form method="post">
```

```
{% csrf_token %}
```

```
<label for="name">Название плана:</label>
```

```
<input type="text" id="name" name="name" required>
```

```

<label for="description">Описание:</label>
<textarea id="description" name="description" rows="4" required></textarea>

<label for="start_date">Дата начала:</label>
<input type="date" id="start_date" name="start_date" required>

<label for="end_date">Дата окончания:</label>
<input type="date" id="end_date" name="end_date">

<label for="courses">Выберите курсы:</label>
<select id="courses" name="courses" multiple>
    {% for course in courses %}
        <option value="{{ course.id }}>{{ course.name }}</option>
    {% endfor %}
</select>

<button type="submit">💾 Сохранить</button>
</form>

{% endblock %}

```

styles.css

```

/* 💾 Общие стили */
:root {
    --primary-color: #ff00ff;
    --secondary-color: #00ffcc;
    --accent-color: #ff3366;
    --background-dark: #1a1a1a;
    --background-light: #2a2a2a;
    --text-primary: #ffffff;
    --text-secondary: #e0e0e0;
    --success-color: #00ff88;
    --danger-color: #ff3366;
    --warning-color: #ffcc00;
    --border-radius: 15px;
    --transition-speed: 0.3s;
}

html, body {
    height: 100%;
    margin: 0;
    padding: 0;
    font-family: 'Inter', system-ui, -apple-system, sans-serif;
    background: var(--background-dark);
    color: var(--text-primary);
    font-size: 18px;
    line-height: 1.6;
}

#root {
    display: flex;
    flex-direction: column;
    min-height: 100vh;
}

header {
    background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));
    color: var(--text-primary);
    padding: 1.5rem 2rem;
    display: flex;
    justify-content: space-between;
}

```

```
align-items: center;
box-shadow: 0 4px 20px rgba(255, 0, 255, 0.3);
position: sticky;
top: 0;
z-index: 1000;
}

.logo {
  font-size: 2rem;
  font-weight: 700;
  color: var(--text-primary);
  text-decoration: none;
  display: flex;
  align-items: center;
  gap: 0.5rem;
  text-shadow: 0 0 10px rgba(255, 255, 255, 0.7);
}

nav {
  display: flex;
  gap: 1.5rem;
  align-items: center;
}

nav a {
  color: var(--text-primary);
  text-decoration: none;
  padding: 0.75rem 1.5rem;
  border-radius: 25px;
  transition: all var(--transition-speed) ease;
  display: flex;
  align-items: center;
  gap: 0.5rem;
  background: rgba(255, 255, 255, 0.1);
  font-weight: 500;
}

nav a:hover {
  background: rgba(255, 255, 255, 0.2);
  transform: translateY(-3px);
  box-shadow: 0 5px 15px rgba(255, 0, 255, 0.3);
}

nav a.active {
  background: var(--text-primary);
  color: var(--primary-color);
  font-weight: 600;
}

main {
  flex: 1;
  padding: 2rem;
  width: 100%;
  max-width: 1400px;
  margin: 0 auto;
}

.container {
  background: var(--background-light);
  border-radius: var(--border-radius);
  padding: 2rem;
  box-shadow: 0 0 30px rgba(0, 0, 0, 0.3);
  border: 1px solid rgba(255, 0, 255, 0.2);
```

```
}

.card {
  background: linear-gradient(135deg, var(--background-light), var(--background-dark));
  border-radius: var(--border-radius);
  padding: 2rem;
  margin: 1.5rem 0;
  transition: all var(--transition-speed) ease;
  border: 1px solid rgba(255, 0, 255, 0.2);
  box-shadow: 0 0 20px rgba(0, 0, 0, 0.2);
}

.card:hover {
  transform: translateY(-5px);
  box-shadow: 0 10px 30px rgba(255, 0, 255, 0.2);
  border-color: var(--primary-color);
}

.btn,
button,
input[type="submit"],
input[type="button"],
.btn-submit,
.btn-mark-complete,
.btn-enroll,
a[class*="btn-"],
.button {
  display: inline-flex;
  align-items: center;
  justify-content: center;
  padding: 0.8rem 2rem;
  font-size: 1.1rem;
  font-weight: 600;
  text-transform: uppercase;
  letter-spacing: 1px;
  border: none;
  border-radius: 50px;
  cursor: pointer;
  transition: all 0.3s cubic-bezier(0.4, 0, 0.2, 1);
  background: linear-gradient(45deg, var(--primary-color), var(--secondary-color));
  color: white;
  text-decoration: none;
  box-shadow: 0 4px 15px rgba(255, 0, 255, 0.3);
  position: relative;
  overflow: hidden;
  margin: 0.5rem;
  min-width: 150px;
}

.btn::before,
button::before,
input[type="submit"]::before,
input[type="button"]::before,
.btn-submit::before,
.btn-mark-complete::before,
.btn-enroll::before,
a[class*="btn-"]::before,
.button::before {
  content: "";
  position: absolute;
  top: 0;
  left: -100%;
  width: 100%;
```

```
height: 100%;  
background: linear-gradient(  
    90deg,  
    transparent,  
    rgba(255, 255, 255, 0.3),  
    transparent  
>);  
transition: 0.5s;  
}  
  
.btn:hover,  
button:hover,  
input[type="submit"]:hover,  
input[type="button"]:hover,  
.btn-submit:hover,  
.btn-mark-complete:hover,  
.btn-enroll:hover,  
a[class*="btn-"]:hover,  
.button:hover {  
    transform: translateY(-3px);  
    box-shadow: 0 8px 25px rgba(255, 0, 255, 0.5);  
    background: linear-gradient(45deg, var(--secondary-color), var(--primary-color));  
}  
  
.btn:hover::before,  
button:hover::before,  
input[type="submit"]:hover::before,  
input[type="button"]:hover::before,  
.btn-submit:hover::before,  
.btn-mark-complete:hover::before,  
.btn-enroll:hover::before,  
a[class*="btn-"]:hover::before,  
.button:hover::before {  
    left: 100%;  
}  
  
.btn:active,  
button:active,  
input[type="submit"]:active,  
input[type="button"]:active,  
.btn-submit:active,  
.btn-mark-complete:active,  
.btn-enroll:active,  
a[class*="btn-"]:active,  
.button:active {  
    transform: translateY(1px);  
    box-shadow: 0 2px 10px rgba(255, 0, 255, 0.4);  
}  
  
.btn-submit,  
input[type="submit"] {  
    background: linear-gradient(45deg, #ff00ff, #ff3366);  
    box-shadow: 0 4px 15px rgba(255, 0, 255, 0.4);  
}  
  
.btn-submit:hover,  
input[type="submit"]:hover {  
    background: linear-gradient(45deg, #ff3366, #ff00ff);  
    box-shadow: 0 8px 25px rgba(255, 0, 255, 0.6);  
}  
  
.btn-mark-complete {  
    background: linear-gradient(45deg, #00ffcc, #00ff88);
```

```
    box-shadow: 0 4px 15px rgba(0, 255, 204, 0.4);
    color: #1a1a1a;
}

.btn-mark-complete:hover {
    background: linear-gradient(45deg, #00ff88, #00ffcc);
    box-shadow: 0 8px 25px rgba(0, 255, 204, 0.6);
}

.btn-enroll {
    background: linear-gradient(45deg, #ffcc00, #ff9900);
    box-shadow: 0 4px 15px rgba(255, 204, 0, 0.4);
    color: #1a1a1a;
    font-size: 1.2rem;
    padding: 1rem 2.5rem;
}

.btn-enroll:hover {
    background: linear-gradient(45deg, #ff9900, #ffcc00);
    box-shadow: 0 8px 25px rgba(255, 204, 0, 0.6);
}

.btn-sm,
.button-sm {
    padding: 0.5rem 1.5rem;
    font-size: 0.9rem;
    min-width: 100px;
}

.btn-lg,
.button-lg {
    padding: 1.2rem 3rem;
    font-size: 1.3rem;
    min-width: 200px;
}

.btn i,
button i,
.btn-submit i,
.btn-mark-complete i,
.btn-enroll i,
a[class*="btn-"] i,
.button i {
    margin-right: 0.5rem;
    font-size: 1.2em;
}

.btn:disabled,
button:disabled,
input[type="submit"]:disabled,
input[type="button"]:disabled,
.btn-submit:disabled,
.btn-mark-complete:disabled,
.btn-enroll:disabled,
a[class*="btn-"].disabled,
.button:disabled {
    opacity: 0.7;
    cursor: not-allowed;
    transform: none !important;
    box-shadow: none !important;
}

.btn-group {
```

```
display: flex;
gap: 1rem;
margin: 1rem 0;
}

.btn-danger {
background: linear-gradient(45deg, #ff3366, #ff0000);
box-shadow: 0 4px 15px rgba(255, 51, 102, 0.3);
}

.btn-danger:hover {
background: linear-gradient(45deg, #ff0000, #ff3366);
box-shadow: 0 8px 25px rgba(255, 51, 102, 0.5);
}

.btn-success {
background: linear-gradient(45deg, #00ff88, #00ffcc);
box-shadow: 0 4px 15px rgba(0, 255, 136, 0.3);
color: #1a1a1a;
}

.btn-success:hover {
background: linear-gradient(45deg, #00ffcc, #00ff88);
box-shadow: 0 8px 25px rgba(0, 255, 136, 0.5);
}

form {
display: flex;
flex-direction: column;
gap: 1.5rem;
}

input, textarea, select {
padding: 1rem 1.5rem;
border-radius: var(--border-radius);
border: 2px solid var(--secondary-color);
background: var(--background-dark);
color: var(--text-primary);
font-size: 1.1rem;
transition: all var(--transition-speed) ease;
}

input:focus, textarea:focus, select:focus {
outline: none;
border-color: var(--primary-color);
box-shadow: 0 0 15px rgba(255, 0, 255, 0.3);
}

h1, h2, h3, h4, h5, h6 {
color: var(--text-primary);
font-weight: 700;
line-height: 1.2;
margin-bottom: 1.5rem;
text-shadow: 0 0 10px rgba(255, 255, 255, 0.5);
}

h1 {
font-size: 3rem;
color: var(--secondary-color);
text-shadow: 0 0 15px rgba(0, 255, 204, 0.7);
}

h2 {
```

```
font-size: 2.5rem;
color: var(--primary-color);
text-shadow: 0 0 15px rgba(255, 0, 255, 0.7);
}

h3 {
font-size: 2rem;
color: var(--secondary-color);
text-shadow: 0 0 10px rgba(0, 255, 204, 0.7);
}

@media (max-width: 768px) {
header {
padding: 1rem;
flex-direction: column;
gap: 1rem;
}

nav {
display: none;
position: absolute;
top: 100%;
left: 0;
right: 0;
background: var(--background-light);
padding: 1rem;
flex-direction: column;
gap: 0.75rem;
box-shadow: 0 4px 20px rgba(0, 0, 0, 0.3);
}

nav.active {
display: flex;
}

.container {
padding: 1.5rem;
}

h1 {
font-size: 2.5rem;
}

h2 {
font-size: 2rem;
}

h3 {
font-size: 1.75rem;
}

}

@keyframes fadeIn {
from {
opacity: 0;
transform: translateY(20px);
}
to {
opacity: 1;
transform: translateY(0);
}
}
```

```
.fade-in {
  animation: fadeIn 0.5s ease forwards;
}

.text-center {
  text-align: center;
}

.mt-1 { margin-top: 0.5rem; }
.mt-2 { margin-top: 1rem; }
.mt-3 { margin-top: 1.5rem; }
.mt-4 { margin-top: 2rem; }
.mt-5 { margin-top: 2.5rem; }

.mb-1 { margin-bottom: 0.5rem; }
.mb-2 { margin-bottom: 1rem; }
.mb-3 { margin-bottom: 1.5rem; }
.mb-4 { margin-bottom: 2rem; }
.mb-5 { margin-bottom: 2.5rem; }

.notification {
  position: fixed;
  bottom: 2rem;
  right: 2rem;
  padding: 1.5rem 2.5rem;
  border-radius: var(--border-radius);
  background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));
  color: var(--text-primary);
  box-shadow: 0 4px 20px rgba(255, 0, 255, 0.3);
  animation: slideIn 0.3s ease forwards;
  z-index: 1000;
  font-weight: 500;
  text-shadow: 0 0 10px rgba(255, 255, 255, 0.5);
}

@keyframes slideIn {
  from {
    transform: translateX(100%);
    opacity: 0;
  }
  to {
    transform: translateX(0);
    opacity: 1;
  }
}

a {
  color: var(--secondary-color);
  text-decoration: none;
  font-weight: 600;
  transition: all var(--transition-speed) ease;
  display: inline-flex;
  align-items: center;
  gap: 0.5rem;
}

a:hover {
  color: var(--primary-color);
  text-shadow: 0 0 10px rgba(255, 255, 255, 0.7);
  transform: translateY(-2px);
}

ul li a {
```

```
display: block;
padding: 1rem 1.5rem;
border-radius: var(--border-radius);
background: linear-gradient(135deg, var(--background-light), var(--background-dark));
color: var(--text-primary);
transition: all var(--transition-speed) ease;
border: 1px solid rgba(255, 0, 255, 0.2);
box-shadow: 0 0 20px rgba(0, 0, 0, 0.2);
}

ul li a:hover {
  background: linear-gradient(135deg, var(--primary-color), var(--secondary-color));
  color: var(--text-primary);
  transform: translateY(-3px);
  box-shadow: 0 5px 15px rgba(255, 0, 255, 0.3);
}

.card a {
  color: var(--secondary-color);
  text-decoration: none;
  font-weight: 600;
  transition: all var(--transition-speed) ease;
  display: inline-flex;
  align-items: center;
  gap: 0.5rem;
  padding: 0.5rem 1rem;
  border-radius: 20px;
  background: rgba(255, 255, 255, 0.1);
}

.card a:hover {
  background: rgba(255, 255, 255, 0.2);
  color: var(--primary-color);
  transform: translateY(-2px);
}

.checkbox-wrapper {
  display: flex;
  align-items: center;
  gap: 0.5rem;
  margin: 1rem 0;
}

.custom-checkbox {
  appearance: none;
  -webkit-appearance: none;
  width: 1.5rem;
  height: 1.5rem;
  border: 2px solid var(--secondary-color);
  border-radius: 5px;
  background: var(--background-dark);
  cursor: pointer;
  position: relative;
  transition: all 0.3s ease;
}

.custom-checkbox:checked {
  background: var(--secondary-color);
  border-color: var(--secondary-color);
}

.custom-checkbox:checked::before {
```

```

        content: '✓';
        position: absolute;
        color: var(--background-dark);
        font-size: 1rem;
        font-weight: bold;
        left: 50%;
        top: 50%;
        transform: translate(-50%, -50%);
    }

.custom-checkbox:hover {
    border-color: var(--primary-color);
    box-shadow: 0 0 10px rgba(255, 0, 255, 0.2);
}

```

track_performance.html

```
{% extends 'bustComProj/base.html' %}
```

```
{% block title %}  Мониторинг успеваемости сотрудников {% endblock %}
```

```
{% block content %}
```

```
<h1>  Мониторинг успеваемости сотрудников </h1>
```

```
<p>Средний балл: <strong>{{ avg_score }}</strong></p>
```

```
<p>Максимальный балл: <strong>{{ max_score }}</strong></p>
```

```
<p>Минимальный балл: <strong>{{ min_score }}</strong></p>
```

```
<h2>  Гистограмма оценок </h2>
```

```
<canvas id="scoreChart"></canvas>
```

```
<h2>  Распределение оценок </h2>
```

```
<canvas id="distributionChart"></canvas>
```

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script>
```

```
    var ctx1 = document.getElementById('scoreChart').getContext('2d');
    new Chart(ctx1, {
        type: 'bar',
        data: {
            labels: ["Средний", "Максимальный", "Минимальный"],
            datasets: [
                {
                    label: "Балы",
                    data: [{ avg_score }, { max_score }, { min_score }],
                    backgroundColor: ['#00ffcc', '#ff00ff', '#ff6600']
                }
            ],
            options: { responsive: true, scales: { y: { beginAtZero: true } } }
        });
    
```

```
    var ctx2 = document.getElementById('distributionChart').getContext('2d');
```

```
    new Chart(ctx2, {
        type: 'pie',
        data: {
            labels: [% for score in scores %]{{ score }}[% if not forloop.last %], {% endif %}[% endfor %],
            datasets: [
                {
                    data: [% for count in counts %]{{ count }}[% if not forloop.last %], {% endif %}[% endfor %],
                    backgroundColor: ['#ff0000', '#ff8000', '#ffff00', '#80ff00', '#00ff00', '#00ff80', '#00ffff']
                }
            ],
        },
    });
}
```

```

        options: { responsive: true }
    });
</script>
{%- endblock %}

```


assign_courses.html

```

{% extends 'bustComProj/base.html' %}

{% block title %}Назначение курсов{% endblock %}

{% block content %}


<h1 class="mb-4">Назначение курсов</h1>

    <div class="card mb-4">
        <h2 class="mb-3">Назначить курс сотруднику</h2>
        <form method="post">
            {% csrf_token %}
            <div class="form-group">
                <label for="employee" class="form-label">Выберите сотрудника</label>
                <select name="employee" id="employee" class="form-control" required>
                    <option value="">-- Выберите сотрудника --</option>
                    {% for employee in employees %}
                        <option value="{{ employee.id }}>{{ employee.get_full_name }}</option>
                    {% endfor %}
                </select>
            </div>

            <div class="form-group">
                <label for="course" class="form-label">Выберите курс</label>
                <select name="course" id="course" class="form-control" required>
                    <option value="">-- Выберите курс --</option>
                    {% for course in courses %}
                        <option value="{{ course.id }}>{{ course.title }}</option>
                    {% endfor %}
                </select>
            </div>

            <button type="submit" class="btn">
                <i class="fas fa-plus"></i> Назначить курс
            </button>
        </form>
    </div>

    <div class="card">
        <h2 class="mb-3">Текущие назначения</h2>
        <div class="table-container">
            <table>
                <thead>
                    <tr>
                        <th>Сотрудник</th>
                        <th>Курс</th>
                        <th>Дата назначения</th>
                        <th>Статус</th>
                        <th>Действия</th>
                    </tr>
                </thead>
                <tbody>
                    {% for enrollment in enrollments %}
                        <tr>


```

```

<td>{{ enrollment.user.get_full_name }}</td>
<td>{{ enrollment.course.title }}</td>
<td>{{ enrollment.date_enrolled|date:"d.m.Y" }}</td>
<td>{{ enrollment.get_status_display }}</td>
<td>
    <form method="post" style="display: inline;">
        {% csrf_token %}
        <input type="hidden" name="action" value="delete">
        <input type="hidden" name="enrollment_id" value="{{ enrollment.id }}">
        <button type="submit" class="btn btn-danger">
            <i class="fas fa-trash"></i>
        </button>
    </form>
</td>
</tr>
{% empty %}
<tr>
    <td colspan="5" class="text-center">Нет активных назначений</td>
</tr>
{% endfor %}
</tbody>
</table>
</div>
</div>
</div>
{% endblock %}

```

training_evaluation.html

```
{% extends 'bustComProj/base.html' %}
```

```
{% block title %}Оценка обучения{% endblock %}
```

```
{% block content %}
<h2>Оценка обучения</h2>
<p>Здесь будет информация о результатах оценки обучения.</p>
{% endblock %}
```

assign_course.html

```
{% extends "dashboards/dashboard_base.html" %}
```

```
{% block title %}Назначение курсов сотрудникам{% endblock %}
```

```
{% block content %}
<h2>Назначение курсов сотрудникам</h2>
<form method="post">
    {% csrf_token %}

    <label for="user_id">Выберите сотрудника:</label>
    <select name="user_id">
        {% for user in users %}
            <option value="{{ user.id }}>{{ user.username }}</option>
        {% endfor %}
    </select>

    <label for="course_id">Выберите курс:</label>
    <select name="course_id">
        {% for course in courses %}
```

```

<option value="{{ course.id }}">{{ course.name }}</option>
{% endfor %}
</select>

<button type="submit">Назначить</button>
</form>
{% endblock %}

```

add_lecture_material.html

```
{% extends 'bustComProj/base.html' %}
```

```
{% block title %}Добавить материал к лекции{% endblock %}
```

```

{% block content %}
<div class="container">
    <h1 class="page-title">Добавить материал к лекции</h1>

    {% if messages %}
        <div class="messages">
            {% for message in messages %}
                <div class="message" {% if message.tags %}{{ message.tags }}{% endif %}>
                    {{ message }}
                </div>
            {% endfor %}
        </div>
    {% endif %}

```

```

    {% if lectures %}
        <form method="post" class="form">
            {% csrf_token %}

            <div class="form-group">
                <label for="lecture_id">Выберите лекцию:</label>
                <select name="lecture_id" id="lecture_id" required class="form-control">
                    <option value="">-- Выберите лекцию --</option>
                    {% for lecture in lectures %}
                        <option value="{{ lecture.id }}>{{ lecture.course.name }} - {{ lecture.title }}</option>
                    {% endfor %}
                </select>
            </div>

```

```

            <div class="form-group">
                <label for="resource_type">Тип ресурса:</label>
                <select name="resource_type" id="resource_type" required class="form-control">
                    <option value="file">Файл</option>
                    <option value="video">Видео</option>
                </select>
            </div>

```

```

            <div class="form-group">
                <label for="resource_path">Ссылка на ресурс:</label>
                <input type="text" name="resource_path" id="resource_path" required class="form-control"
                       placeholder="Введите URL файла или видео">
            </div>

```

```

            <div class="form-actions">
                <button type="submit" class="btn btn-primary">Добавить материал</button>
                <a href="{% url 'teacher_dashboard' %}" class="btn btn-secondary">Назад</a>
            </div>
        </form>

```

```
{% else %}
<div class="no-lectures">
    <p>У вас пока нет лекций. Сначала создайте курс и добавьте в него лекции.</p>
    <a href="{% url 'create_course' %}" class="btn btn-primary">Создать курс</a>
</div>
{% endif %}
</div>

<style>
.container {
    max-width: 800px;
    margin: 0 auto;
    padding: 2rem 1rem;
}

.page-title {
    font-size: clamp(1.5rem, 4vw, 2rem);
    color: #00ffff;
    margin-bottom: 2rem;
    text-align: center;
    text-shadow: 0 0 10px rgba(0, 255, 255, 0.3);
}

.form {
    background: rgba(22, 22, 22, 0.95);
    border: 1px solid rgba(0, 255, 255, 0.1);
    border-radius: 12px;
    padding: 2rem;
    display: flex;
    flex-direction: column;
    gap: 1.5rem;
}

.form-group {
    display: flex;
    flex-direction: column;
    gap: 0.5rem;
}

.form-group label {
    color: rgba(255, 255, 255, 0.9);
    font-size: 0.9rem;
}

.form-control {
    background: rgba(255, 255, 255, 0.05);
    border: 1px solid rgba(0, 255, 255, 0.2);
    border-radius: 6px;
    padding: 0.75rem;
    color: #fff;
    font-size: 0.9rem;
    width: 100%;
    transition: all 0.2s ease;
}

.form-control:focus {
    outline: none;
    border-color: rgba(0, 255, 255, 0.5);
    box-shadow: 0 0 2px rgba(0, 255, 255, 0.1);
}

.form-control option {
    background: #1a1a1a;
```

```
        color: #fff;
    }

.form-actions {
    display: flex;
    gap: 1rem;
    margin-top: 1rem;
}

.btn {
    display: inline-flex;
    align-items: center;
    justify-content: center;
    padding: 0.75rem 1.5rem;
    border-radius: 8px;
    font-size: 0.9rem;
    font-weight: 500;
    text-decoration: none;
    transition: all 0.2s ease;
    border: none;
    cursor: pointer;
}

.btn-primary {
    background: rgba(0, 255, 255, 0.1);
    color: #00ffff;
    border: 1px solid rgba(0, 255, 255, 0.2);
}

.btn-primary:hover {
    background: rgba(0, 255, 255, 0.15);
    border-color: rgba(0, 255, 255, 0.3);
}

.btn-secondary {
    background: rgba(255, 255, 255, 0.1);
    color: #fff;
    border: 1px solid rgba(255, 255, 255, 0.2);
}

.btn-secondary:hover {
    background: rgba(255, 255, 255, 0.15);
    border-color: rgba(255, 255, 255, 0.3);
}

.messages {
    margin-bottom: 1.5rem;
}

.message {
    padding: 1rem;
    border-radius: 8px;
    margin-bottom: 0.5rem;
}

.message.error {
    background: rgba(255, 68, 68, 0.1);
    border: 1px solid rgba(255, 68, 68, 0.2);
    color: #ff4444;
}

.message.success {
    background: rgba(0, 255, 149, 0.1);
```

```

border: 1px solid rgba(0, 255, 149, 0.2);
color: #00ff95;
}

.no-lectures {
    text-align: center;
    padding: 3rem;
    background: rgba(22, 22, 22, 0.95);
    border: 1px solid rgba(0, 255, 255, 0.1);
    border-radius: 12px;
}

.no-lectures p {
    color: rgba(255, 255, 255, 0.7);
    margin-bottom: 1.5rem;
}

@media (max-width: 768px) {
    .form {
        padding: 1.5rem;
    }

    .form-actions {
        flex-direction: column;
    }

    .btn {
        width: 100%;
    }
}
</style>
{% endblock %}

```

create_course.html

```

{% extends 'bustComProj/base.html' %}

{% block title %}Создать курс{% endblock %}

{% block content %}
<div class="container">
    <div class="form-container">
        <h1 class="page-title">Создать новый курс</h1>

        {% if messages %}
            <div class="messages">
                {% for message in messages %}
                    <div class="message" {% if message.tags %}{% message.tags %}{% endif %}>
                        {{ message }}
                    </div>
                {% endfor %}
            </div>
        {% endif %}

        <form method="post" class="form" action="{% url 'add_course' %}">
            {% csrf_token %}

            {% for field in form %}
                <div class="form-group">
                    <label for="{{ field.id_for_label }}>{{ field.label }}</label>
                    {{ field }}
                    {% if field.help_text %}

```

```

        <small class="help-text">{{ field.help_text }}</small>
    {% endif %}
    {% if field.errors %}
        <div class="error-messages">
            {% for error in field.errors %}
                <span class="error">{{ error }}</span>
            {% endfor %}
        </div>
    {% endif %}
</div>
{% endfor %}

<div class="form-actions">
    <button type="submit" class="btn btn-primary">
        <i class="fas fa-save"></i> Создать курс
    </button>
    <a href="{% url 'teacher_dashboard' %}" class="btn btn-secondary">
        <i class="fas fa-times"></i> Отмена
    </a>
</div>
</form>
</div>
</div>

<style>
.container {
    max-width: 800px;
    margin: 0 auto;
    padding: 2rem 1rem;
}

.form-container {
    background: rgba(22, 22, 22, 0.95);
    border: 1px solid rgba(0, 255, 255, 0.1);
    border-radius: 12px;
    padding: 2rem;
}

.page-title {
    font-size: clamp(1.5rem, 4vw, 2rem);
    color: #00ffff;
    margin-bottom: 2rem;
    text-align: center;
    text-shadow: 0 0 10px rgba(0, 255, 255, 0.3);
}

.form {
    display: flex;
    flex-direction: column;
    gap: 1.5rem;
}

.form-group {
    display: flex;
    flex-direction: column;
    gap: 0.5rem;
}

.form-group label {
    color: rgba(255, 255, 255, 0.9);
    font-size: 0.9rem;
}

```

```
.form-control {  
    background: rgba(255, 255, 255, 0.05);  
    border: 1px solid rgba(0, 255, 255, 0.2);  
    border-radius: 6px;  
    padding: 0.75rem;  
    color: #fff;  
    font-size: 0.9rem;  
    width: 100%;  
    transition: all 0.2s ease;  
}  
  
.form-control:focus {  
    outline: none;  
    border-color: rgba(0, 255, 255, 0.5);  
    box-shadow: 0 0 0 2px rgba(0, 255, 255, 0.1);  
}  
  
.help-text {  
    color: rgba(255, 255, 255, 0.5);  
    font-size: 0.8rem;  
}  
  
.error-messages {  
    color: #ff4444;  
    font-size: 0.8rem;  
    margin-top: 0.25rem;  
}  
  
.form-actions {  
    display: flex;  
    gap: 1rem;  
    margin-top: 1rem;  
}  
  
.btn {  
    display: inline-flex;  
    align-items: center;  
    gap: 0.5rem;  
    padding: 0.75rem 1.5rem;  
    border-radius: 8px;  
    font-size: 0.9rem;  
    font-weight: 500;  
    text-decoration: none;  
    transition: all 0.2s ease;  
    border: none;  
    cursor: pointer;  
}  
  
.btn i {  
    font-size: 1rem;  
}  
  
.btn-primary {  
    background: rgba(0, 255, 255, 0.1);  
    color: #00ffff;  
    border: 1px solid rgba(0, 255, 255, 0.2);  
}  
  
.btn-primary:hover {  
    background: rgba(0, 255, 255, 0.15);  
    border-color: rgba(0, 255, 255, 0.3);  
}
```

```

.btn-secondary {
    background: rgba(255, 255, 255, 0.1);
    color: #fff;
    border: 1px solid rgba(255, 255, 255, 0.2);
}

.btn-secondary:hover {
    background: rgba(255, 255, 255, 0.15);
    border-color: rgba(255, 255, 255, 0.3);
}

.messages {
    margin-bottom: 1.5rem;
}

.message {
    padding: 1rem;
    border-radius: 8px;
    margin-bottom: 0.5rem;
}

.message.success {
    background: rgba(0, 255, 149, 0.1);
    border: 1px solid rgba(0, 255, 149, 0.2);
    color: #00ff95;
}

.message.error {
    background: rgba(255, 68, 68, 0.1);
    border: 1px solid rgba(255, 68, 68, 0.2);
    color: #ff4444;
}

@media (max-width: 768px) {
    .container {
        padding: 1rem;
    }

    .form-container {
        padding: 1.5rem;
    }

    .form-actions {
        flex-direction: column;
    }

    .btn {
        width: 100%;
        justify-content: center;
    }
}
</style>
{%
    endblock %}

```

edit_course.html

```

{% extends "bustComProj/base.html" %}

{% block title %}Редактировать курс{% endblock %}

{% block content %}
<h2>🔗 Редактирование курса: {{ course.name }}</h2>

```



```
<div class="no-cert-icon">
    <i class="fas fa-award"></i>
</div>
<p>У вас пока нет сертификатов. Завершите курс, чтобы получить сертификат.</p>
<button class="btn-action">
    <a href="{% url 'courses_list' %}">
        <i class="fas fa-graduation-cap"></i>
        <span>Перейти к курсам</span>
    </a>
</button>
</div>
{% endif %}
</div>
</div>

<style>
.page-title {
    text-align: center;
    color: #00ffff;
    font-size: 2.5rem;
    margin-bottom: 2rem;
    text-shadow: 0 0 15px rgba(0, 255, 255, 0.3);
    font-weight: 600;
}

.certificates-grid {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
    gap: 1.5rem;
    padding: 1rem;
}

.certificate-card {
    background: rgba(22, 22, 22, 0.95);
    border: 1px solid rgba(0, 255, 255, 0.1);
    border-radius: 16px;
    padding: 1.5rem;
    transition: all 0.3s ease;
    position: relative;
    overflow: hidden;
    width: 100%;
    box-sizing: border-box;
}

.certificate-card::before {
    content: "";
    position: absolute;
    inset: 0;
    border-radius: 16px;
    padding: 1px;
    background: linear-gradient(45deg, #00ffff20, #ff00ff20);
    -webkit-mask: linear-gradient(#fff 0 0) content-box, linear-gradient(#fff 0 0);
    mask: linear-gradient(#fff 0 0) content-box, linear-gradient(#fff 0 0);
    -webkit-mask-composite: xor;
    mask-composite: exclude;
    pointer-events: none;
}

.certificate-card:hover {
    transform: translateY(-4px);
    box-shadow: 0 8px 30px rgba(0, 255, 255, 0.15);
}
```

```
.certificate-card:hover::before {
  background: linear-gradient(45deg, #00ffff40, #ff00ff40);
}

.certificate-header {
  display: flex;
  align-items: center;
  gap: 1rem;
  margin-bottom: 1.5rem;
}

.certificate-icon {
  width: 40px;
  height: 40px;
  display: flex;
  align-items: center;
  justify-content: center;
  background: linear-gradient(45deg, #00ffff20, #ff00ff20);
  border-radius: 10px;
  font-size: 1.2rem;
  color: #00ffff;
}

.certificate-icon i {
  filter: drop-shadow(0 0 8px rgba(0, 255, 255, 0.5));
}

.certificate-content h3 {
  color: #ffff;
  font-size: 1.2rem;
  font-weight: 500;
  margin: 0;
  flex-grow: 1;
}

.certificate-info {
  margin-bottom: 1.5rem;
}

.info-row {
  display: flex;
  align-items: center;
  gap: 0.75rem;
  padding: 0.5rem 0;
  color: rgba(255, 255, 255, 0.7);
  font-size: 0.9rem;
}

.info-row i {
  color: #00ffff;
  font-size: 1rem;
  width: 20px;
  text-align: center;
}

.info-row.verified {
  color: #00ff95;
}

.info-row.verified i {
  color: #00ff95;
}
```

```
.certificate-actions {
  display: grid;
  grid-template-columns: 1fr 1fr;
  gap: 0.75rem;
  margin-top: 1rem;
  width: 100%;
}

.btn-action {
  background: transparent;
  border: none;
  padding: 0;
  width: 100%;
  min-width: 0;
}

.btn-action a {
  display: flex;
  align-items: center;
  justify-content: center;
  gap: 0.5rem;
  padding: 0.75rem 0.5rem;
  color: #00ffff;
  background: rgba(0, 255, 255, 0.1);
  border: 1px solid rgba(0, 255, 255, 0.2);
  border-radius: 8px;
  font-size: 0.9rem;
  font-weight: 500;
  text-decoration: none;
  transition: all 0.2s ease;
  width: 100%;
  white-space: nowrap;
  overflow: hidden;
  text-overflow: ellipsis;
}

.btn-action span {
  overflow: hidden;
  text-overflow: ellipsis;
}

.btn-action:hover a {
  background: rgba(0, 255, 255, 0.15);
  border-color: rgba(0, 255, 255, 0.3);
  transform: translateY(-2px);
}

.btn-action.download a {
  color: #00ffff;
  background: rgba(0, 255, 255, 0.1);
  border-color: rgba(0, 255, 255, 0.2);
}

.btn-action.view a {
  color: #ff00ff;
  background: rgba(255, 0, 255, 0.1);
  border-color: rgba(255, 0, 255, 0.2);
}

.btn-action.download:hover a {
  background: rgba(0, 255, 255, 0.15);
  border-color: rgba(0, 255, 255, 0.3);
}
```

```
.btn-action.view:hover a {
    background: rgba(255, 0, 255, 0.15);
    border-color: rgba(255, 0, 255, 0.3);
}

.no-certificates {
    grid-column: 1 / -1;
    text-align: center;
    padding: 3rem 2rem;
    background: rgba(22, 22, 22, 0.95);
    border-radius: 16px;
    border: 1px solid rgba(0, 255, 255, 0.1);
    position: relative;
}

.no-certificates::before {
    content: "";
    position: absolute;
    inset: 0;
    border-radius: 16px;
    padding: 1px;
    background: linear-gradient(45deg, #00ffff20, #ff00ff20);
    -webkit-mask: linear-gradient(#fff 0 0) content-box, linear-gradient(#fff 0 0);
    mask: linear-gradient(#fff 0 0) content-box, linear-gradient(#fff 0 0);
    -webkit-mask-composite: xor;
    mask-composite: exclude;
    pointer-events: none;
}

.no-cert-icon {
    font-size: 3rem;
    color: #00ffff;
    margin-bottom: 1.5rem;
    filter: drop-shadow(0 0 8px rgba(0, 255, 255, 0.5));
}

.no-certificates p {
    font-size: 1rem;
    margin-bottom: 1.5rem;
    color: rgba(255, 255, 255, 0.8);
}

.no-certificates .btn-action {
    width: auto;
    display: inline-block;
}

.no-certificates .btn-action a {
    padding: 0.75rem 1.5rem;
}

@media (max-width: 768px) {
    .page-title {
        font-size: 2rem;
        margin: 1rem 0 1.5rem;
    }
}

.certificates-grid {
    grid-template-columns: 1fr;
    gap: 1rem;
    padding: 0.5rem;
}
```

```

.certificate-card {
    padding: 1rem;
}

.certificate-actions {
    grid-template-columns: 1fr;
    gap: 0.5rem;
}

.btn-action a {
    padding: 0.75rem;
    font-size: 0.85rem;
}

.no-certificates {
    padding: 2rem 1rem;
}
}

</style>
{% endblock %}

```

view_certificate.html

```

{% extends 'bustComProj/base.html' %}
{% load static %}

{% block title %}Просмотр сертификата{% endblock %}

{% block content %}
<div class="container fade-in">
    <div class="certificate-view">
        <div class="certificate-header">
            <h1>Сертификат об окончании курса</h1>
            <p class="certificate-number">№ {{ certificate.id }}</p>
        </div>

        <div class="certificate-body">
            <p class="certificate-text">
                Настоящим подтверждается, что
                <span class="certificate-user">{{ certificate.user.get_full_name }}</span>
                успешно завершил(а) курс
                <span class="certificate-course">{{ certificate.course.name }}</span>
            </p>

            <p class="certificate-date">
                Дата выдачи: {{ certificate.issued_at|date:"d.m.Y" }}
            </p>
        </div>

        <div class="certificate-footer">
            <div class="certificate-actions">
                <a href="{% url 'download_certificate' certificate.id %}" class="btn-neon">
                    <i class="fas fa-download"></i> Скачать сертификат
                </a>
                <a href="{% url 'my_certificates' %}" class="btn-neon">
                    <i class="fas fa-arrow-left"></i> Назад к сертификатам
                </a>
            </div>
        </div>
    </div>
</div>

```

```
<style>
.certIFICATE-view {
    max-width: 800px;
    margin: 2rem auto;
    padding: 3rem;
    background: rgba(18, 18, 18, 0.9);
    border: 2px solid #00ffff;
    border-radius: 15px;
    box-shadow: 0 0 40px rgba(0, 255, 255, 0.1);
    color: #fff;
}

.certIFICATE-header {
    text-align: center;
    margin-bottom: 3rem;
    border-bottom: 2px solid #00ffff;
    padding-bottom: 1rem;
}

.certIFICATE-header h1 {
    color: #00ffff;
    text-shadow: 0 0 10px rgba(0, 255, 255, 0.3);
    font-size: 2.5rem;
    margin-bottom: 1rem;
}

.certIFICATE-number {
    color: rgba(255, 255, 255, 0.7);
    font-size: 1.1rem;
}

.certIFICATE-body {
    text-align: center;
    margin: 3rem 0;
    font-size: 1.2rem;
    line-height: 1.8;
}

.certIFICATE-text {
    color: rgba(255, 255, 255, 0.9);
}

.certIFICATE-user,
.certIFICATE-course {
    color: #00ffff;
    font-weight: 600;
    font-size: 1.3rem;
    text-shadow: 0 0 10px rgba(0, 255, 255, 0.3);
}

.certIFICATE-date {
    margin-top: 2rem;
    color: rgba(255, 255, 255, 0.7);
}

.certIFICATE-footer {
    margin-top: 3rem;
    text-align: center;
    border-top: 2px solid rgba(0, 255, 255, 0.2);
    padding-top: 2rem;
}
```

```
.certificate-actions {
  display: flex;
  gap: 1rem;
  justify-content: center;
}

.btn-neon {
  display: inline-flex;
  align-items: center;
  gap: 0.5rem;
  padding: 0.8rem 1.5rem;
  background: transparent;
  color: #00ffff;
  border: 1px solid #00ffff;
  border-radius: 25px;
  font-weight: 600;
  text-transform: uppercase;
  letter-spacing: 1px;
  text-decoration: none;
  transition: all 0.3s ease;
  position: relative;
  overflow: hidden;
  box-shadow: 0 0 10px rgba(0, 255, 255, 0.2);
}

.btn-neon:hover {
  background: rgba(0, 255, 255, 0.1);
  color: #fff;
  box-shadow: 0 0 20px rgba(0, 255, 255, 0.4);
  text-decoration: none;
}

.btn-neon:active {
  transform: scale(0.98);
}

@media (max-width: 768px) {
  .certificate-view {
    padding: 1.5rem;
    margin: 1rem;
  }

  .certificate-header h1 {
    font-size: 2rem;
  }

  .certificate-body {
    font-size: 1.1rem;
  }

  .certificate-actions {
    flex-direction: column;
  }

  .btn-neon {
    width: 100%;
    justify-content: center;
  }
}

</style>
{% endblock %}
```

ПРИЛОЖЕНИЕ В. Программа испытаний

АННОТАЦИЯ

В данном программном документе приведены:

В разделе «Объект испытаний» указан объект испытания.

В разделе «Цель испытаний» указана цель проведения испытаний.

В разделе «Требования к программе» указаны требования, подлежащие проверке во время испытаний и заданные в техническом задании на программу.

В данном программном документе, в разделе «Требования к программной документации» указан состав программной документации, предъявляемый на испытания.

В разделе «Средства и порядок испытаний» указаны технические и программные средства, используемые во время испытаний, а также порядок проведения испытаний.

В разделе «Методы испытаний» приведено описание используемых методов испытаний.

В разделе «Тестовые приёмы» приведены таблицы с ожидаемыми результатами.

В разделе «Результаты тестовых испытаний» приведены таблицы с фактическими результатами тестовых испытаний.

В разделе «Протокол тестовых испытаний» содержит итоговую информацию по проведенным испытаниям и заключение о соответствии требованиям.

1. ОБЪЕКТ ИСПЫТАНИЙ

1.1. Назначение системы

Описание: Система BustCom представляет собой комплексное решение для автоматизации процессов корпоративного обучения. Основное назначение системы заключается в организации, управлении и мониторинге учебного процесса в рамках корпоративного обучения. Система обеспечивает автоматизацию следующих задач:

Управление учебными курсами и материалами

Отслеживание прогресса обучения сотрудников

Проведение тестирования и оценки знаний

Управление сертификацией

Организация коммуникации между участниками процесса обучения

Формирование отчетности по результатам обучения

1.2. Цели создания

Описание: Основные цели создания системы BustCom:

Сокращение времени на организацию и проведение учебных мероприятий на 40%

Уменьшение затрат на администрирование процесса обучения на 35%

Повышение качества обучения за счет автоматизированного контроля знаний

Снижение количества ошибок в учетной документации на 90%

Обеспечение прозрачности процесса обучения для всех участников

Оптимизация времени преподавателей на подготовку и проведение занятий

Улучшение доступности учебных материалов для сотрудников

1.3. Область применения

Описание: Система BustCom предназначена для применения в следующих областях:

Корпоративное обучение в компаниях различных отраслей

Профессиональная переподготовка персонала

Внутреннее обучение сотрудников

Управление компетенциями персонала

Организация дистанционного обучения

Проведение аттестации сотрудников

Мониторинг эффективности обучения

Система может быть использована как в небольших компаниях, так и в крупных корпорациях с разветвленной структурой подразделений.

2. ЦЕЛЬ ИСПЫТАНИЙ

Целью проведения испытаний является проверка соответствия разработанной программы требованиям, указанным в документе «Техническое задание».

3. ТРЕБОВАНИЯ К ПРОГРАММЕ

3.1. Схема тестирования приложения

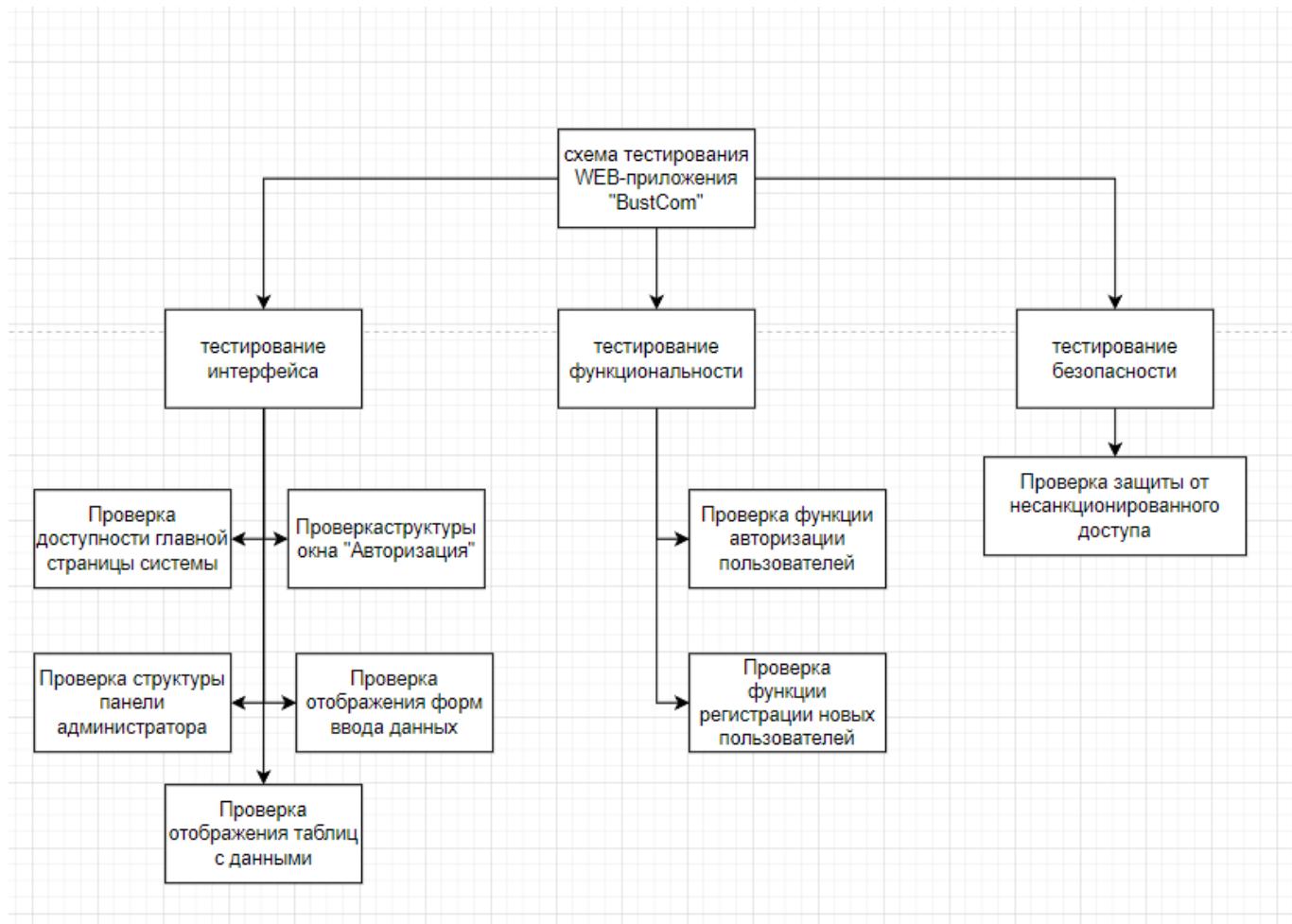


Рисунок 1 - Схема тестирования приложения

3.2. Требования, подлежащие проверке в приложении

Тестирование интерфейса:

TC1 - Проверка доступности главной страницы системы

TC2 - Проверка структуры окна "Авторизация"

TC3 - Проверка структуры панели администратора

TC4 - Проверка отображения форм ввода данных

TC5 - Проверка отображения таблиц с данными

Тестирование функций:

TC6 - Проверка функции авторизации пользователей

TC7 - Проверка функции регистрации новых пользователей

Тестирование безопасности:

ТС8 - Проверка защиты от несанкционированного доступа

4. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

Состав технических средств, используемых для проведения тестирования программного комплекса представлен в таблице 1.

Таблица 1 - Технические характеристики

№	Тип средства	Название средства
1	2	3
Ноутбук Lenovo Victum 16		
1	Размер экрана	14,0
2	Разрешение экрана	2560x1440
3	Линейка процессора	Ryzen 7 7700H
4	Количество ядер процессора	8
5	Оперативная память	16 ГБ
6	Тип видеокарты	встроенная
7	Видеокарта	Geforce RTX 3060
8	Конфигурация накопителей	SSD
9	Общий объем всех накопителей	1 ТБ
10	Операционная система	Windows 11 Pro

Состав программных средств, используемых для проведения тестирования программного комплекса представлен в таблице 2.

Таблица 2 - Программные средства разработки ПО

№	Тип средства	Название средства	Назначение
1	2	3	4
1	Браузер	Google Chrome ver. 111.0.5563.65	Организация взаимодействия пользователя и программы.
2	Инструментальные средства разработки программного решения	Visual Studio Code ver. 1.95.3	Разработка программного обеспечения
3	СУБД	SQLite	Разработка базы данных

5. МЕТОДЫ ИСПЫТАНИЙ

- По формальности тестирования.

Тестирование по тестам – тестирование по предварительно написанным тест кейсам.

- По исполнению кода.

Динамическое тестирование – во время тестирования код исполняется.

- По целям.

Функциональное тестирование - направлено на проверку того, какие функции программного обеспечения реализованы и насколько они соответствуют требованиям.

Нефункциональное тестирование направлено на проверку качественных характеристик ПО, таких как производительность, удобство использования, надежность и безопасность.

- По степени автоматизации.

Ручное – без использования дополнительных программных средств.

- По позитивности сценария.

Позитивным – проверка ПО на соответствие ожидаемому поведению.

Негативным – проверяет, будет ли ПО работать в случае, когда поведение пользователя отличается от ожидаемого.

- По знанию системы.

Тестирование «черного ящика» – тестирование без доступа к коду продукта.

- По разработке тестовых испытаний.

На основе требований – требование было определено до начала тестирования.

6. ТЕСТОВЫЕ ПРИЁМЫ

Все тестовые данные для проведения тестирования WEB-приложения «BustCom» представлены в таблицах 3-9.

Таблица 3 - ТС1 - Проверка доступности главной страницы системы

TC_MAIN_PAGE	Приоритет теста	Название тестирования	Резюме испытания	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Предпосылки	Постусловия	Статус (Pass/ Fail)	Комментарии
1	Высокий	Проверка доступности главной страницы	Проверка корректного отображения главной страницы системы при различных условиях	Запустить веб-браузер Перейти на главную страницу системы Проверить загрузку всех элементов страницы	Стандартный доступ: URL: http://localhost:8000/ Браузер: Chrome Скорость интернета: Нормальная	Стандартный доступ: Страница загружается полностью Отображается логотип системы Отображается навигационное меню Отображается футер Все элементы страницы корректно отображаются		Система развернута и доступна Сервер приложения работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования	Пользователь находится на главной странице Все элементы страницы загружены Система готова к дальнейшему использованию		
2					Медленное соединение: URL: http://localhost:8000/ Браузер: Chrome Скорость интернета: Медленная (2G)	Медленное соединение: Страница загружается постепенно Отображается индикатор загрузки Все		Система развернута и доступна Сервер приложения работает База данных доступна	Пользователь находится на главной странице Все элементы страницы загружены Система готова к		

					элементы загружаются в правильном порядке Функциональность сохраняется		Установлены различные браузеры Доступны различные устройства для тестирования	дальнейшему и спользованию		
3				Разные браузеры: URL: http://localhost:8000/ Браузеры: Firefox, Safari, Edge Скорость интернета: Нормальная	Разные браузеры: Страница корректно отображается во всех браузерах Функциональность работает одинаково Стили применяются корректно		Система развернута и доступна Сервер приложения работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования	Пользователь находится на главной странице Все элементы страницы загружены Система готова к дальнейшему и спользованию		
4				Мобильное устройство: URL: http://localhost:8000/	Мобильное устройство: Страница адаптируется под мобильный		Система развернута и доступна Сервер приложения	Пользователь находится на главной странице Все элементы		

				Браузер: Chrome Mobile Разрешение экрана: 375x667	экран Все элементы доступны для тач-интерфейса Меню преобразуется в "бургер"		я работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирован ия	страницы загружены Система готова к дальнейшему и спользованию		
5				Планшет: URL: http://localhost:8000/ Браузер: Safari Разрешение экрана: 768x1024	Планшет: Страница адаптируется под размер экрана планшета Все элементы масштабируются корректно Функциональность доступна		Система развернута и доступна Сервер приложения работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования	Пользователь находится на главной странице Все элементы страницы загружены Система готова к дальнейшему и спользованию		
6				Неверный URL:	Неверный URL:		Система	Пользователь н		

				<p>URL: http://localhost:8000/nonexistent/ Браузер: Chrome Скорость интернета: Нормальная</p>	<p>Отображается страница 404 Предлагается вернуться на главную Сохраняется навигация</p>	<p>развернута и доступна Сервер приложени я работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирован ия</p>	<p>аходится на главной странице Все элементы страницы загружены Система готова к дальнейшему и спользованию</p>	
7				<p>Отключенный JavaScript: URL: http://localhost:8000/ Браузер: Chrome JavaScript: Отключен</p>	<p>Отключенный JavaScript: Базовая функциональнос ть работает Отображается предупреждение о необходимости включить JavaScript Основной контен т доступен</p>	<p>Система развернута и доступна Сервер приложени я работает База данных доступна Установлены различные браузеры Доступны различные устройства для</p>	<p>Пользователь н аходится на главной странице Все элементы страницы загружены Система готова к дальнейшему и спользованию</p>	

							тестирован ия			
8					Отключенные стили: URL: http://localhost:8000/ Браузер: Chrome CSS: Отключен	Отключенные стили: Контент отображается в базовом виде Сохраняется структура страницы Функциональность не нарушена	Система развернута и доступна Сервер приложения работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования	Пользователь находится на главной странице Все элементы страницы загружены Система готова к дальнейшему использованию		

Таблица 4 -- ТС2 - Проверка структуры окна "Авторизация":

TC_AUTH_WINDOW	Приоритет теста	Название тестирования	Резюме испытания	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Предпосылки	Постулаты	Статус (Pass/Fail)	Комментарии
1	Высокий	Проверка структуры	Проверка корректного отображения	Запустить веб-браузер Перейти	Стандартное отображение: URL: http://localhost:8000	Стандартное отображение: Форма отображается по	Система развернута и доступна		Пользователь либо авторизован, либо		

		окна авторизации	ния и функционирования всех элементов окна авторизации	на страницу авторизации Проверить наличие и функционирование всех элементов формы	/login/ Браузер: Chrome Разрешение экрана: 1920x1080	центру экрана Все поля и кнопки видны полностью Стили применяются корректно	Сервер приложения работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования	видит сообщение об ошибке Форма готова к повторному вводу данных Состояние системы соответствует результату теста	
2				Мобильное отображение: URL: http://localhost:8000 /login/ Браузер: Chrome Mobile Разрешение экрана: 375x667	Мобильное отображение: Форма адаптируется под размер экрана Все элементы доступны для тач-интерфейса Кнопки имеют достаточный размер для нажатия	Система развернута и доступна Сервер приложения работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования	Пользователь либо авторизован, либо видит сообщение об ошибке Форма готова к повторному вводу данных Состояние системы соответствует результату		

						ования		у теста		
3					Разные браузеры: URL: http://localhost:8000 /login/ Браузеры: Firefox, Safari, Edge Разрешение экрана: 1920x1080	Разные браузеры: Форма отображается одинаково во всех браузерах Стили применяются корректно Функциональность идентична	Система развернут а и доступна Сервер пр иложения работает База данных доступна Установле ны различные браузеры Доступны различные устройств а для тестиру ования		Пользователю либо авторизован, либо видит сообщен ие об ошибке Форма готова к повторному вводу данных Состояние системы соответст вует результату теста	
4					Ввод корректных данных: Email: user@example.com Пароль: Password123!	Ввод корректных данных: Форма принимает данные Происходит успешная авториза ция Перенаправление на главную страницу	Система развернут а и доступна Сервер пр иложения работает База данных доступна Установле ны различные		Пользователю либо авторизован, либо видит сообщен ие об ошибке Форма готова к повторному вводу данных	

						браузеры Доступны различные устройств а для тестир ования		Состояни е системы соответст вует результат у теста	
5					Ввод некорректного email: Email: invalid.email Пароль: Password123!	Ввод некорректного email: Отображается сообщение об ошибке Поле email подсвечивает ся красным Форма не отправляется	Система развернут а и доступна Сервер пр иложения работает База данных доступна Установле ны различные браузеры Доступны различные устройств а для тестир ования	Пользова тель либо авторизо ван, либо видит сообщен ие об ошибке Форма го това к повторно му вводу данных Состояни е системы соответст вует результат у теста	
6					Ввод короткого пароля: Email: user@example.com Пароль: 123	Ввод короткого пароля: Отображается сообщение о минимальной длин е Поле пароля	Система развернут а и доступна Сервер пр иложения работает	Пользова тель либо авторизо ван, либо видит сообщен ие об	

					подсвечивается красным Форма не отправляется	База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования		ошибке Форма готова к повторному вводу данных Состояние системы соответствует результату теста	
7					Пустые поля: Email: (пусто) Пароль: (пусто)	Пустые поля: Отображается сообщение "Заполните все поля" Все поля подсвечиваются красным Форма не отправляется	Система развернута и доступна Сервер приложения работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования	Пользователь либо авторизован, либо видит сообщение об ошибке Форма готова к повторному вводу данных Состояние системы соответствует результату теста	
8					Специальные	Специальные	Система	Пользова	

				<p>символы: Email: user@example.com Пароль: !@#\$%^&*</p>	<p>символы: Форма принимает данные Происходит успешная авторизация Специальные символы обрабаты ваются корректно</p>	<p>развернут а и доступна Сервер пр иложения работает База данных доступна Установле ны различные браузеры Доступны различные устройств а для тестир ования</p>	<p>тель либо авторизо ван, либо видит сообщен ие об ошибке Форма го това к повторно му вводу данных Состояни е системы соответст вует результат у теста</p>	
9				<p>Пробелы в начале/конце: Email: " user@example.co m " Пароль: " Password123! "</p>	<p>Пробелы в начале/конце: Пробелы автоматически удаляются Форма принимает очищенные данные Происходит успешная авторизация</p>	<p>Система развернут а и доступна Сервер пр иложения работает База данных доступна Установле ны различные браузеры Доступны различные</p>	<p>Пользова тель либо авторизо ван, либо видит сообщен ие об ошибке Форма го това к повторно му вводу данных Состояни е системы</p>	

							устройств а для тестир ования		соответст вует результат у теста		
10					Максимальная длина полей: Email: ver y.long.email.address @example.com Пароль: VeryLongPassword1 23456789!@#	Максимальная длина полей: Форма принимает данные Текст не обрезается Происходит успешная авторизация	Система развернут а и доступна Сервер пр иложения работает База данных доступна Установле ны различные браузеры Доступны различные устройств а для тестир ования		Пользова тель либо авторизо ван, либо видит сообщен ие об ошибке Форма го това к повторно му вводу данных Состояни е системы соответст вует результат у теста		

Таблица 5 – ТС3 - Проверка структуры панели администратора

TC_ADMI N_PANEL	Прио ритет теста	Названи е тестиро вания	Резюме испытани я	Шаги тестировани я	Данные тестирования	Ожидаемый результат	Факти ческий результ ат	Предпос ылки	Постусловия	Ст ату с (Pa ss/ Fai l)	Комме нтарии
1	Высо кое	Провер	Проверка	Войти в	Основная панель	Основная пане		Система	Панель		

	кий	ка панели администратора	корректного отображения и функционирования панели администратора	систему как администратор Проверить отображение панели Проверить функциональность всех элементов	управления: URL: http://localhost:8000/admin/dashboard/ Роль: Администратор Разрешение экрана: 1920x1080 Браузер: Chrome	ль управления: Отображается главное меню Видны основные метрики Доступны быстрые действия Навигация работает корректно		развернута и доступна Сервер приложения работает База данных доступна Создан аккаунт администратора Настроены права доступа	администратора отображается корректно Все функции работают Состояние системы соответствует результату теста	
2				Управление пользователями: URL: http://localhost:8000/admin/users/ Действия: Просмотр списка пользователей Добавление нового пользователя Редактирование пользователя Удаление пользователя	Управление пользователями: Таблица пользователей отображается Формы добавления/редактирования работают Фильтрация и поиск функционируют Действия		Система развернута и доступна Сервер приложения работает База данных доступна Создан аккаунт	Панель администратора отображается корректно Все функции работают Состояние системы соответствует результату теста		

					выполняются корректно		администратора Настроены права доступа			
3					Управление курсами: URL: http://localhost:8000/admin/courses/ Действия: Просмотр списка курсов Создание нового курса Редактирование курса Удаление курса	Управление курсами: Список курсов отображается Формы создания/редактирования работают Загрузка материалов функционирует Управление доступом работает		Система развернута и доступна Сервер приложения работает База данных доступна Создан аккаунт администратора Настроены права доступа	Панель администратора отображается корректно Все функции работают Состояние системы соответствует результату теста	
4					Управление материалами: URL: http://localhost:8000/admin/materials/ Действия:	Управление материалами: Список материалов отображается Загрузка файлов работает		Система развернута и доступна Сервер приложе	Панель администратора отображается корректно Все функции работают Состояние	

				<p>Просмотр списка материалов Загрузка новых материалов Редактирование материалов Удаление материалов</p>	<p>Редактирование доступно Удаление функционирует</p>		<p>ния работает База данных доступна Создан аккаунт администратора Настроены права доступа</p>	<p>системы соответствует результату теста</p>
5				<p>Управление тестами: URL: http://localhost:8000/admin/tests/ Действия: Просмотр списка тестов Создание нового теста Редактирование теста Удаление теста</p>	<p>Управление тестами: Список тестов отображается Создание тестов работает Редактирование доступно Удаление функционирует</p>		<p>Система развернута и доступна Сервер приложения работает База данных доступна Создан аккаунт администратора Настроены права доступа</p>	<p>Панель администратора отображается корректно Все функции работают Состояние системы соответствует результату теста</p>

6				<p>Настройки системы: URL: http://localhost:8000/admin/settings/</p> <p>Действия:</p> <ul style="list-style-type: none"> Настройка параметров системы Управление правами доступа Настройка уведомлений Управление резервными копиями 	<p>Настройки системы: Формы настроек работают Сохранение настроек функционирует Права доступа применяются Резервное копирование работает</p>		<p>Система развернута и доступна Сервер приложения работает База данных доступна Создан аккаунт администратора Настроены права доступа</p>	<p>Панель администратора отображается корректно Все функции работают Состояние системы соответствует результату теста</p>	
7				<p>Управление сертификатами: URL: http://localhost:8000/admin/certificates/</p> <p>Действия:</p> <ul style="list-style-type: none"> Просмотр списка сертификатов Создание сертификата Редактирование сертификата Удаление сертификата 	<p>Управление сертификатами: Список сертификатов отображается Создание работает Редактирование доступно Удаление функционирует</p>		<p>Система развернута и доступна Сервер приложения работает База данных доступна</p>	<p>Панель администратора отображается корректно Все функции работают Состояние системы соответствует результату теста</p>	

Таблица 6 – ТС4 - Проверка отображения форм ввода данных

TC_INPUT_FORMS	Приоритет теста	Название тестирования	Резюме испытания	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Предпосылки	Постусловия	Статус (Pass/ Fail)	Комментарии
1	Высокий	Проверка форм ввода данных	Проверка корректного отображения и функционирования всех форм ввода данных в системе	Открыть различные формы ввода Проверить отображение элементов Проверить валидацию и функциональность	Форма регистрации: URL: http://localhost:8000/register/ Поля: Имя Фамилия Email Пароль Подтверждение пароля Роль	Форма регистрации: Все поля отображаются Валидация работает Сообщения об ошибках видны Кнопка отправки активна		Система развернута и доступна Сервер приложения работает База данных доступна Созданы тестовые	Формы отображаются корректно Валидация работает Данные сохраняются Состояние системы соответствует результату теста		

							данные Настро ны права доступа			
2					Форма создания курса: URL: http://localhost:8000/courses/create/ Поля: Название курса Описание Длительность Уровень сложности Категория Цена	Форма создания курса: Поля заполняются Валидация работает Загрузка файлов доступна Сохранение функционирует	Система развернута и доступна Сервер приложения работает База данных доступна Созданы тестовые данные Настройны права доступа	Формы отображаются корректно Валидация работает Данные сохраняются Состояние системы соответствует результату теста		
3					Форма создания лекции: URL: http://localhost:8000/lectures/create/ Поля:	Форма создания лекции: Поля отображаются Валидация	Система развернута и доступна	Формы отображаются корректно		

4				<p>Название лекции Описание Курс Длительность Материалы Тесты</p>	<p>работает Загрузка материалов доступна Связь с курсом работает</p>	<p>а Сервер приложения работает База данных доступна Созданы тестовые данные Настроены права доступа</p>	<p>Валидация работает Данные сохраняются Состояние системы соответствует результату теста</p>	
				<p>Форма создания теста: URL: http://localhost:8000/tests/create/ Поля: Название теста Описание Время на выполнение Проходной балл Вопросы Настройки</p>	<p>Форма создания теста: Поля заполняются Валидация работает Добавление вопросов доступно Настройки сохраняются</p>	<p>Система развернута и доступна Сервер приложения работает База данных доступна Созданы</p>	<p>Формы отображаются корректно Валидация работает Данные сохраняются Состояние системы соответствует</p>	

							ы тестовы е данные Настро ены права доступа	ствует результату теста
5				Форма профиля пользователя: URL: http://localhost:8000/profi le/edit/ Поля: Фото профиля Имя Фамилия Email Телефон О себе	Форма профиля пользо вателя: Поля отображаются Загрузка фото работает Валидация фун кционирует Сохранение доступно	Систем а разверн ута и доступн а Сервер прилож ения работае т База данных доступн а Создан ы тестовы е данные Настро ены права доступа	Формы отобра жаются коррект но Валидац ия работае т Данные сохраня ются Состоян ие системы соответ ствует результата ту теста	
6				Форма заявки на обучение:	Форма заявки на обучение:		Систем а	Формы отобра

						URL: http://localhost:8000/training/request/ Поля: ФИО Email Курс Компания Должность Комментарий	Поля заполняются Валидация работает Выбор курса доступен Отправка функционирует	развернута и доступна Сервер приложения работает База данных доступна Созданы тестовые данные Настроены права доступа	аются корректно Валидация работает Данные сохраняются Состояние системы соответствует результату теста	
--	--	--	--	--	--	---	--	---	---	--

Таблица 7 – ТС5 - Проверка отображения таблиц и списков:

TC_TABLES_LISTS	Приоритет теста	Название тестированния	Резюме испытания	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Предпосылки	Постусловия	Статус (Pass/ Fail)	Комментарии
1	Высокий	Проверка отображения таблиц и списков	Проверка корректного отображения и	Открыть различные таблицы и списки Проверить от	Список курсов: URL: http://localhost:8000/courses/ Элементы:	Список курсов: Данные отображаются Сортировка работает		Система развернута и доступна	Таблицы и списки отображаются корректно		

			функционирования всех таблиц и списков в системе	отображение данных Проверить функциональность управления	Название курса Описание Преподаватель Количество сотрудников Рейтинг Статус	Фильтрация доступна Пагинация функционирует	на Сервер приложения работает База данных доступна Созданы тестовые данные Настроены права доступа	Функциональность работает Данные актуальны Состояние системы соответствует результату теста	
2				Таблица сотрудников: URL: http://localhost:8000/employee/ Элементы: ФИО Email Группа Прогресс Оценки Статус	Таблица сотрудников: Информация отображается Поиск работает Экспорт доступен Действия функционируют	Система развернута и доступна на Сервер приложении работает База данных доступна Созданы	Таблицы и списки отображаются корректно Функциональность работает Данные актуальны Состояние системы соответствует результату теста		

							ы тестовы е данные Настро ены права д оступа			
3				Список лекций: URL: http://localhost:8000/lectures/ Элементы: Название Курс Длительность Материалы Тесты Статус	Список лекций: Элементы отобраются Группировка работает Фильтры доступны Навигация функционирует		Систем а разверн ута и доступ на Сервер прилож ения работае т База данных доступ на Создан ы тестовы е данные Настро ены права д оступа	Таблицы и списки отобража ются корректно Функцион альность работает Данные актуальны Состояние системы соответств ует результату теста		
4				Таблица тестов: URL:	Таблица тестов: Данные отобра		Систем а	Таблицы и списки		

5				<p>http://localhost:8000/tests/</p> <p>Элементы:</p> <p>Название Курс Время Вопросы Проходной балл Статус</p>	<p>жаются Сортировка работает Статистика доступна Управление функционирует</p>		<p>развернута и доступна Сервер приложения работает База данных доступна Созданы тестовые данные Настройки права доступа</p>	<p>отображаются корректно Функциональность работает Данные актуальны Состояние системы соответствует результату теста</p>		

						доступ на Созданы тестовые данные Настроены права доступа	ует результату теста				
6					Таблица прогресса: URL: http://localhost:8000/progress/ Элементы: Сотрудник Курс Лекции Тесты Оценки Сертификаты	Таблица прогресса: Прогресс отображается Фильтры работают Экспорт доступен Анализ функционирует		Система развернута и доступна на Сервере приложения работает База данных доступна Созданы тестовые данные Настроены права доступа	Таблицы и списки отображаются корректно Функциональность работает Данные актуальны Состояние системы соответствует результату теста		

7					<p>Список заявок: URL: http://localhost:800 0/requests/ Элементы: ID Тип Статус Дата Исполнитель Приоритет</p>	<p>Список заявок: Заявки отображаются Сортировка работает Статусы доступны Управление функционирует</p>		Система развернута и доступна Сервер приложения работает База данных доступна Созданы тестовые данные Настройки права доступа	Таблицы и списки отображаются корректно Функциональность работает Данные актуальны Состояние системы соответствует результату теста		

Таблица 8 – ТС6 - Проверка функции авторизации пользователей:

TC_AUTH_FUNCTION	Приоритет теста	Название тестирования	Резюме испытания	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Предпосылки	Постуловия	Статус (Pass/Fail)	Комментарии
1	Высокий	Проверка функции авториза	Проверка корректной	Проверить вход в систему	Вход с корректными данными:	Вход с корректными данными		Система развернута и доступна	Авторизация работает		

		ции пользова телей	работы функции авториза ции пользова телей в системе	Проверить валидацию данных Проверить обработку о шибок Проверить безопасност ь	URL: http://localhost:8 000/login/ Тестовые данные: Email: test@example.co m Пароль: Test123! Ожидаемый результат: Успешный вход	данными: Перенапра вление на главную Создание сессии Отображен ие профиля Доступ к функциям	Сервер прил ожения работает База данных доступна Созданы тестовые акк аунты Настроены п рава доступа	корректн о Безопасн ость обеспече на Сессии управля ются Состоян ие системы соответс твует результату теста	
2				Вход с неверным email: URL: http://localhost:8 000/login/ Тестовые данные: Email: wrong@example. com Пароль: Test123! Ожидаемый результат: Ошибка авторизации	Вход с неверным email: Сообщение об ошибке Сохранени е email Очистка пароля Возможнос ть повтора		Система развернута и доступна Сервер прил ожения работает База данных доступна Созданы тестовые акк аунты Настроены п рава доступа	Авториз ация работает корректн о Безопасн ость обеспече на Сессии управля ются Состоян ие системы соответс твует результату теста	

3				<p>Вход с неверным паролем: URL: http://localhost:8000/login/ Тестовые данные: Email: test@example.com Пароль: WrongPass123! Ожидаемый результат: Ошибка авторизации</p>	<p>Вход с неверным паролем: Сообщение об ошибке Сохранение email Очистка пароля Счетчик попыток</p>		<p>Система развернута и доступна Сервер приложения работает База данных доступна Созданы тестовые аккаунты Настроены права доступа</p>	<p>Авторизация работает корректно Безопасность обеспечена Сессии управляются Состояние системы соответствует результату теста</p>	
4				<p>Вход с пустыми полями: URL: http://localhost:8000/login/ Тестовые данные: Email: (пусто) Пароль: (пусто) Ожидаемый результат: Валидация полей</p>	<p>Вход с пустыми полями: Подсветка полей Сообщения об ошибках Предотвращение отправки Валидация в реальном времени</p>		<p>Система развернута и доступна Сервер приложения работает База данных доступна Созданы тестовые аккаунты Настроены права доступа</p>	<p>Авторизация работает корректно Безопасность обеспечена Сессии управляются Состояние системы</p>	

										соответствует результату теста		
--	--	--	--	--	--	--	--	--	--	--------------------------------	--	--

Таблица 9 – ТС7 - Проверка функции регистрации новых пользователей

TC_REGISTRATION_FUNCTION	Приоритет теста	Название тестированния	Резюме испытания	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Предпосылки	Постусловия	Статус (Pass/ Fail)	Комментарии
1	Высокий	Проверка функции регистрации новых пользователей	Проверка корректной работы функции регистрации новых пользователей в системе	Проверить форму регистрации Проверить валидацию данных Проверить создание аккаунта	Регистрация с корректными данными: URL: http://localhost:8000/register/ Тестовые данные: Имя: Иван Фамилия: Иванов Email: ivan@example.com Пароль: Test123! Подтверждение пароля: Test123! Роль: Сотрудник	Регистрация с корректными данными: Создание аккаунта Отправка email Перенаправление Сообщение об успехе		Система развернута и доступна Сервер приложения работает База данных доступна Настроена отправка email Настроены права доступа	Регистрация работает корректно Email отправляется Аккаунты создаются Состояние системы соответствует результату теста		
2					Регистрация	Регистрац		Система	Регистрация		

					с существующим email: URL: http://localhost:8000/register/ Тестовые данные: Имя: Петр Фамилия: Петров Email: test@example.com (существующий) Пароль: Test123! Подтверждение пароля: Test123! Роль: Сотрудник	ия с существующим email: Сообщение об ошибке Сохранение данных Подсветка поля Возможность исправления		развернута и доступна Сервер приложения работает База данных доступна Настроена отправка email Настроены права доступа	работает корректно Email отправляется Аккаунты создаются Состояние системы соответствует результату теста	
3					Регистрация с несовпадающими паролями: URL: http://localhost:8000/register/ Тестовые данные: Имя: Сергей Фамилия: Сергеев Email: sergey@example.com	Регистрация с несовпадающими паролями: Сообщение об ошибке Очистка паролей Подсветка полей Валидации		Система развернута и доступна Сервер приложения работает База данных доступна Настроена отправка email Настроены права	Регистрация работает корректно Email отправляется Аккаунты создаются Состояние системы соответствует результату теста	

					Пароль: Test123! Подтверждение пароля: Test456! Роль: Сотрудник	я в реально м времени		доступа			
4					Регистрация с некорректным email: URL: http://localhost:8 000/register/ Тестовые данные: Имя: Алексей Фамилия: Алексеев Email: invalid- email Пароль: Test123! Подтверждение пароля: Test123! Роль: Сотрудник	Регистрац ия с некоррект ным email: Сообщен ие об ошибке Подсветк а поля Валидац ия формата Подсказка формата		Система развернута и доступна Сервер при ложении работает База данных доступна Настроена отправка email Настроены права доступа	Регистрация работает корректно Email отправляет ся Аккаунты создаются Состояние системы соответству ет результату т еста		
5					Регистрация с пустыми полями: URL: http://localhost:8 000/register/ Тестовые	Регистрац ия с пустыми полями: Подсветк а полей Сообщен		Система развернута и доступна Сервер при ложении работает База	Регистрация работает корректно Email отправляет ся Аккаунты		

					данные: Имя: (пусто) Фамилия: (пусто) Email: (пусто) Пароль: (пусто) Подтверждение пароля: (пусто) Роль: (не выбрана)	ия об ошибках Предотвращение отправки Валидация в реальном времени	данных доступна Настроена отправка email Настроены права доступа	создаются Состояние системы соответствует результату теста		
6					Регистрация с недопустимыми символами: URL: http://localhost:8000/register/ Тестовые данные: Имя: Иван@#\$% Фамилия: Иван ов@#\$% Email: ivan@example.com Пароль: Test123! Подтверждение пароля: Test123! Роль: Сотрудник	Регистрация с недопустимыми символами: Сообщение об ошибке Подсветка полей Валидация символов Подсказка допустимых символов	Система развернута и доступна Сервер приложения работает База данных доступна Настроена отправка email Настроены права доступа	Регистрация работает корректно Email отправляется Аккаунты создаются Состояние системы соответствует результату теста		

6. РЕЗУЛЬТАТЫ ТЕСТОВЫХ ИСПЫТАНИЙ

Все фактические результаты проведения тестирования WEB-приложения «BustCom» представлены в таблицах 10-16.

Таблица 10 - ТС1 - Проверка доступности главной страницы системы

TC_MAIN_PAGE	Приоритет теста	Название тестирования	Резюме испытания	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Предпосылки	Постусловия	Статус (Pass/Fail)	Комментарии
1	Высокий	Проверка доступности главной страницы	Проверка корректного отображения главной страницы системы при различных условиях	Запустить веб-браузер Перейти на главную страницу системы Проверить загрузку всех элементов страницы	Стандартный доступ: URL: http://localhost:8000/ Браузер: Chrome Скорость интернета: Нормальная	Стандартный доступ: Страница загружается полностью Отображается логотип системы Отображается навигационное меню Отображается футер Все элементы страницы корректно отображаются	Стандартный доступ: Страница загружается полностью Отображается логотип системы Отображается навигационное меню Отображается футер Все элементы страницы корректно отображаются	Система развернута и доступна Сервер приложения работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования	Пользователь находится на главной странице Все элементы страницы загружены Система готова к дальнейшему использованию	Pas	
2					Медленное соединение:	Медленное соединение:	Медленное соединение:	Система развернута	Пользователь находится на	Pas	

3				<p>URL: http://localhost:8000/</p> <p>Браузер: Chrome</p> <p>Скорость интернета: Медленная (2G)</p>	<p>Страница загружается постепенно Отображается индикатор загрузки Все элементы загружаются в правильном порядке Функциональность сохраняется</p>	<p>Страница загружается постепенно Отображается индикатор загрузки Все элементы загружаются в правильном порядке Функциональность сохраняется</p>	<p>и доступна Сервер приложения работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования</p>	<p>главной странице Все элементы страницы загружены Система готова к дальнейшему использованию</p>	
				<p>Разные браузеры: URL: http://localhost:8000/</p> <p>Браузеры: Firefox, Safari, Edge</p> <p>Скорость интернета: Нормальная</p>	<p>Разные браузеры: Страница корректно отображается во всех браузерах Функциональность работает одинаково Стили применяются корректно</p>	<p>Разные браузеры: Страница корректно отображается во всех браузерах Функциональность работает одинаково Стили применяются корректно</p>	<p>Система развернута и доступна Сервер приложения работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования</p>	<p>Пользователь находится на главной странице Все элементы страницы загружены Система готова к дальнейшему использованию</p>	Pas s

4		Мобильное устройство: URL: http://localhost:8000/ Браузер: Chrome Mobile Разрешение экрана: 375x667	Мобильное устройство: Страница адаптируется под мобильный экран Все элементы доступны для тач-интерфейса Меню преобразуется в "бургер"	Мобильное устройство: Страница адаптируется под мобильный экран Все элементы доступны для тач-интерфейса Меню преобразуется в "бургер"	Система развернута и доступна Сервер приложения работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования	Пользователь находится на главной странице Все элементы страницы загружены Система готова к дальнейшему использованию	Pas s	
5		Планшет: URL: http://localhost:8000/ Браузер: Safari Разрешение экрана: 768x1024	Планшет: Страница адаптируется под размер экрана планшета Все элементы масштабируются корректно Функциональность доступна	Планшет: Страница адаптируется под размер экрана планшета Все элементы масштабируются корректно Функциональность доступна	Система развернута и доступна Сервер приложения работает База данных доступна Установлены различные браузеры Доступны различные браузеры	Пользователь находится на главной странице Все элементы страницы загружены Система готова к дальнейшему использованию	Pas s	

6								
7								

					JavaScript Основной контент доступен	JavaScript Основной контент доступен	ны различные браузеры Доступны различные устройства для тестирования	использовани ю		
8					Отключенные стили: Контент отображается в базовом виде Сохраняется структура страницы Функциональность не нарушена	Отключенные стили: Контент отображается в базовом виде Сохраняется структура страницы Функциональность не нарушена	Система развернута и доступна Сервер приложения работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования	Пользователь находится на главной странице Все элементы страницы загружены Система готова к дальнейшему использованию	Pas s	

Таблица 11 – ТС2 - Проверка структуры окна "Авторизация":

TC_AUTH_WINDOW	Приоритет теста	Название испытания	Резюме испытания	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Предположки	Постулаты	Статус (Pa)	Комментарии
----------------	-----------------	--------------------	------------------	-------------------	---------------------	---------------------	-----------------------	-------------	-----------	-------------	-------------

		я								ss/ Fail)	
1	Высокий	Проверка структуры окна авторизации	Проверка корректного отображения и функционирования всех элементов в окна авторизации	Запустить веб-браузер Перейти на страницу авторизации Проверить наличие и функционирование всех элементов в форме	Стандартное отображение: URL: http://localhost:8000/login/ Браузер: Chrome Разрешение экрана: 1920x1080	Стандартное отображение: Форма отображается по центру экрана Все поля и кнопки видны полностью Стили применяются корректно	Стандартное отображение: Форма отображается по центру экрана Все поля и кнопки видны полностью Стили применяются корректно	Система развернута и доступна Сервер приложения работает База данных доступна Установлены различные браузеры Доступные устройства соответствуют требованиям для тестирования	Пользователь либо авторизован, либо видит сообщение об ошибке Форма готова к повторному вводу данных Состояние системы соответствует результату тестирования	Pas	
2				Мобильное отображение: URL: http://localhost:8000/login/ Браузер: Chrome Mobile Разрешение экрана	Мобильное отображение: Форма адаптируется под размер экрана Все элементы доступны для тач-интерфейса	Мобильное отображение: Форма адаптируется под размер экрана Все элементы доступны для тач-интерфейса	Мобильное отображение: Форма адаптируется под размер экрана Все элементы доступны для тач-интерфейса	Система развернута и доступна Сервер приложения работает	Пользователь либо авторизован, либо видит сообщение об ошибке	Pas	

				a: 375x667	Кнопки имеют достаточный размер для нажатия	Кнопки имеют достаточный размер для нажатия	База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования
3				Разные браузеры: URL: http://localhost:8000/login/ Браузеры: Firefox, Safari, Edge Разрешение экрана: 1920x1080	Разные браузеры: Форма отображается одинаково во всех браузерах Стили применяются корректно Функциональность идентична	Разные браузеры: Форма развернута и отображается одинаково во всех браузерах Стили применяются корректно Функциональность идентична	Система авторизована, либо видит сообщение об ошибке Форма готова к повторному вводу данных Состояние системы соответствует результату теста

						е устройст ва для тести рования	твует результату теста	
4		Ввод корректных данных: Email: user@example.co m Пароль: Password123!	Ввод корректных данных: Форма принимает данные Происходит успешная авториза ция Перенаправление на главную страницу	Ввод корректных данных: Форма принимает данные Происходит успешная авториза ция Перенаправление на главную страницу	Система развернута и доступна Сервер приложения работает База данных доступна Установлены различны е браузеры Доступны различные устройства для тести рования	Пользователь либо авторизован, либо видит сообщение об ошибке Форма готова к повторному вводу данных Состояние системы соответствует результату теста	Pas s	
5		Ввод некорректного email: Email: invalid.email	Ввод некорректного email: Отображается сообщение об	Ввод некорректного email: Отображается сообщение об	Система развернута и доступна Сервер приложения	Пользователь либо авторизован, либо	Pas s	

6				<p>Пароль: Password123!</p>	<p>ошибке Поле email подсвечива ется красным Форма не отправляется</p>	<p>ошибке Поле email подсвечива ется красным Форма не отправляется</p>	<p>риложен ия работает База данных доступна Установл ены различны е браузеры Доступн ы различны е устройст ва для тести рования</p>	<p>о видит сообщен ие об ошибке Форма г отова к повторн ому вводу данных Состоян ие системы соответс твует результату теста</p>

							Доступны различные устройства для тестирования	ие системы соответствует результату теста	
7				Пустые поля: Email: (пусто) Пароль: (пусто)	Пустые поля: Отображается сообщение "Заполните все поля" Все поля подсвечиваются красным Форма не отправляется	Пустые поля: Отображается сообщение "Заполните все поля" Все поля подсвечиваются красным Форма не отправляется	Система развернута и доступна Сервер приложения работает База данных доступна Установлены различные браузеры Доступны различные системы	Пользователь либо авторизован, либо видит сообщение об ошибке Форма готова к повторному вводу данных Состояние системы соответствует результату теста	Pas
8				Специальные символы:	Специальные символы:	Специальные символы:	Система развернута	Пользователь	Pas

					Email: user@example.co m Пароль: !@#\$%&*	Форма принимает данные Происходит успешная авторизация Специальные символы обрабат ываются коррект но	Форма принимает данные Происходит успешная авторизация Специальные символы обрабат ываются коррект но	та и доступна Сервер п риложен ия работает База данных доступна Установл ены различны е браузеры Доступн ы различны е устройст ва для тести рования	либо авторизо ван, либ о видит сообщен ие об ошибке Форма г отова к повторн ому вводу данных Состоян ие системы соответс твует результату теста	
9					Пробелы в начале/конце: Email: " user@example.c om " Пароль: " Password123! "	Пробелы в начале/конце: Пробелы автоматически удаляются Форма принимает очищенные данные Происходит успешная авторизация	Пробелы в начале/конце: Пробелы автоматически удаляются Форма принимает очищенные данные Происходит успешная авторизация	Система разверну та и доступна Сервер п риложен ия работает База данных доступна Установл ены	Пользов атель либо авторизо ван, либ о видит сообщен ие об ошибке Форма г отова к повторн ому	Pas s

				Максимальная длина полей: Email: ver y.long.email.addre ss@example.com Пароль: VeryLongPasswor d123456789!@#	Максимальная длина полей: Форма принимает данные Текст не обрезается Происходит успешная авторизация	Максимальная длина полей: Форма принимает данные Текст не обрезается Происходит успешная авторизация	Система развернута и доступна Сервер приложений работает База данных доступна Установлены различные браузеры Доступны различные устройства для тестирования	вводу данных Состояние системы соответствует результату теста	Pas s

--	--	--	--	--	--	--	--	--	--	--	--

Таблица 12 – ТС3 - Проверка структуры панели администратора

TC_ADMIN_PANEL	Приоритет теста	Название испытания	Резюме испытания	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Предпосылки	Постусловия	Статус (Pass/Fail)	Комментарии
1	Высокий	Проверка панели администратора	Проверка корректного отображения и функционирования панели администратора	Войти в систему как администратор Проверить отображение панели администратора Проверить функциональность всех элементов	Основная панель управления: URL: http://localhost:8000/admin/dashboard/ Роль: Администратор Разрешение экрана: 1920x1080 Браузер: Chrome	Основная панель управления: Отображается главное меню Видны основные метрики Доступны быстрые действия Навигация работает корректно	Основная панель управления: Отображается главное меню Видны основные метрики Доступны быстрые действия Навигация работает корректно	Система развернута и доступна Сервер приложения работает База данных доступна Создан аккаунт администратора Настроены права доступа	Панель администратора отображается корректно Все функции работают Состояние системы соответствует результату теста	Pass	
2					Управление пользователями:	Управление пользователя	Управление пользователя	Система развернута	Панель администратора	Pass	

				<p>URL: http://localhost:8000/admin/users/</p> <p>Действия:</p> <ul style="list-style-type: none"> Просмотр списка пользователей Добавление нового пользователя Редактирование пользователя Удаление пользователя 	<p>ми: Таблица пользователь отображается</p> <p>Формы добавления/редактирования работают</p> <p>Фильтрация и поиск функционируют</p> <p>Действия выполняются корректно</p>	<p>ми: Таблица пользователь отображается</p> <p>Формы добавления/редактирования работают</p> <p>Фильтрация и поиск функционируют</p> <p>Действия выполняются корректно</p>	<p>ута и доступна</p> <p>Сервер приложения работает</p> <p>База данных доступна</p> <p>Создан аккаунт администратора</p> <p>Настроены права доступа</p>	<p>ра отображается корректно</p> <p>Все функции работают</p> <p>Состояние системы соответствует результату теста</p>	
3				<p>Управление курсами:</p> <p>URL: http://localhost:8000/admin/courses/</p> <p>Действия:</p> <ul style="list-style-type: none"> Просмотр списка курсов Создание нового курса Редактирование курса Удаление курса 	<p>Управление курсами:</p> <p>Список курсов отображается</p> <p>Формы создания/редактирования работают</p> <p>Загрузка материалов функционирует</p> <p>Управление</p>	<p>Управление курсами:</p> <p>Список курсов отображается</p> <p>Формы создания/редактирования работают</p> <p>Загрузка материалов функционирует</p> <p>Управление</p>	<p>Система развернута и доступна</p> <p>Сервер приложения работает</p> <p>База данных доступна</p>	<p>Панель администратора отображается корректно</p> <p>Все функции работают</p> <p>Состояние системы соответствует результату теста</p>	Pass

					доступом работает	доступом работает	Создан аккаунт администратора Настроены права доступа			
4				Управление материалами: URL: http://localhost:8000/admin/materials/ Действия: Просмотр списка материалов Загрузка новых материалов Редактирование материалов Удаление материалов	Управление материалами: Список материалов отображается Загрузка файлов работает Редактирование доступно Удаление функционирует	Управление материалами: Список материалов отображается Загрузка файлов работает Редактирование доступно Удаление функционирует	Система развернута и доступна Сервер приложения работает База данных доступна Создан аккаунт администратора Настроены права доступа	Панель администратора отображается корректно Все функции работают Состояние системы соответствует результату теста	Pass	
5				Управление тестами: URL:	Управление тестами:	Управление тестами:	Система развернута	Панель администратора	Pass	

					http://localhost:8000/admin/tests/ Действия: Просмотр списка тестов Создание нового теста Редактирование теста Удаление теста	Список тестов отображается Создание тестов работает Редактирование доступно Удаление функционирует	Список тестов отображается Создание тестов работает Редактирование доступно Удаление функционирует	ута и доступна Сервер приложения работает База данных доступна Создан аккаунт администратора Настройки права доступа	ра отображается корректно Все функции работают Состояние системы соответствует результату теста		
6					Настройки системы: URL: http://localhost:8000/admin/settings/ Действия: Настройка параметров системы Управление правами доступа Настройка уведомлений Управление резервными копиями	Настройки системы: Формы настроек работают Сохранение настроек функционирует Права доступа применяются Резервное копирование работает	Настройки системы: Формы настроек работают Сохранение настроек функционирует Права доступа применяются Резервное копирование работает	Система развернута и доступна Сервер приложения работает База данных доступна	Панель администратора отображается корректно Все функции работают Состояние системы соответствует результату теста	Pass	

							Создан аккаунт админи стратор а Настрое ны прав а доступа				
7					Управление сертификатами: URL: http:// localhost:8000/ admin/certificates/ Действия: Просмотр списка сертификатов Создание сертифи ката Редактирование сертификата Удаление сертификата	Управление сертификатам и: Список сертификатов отображается Создание работает Редактирован ие доступно Удаление функциониру ет	Управление сертификатам и: Список сертификатов отображается Создание работает Редактирован ие доступно Удаление функциониру ет	Система разверн ута и доступн а Сервер прилож ения работае т База данных доступн а Создан аккаунт админи стратор а Настрое ны прав а доступа	Панель администратора отображает ся корректно Все функции работают Состояние системы соответствует результату теста	Pass	

Таблица 13 – ТС4 - Проверка отображения форм ввода данных

TC_INPUT_FORMS	Приоритет теста	Название тестирования	Резюме испытания	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Предпосылки	Постусловия	Статус (Pass/Fail)	Комментарии
1	Высокий	Проверка форм ввода данных	Проверка корректного отображения и функционирования всех форм ввода данных в системе	Открыть различные формы ввода Проверить отображение элементов Проверить валидацию и функциональность	Форма регистрации: URL: http://localhost:8000/register/ Поля: Имя Фамилия Email Пароль Подтверждение пароля Роль	Форма регистрации: Все поля отображаются Валидация работает Сообщения об ошибках видны Кнопка отправки активна	Форма регистрации: Все поля отображаются Валидация работает Сообщения об ошибках видны Кнопка отправки активна	Система развернута и доступна Сервер приложения работает База данных доступна Созданы тестовые данные Настроены права доступа	Формы отображаются корректно Валидация работает Данные сохраняются Состояние системы соответствует результату теста	Pass	
2					Форма создания курса: URL: http://localhost:8000/courses/create/	Форма создания курса: Поля	Форма создания курса: Поля	Система развернута и	Формы отображаются корректно	Pass	

3				<p>Поля: Название курса Описание Длительность Уровень сложности Категория Цена</p>	<p>заполняются Валидация работает Загрузка файлов доступна Сохранение функционирует</p>	<p>заполняются Валидация работает Загрузка файлов доступна Сохранение функционирует</p>	<p>доступ на Сервер приложения работает База данных доступна Созданы тестовые данные Настроены права доступа</p>	<p>но Валидация работает Данные сохраняются Состояние систем соответствует результату теста</p>
				<p>Форма создания лекции: URL: http://localhost:8000/lectures/create/ Поля: Название лекции Описание Курс Длительность Материалы Тесты</p>	<p>Форма создания лекции: Поля отображаются Валидация работает Загрузка материалов доступна Связь с курсом работает</p>	<p>Форма создания лекции: Поля отображаются Валидация работает Загрузка материалов доступна Связь с курсом работает</p>	<p>Система развернута и доступна Сервер приложения работает База данных доступна</p>	<p>Формы отбираются корректно Валидация работает Данные сохраняются Состояние</p>

							на Создан ы тестов ые данные Настро ены права доступ а	систем ы соотв етствует результ ату теста			
4					Форма создания теста: URL: http:// localhost:8000/tests/ create/ Поля: Название теста Описание Время на выполнение Проходной балл Вопросы Настройки	Форма создания теста : Поля заполняются Валидация работает Добавление вопросов доступно Настройки сохраняются	Форма создания теста : Поля заполняются Валидация работает Добавление вопросов доступно Настройки сохраняются	Систе ма разверн ута и доступ на Сервер прилож ения работае т База данных доступ на Создан ы	Формы отобра жаются коррект но Валида ция работае т Данные сохраня ются Состоя ние систем	Pas s	

							a						
5					<p>Форма профиля пользователя: URL: http://localhost:8000/profile/edit/</p> <p>Поля: Foto профиля Имя Фамилия Email Телефон О себе</p>	<p>Форма профиля пользователя: Поля отображаются Загрузка фото работает Валидация функционирует Сохранение доступно</p>	<p>Форма профиля пользователя: Поля отображаются Загрузка фото работает Валидация функционирует Сохранение доступно</p>	<p>Система развернута и доступна на Сервере приложения</p>	<p>Формы отображаются корректно</p>	<p>Формы отображаются корректно</p>	<p>Формы отображаются корректно</p>	<p>Формы отображаются корректно</p>	
6					<p>Форма заявки на обучение: URL: http://localhost:8000/training/request/</p> <p>Поля: ФИО Email</p>	<p>Форма заявки на обучение: Поля заполняются Валидация работает Выбор курса доступен</p>	<p>Форма заявки на обучение: Поля заполняются Валидация работает Выбор курса доступен</p>	<p>Система развернута и доступна на Сервере прилож</p>	<p>Формы отображаются корректно</p>	<p>Формы отображаются корректно</p>	<p>Формы отображаются корректно</p>		

					Курс Компания Должность Комментарий	Отправка функционирует	Отправка функционирует	ения работают База данных доступна Созданы тестовые данные Настроены права доступа	т Данные сохраняются Состояние систем соответствует результату теста		
--	--	--	--	--	--	------------------------	------------------------	---	--	--	--

Таблица 14 – ТС5 - Проверка отображения таблиц и списков:

TC_TABLES_LISTS	Приоритет теста	Название тестирования	Резюме испытания	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Предпосылки	Постусловия	Статус (Pass/ Fail)	Комментарии
1	Высокий	Проверка отображения таблиц и списков	Проверка корректного отображения и функционирования всех таблиц и списков	Открыть различные таблицы и списки Проверить отображение данных Проверить функциональность	Список курсов: URL: http://localhost:8000/courses/ Элементы: Название курса Описание Преподаватель Количество сотрудников	Список курсов: Данные отбираются Сортировка работает Сортировка работает Фильтрация доступна Пагинация функционирует	Список курсов: Данные отбираются Сортировка работает Фильтрация доступна Пагинация функционирует	Система развернута и доступна Сервер приложения работает	Таблицы и списки отображаются корректно Функциональность работает Данные актуальны	Pass	

			в системе	управления	Рейтинг Статус			ет База данных доступ на Создан ы тестов ые данные Настро ены права д оступа	ы Состояни е системы соответст вует результат у теста		
2					Таблица сотрудников: URL: http://localhost:8000/employee/ Элементы: ФИО Email Группа Прогресс Оценки Статус	Таблица сотрудников: Информация отображается Поиск работает ФИО Email Группа Прогресс Оценки Статус	Таблица сотрудников: Информация отображается Поиск работает Экспорт доступен Действия функциониру ют	Систем а развер нута и доступ на Сервер прило жения работа ет База данных доступ на Создан ы тестов ые данные Настро	Таблицы и списки отобража ются корректно Функцио нальность работает Данные актуальн ы Состояни е системы соответст вует результат у теста	Pas s	

							ены права д оступа			
3				<p>Список лекций: URL: http://localhost:8000/lectures/</p> <p>Элементы: Название Курс Длительность Материалы Тесты Статус</p>	<p>Список лекций: Элементы ото бражаются Группировка работает Фильтры доступны Навигация функционируе т</p>	<p>Список лекций: Элементы ото бражаются Группировка работает Фильтры доступны Навигация функционируе т</p>	<p>Систем а развер нута и доступ на Сервер прило жения работа ет База данных доступ на Создан ы тестов ые данные Настро ены права д оступа</p>	<p>Таблицы и списки отобра ются корректно Функцио нальность работает Данные актуальн ы Состояни е системы соответст вует результат у теста</p>		Pas s
4				<p>Таблица тестов: URL: http://localhost:8000/tests/</p> <p>Элементы: Название Курс</p>	<p>Таблица тестов: Данные отобра жаются Сортировка ра ботает Статистика</p>	<p>Таблица тестов: Данные отобра жаются Сортировка ра ботает Статистика</p>	<p>Систем а развер нута и доступ на Сервер</p>	<p>Таблицы и списки отобра ются корректно Функцио нальность</p>	Pas s	

5				<p>Время Вопросы Проходной балл Статус</p>	<p>доступна Управление функционирует</p>	<p>доступна Управление функционирует</p>	<p>приложения работает База данных доступна Созданы тестовы е данные Настро ены права д оступа</p>	<p>работает Данные актуальн ы Состояни е системы соответст вует результат у теста</p>
				<p>Список уведомлений: URL: http://localhost:8000/notifications/ Элементы: Тип Текст Время Статус Действия Приоритет</p>	<p>Список уведомлений: Уведомления отображаются Фильтрация работает Действия доступны Обновление функционирует</p>	<p>Список уведомлений: Уведомления отображаются Фильтрация работает Действия доступны Обновление функционирует</p>	<p>Система развернута и доступна на Сервере</p> <p>приложения работают База данных доступна Созданы тестовы</p>	<p>Таблицы и списки отображаются корректно Функциональность работает Данные актуальны Состояние системы соответствует результату теста</p>

							ые даннны е настро ены права д оступа			
6				Таблица прогресса: URL: http://localhost:8000/progress/ Элементы: Сотрудник Курс Лекции Тесты Оценки Сертификаты	Таблица прогресса: Прогресс отображается Фильтры работают Экспорт доступен Анализ функц ионирует	Таблица прогресса: Прогресс отображается Фильтры работают Экспорт доступен Анализ функц ионирует	Систем а развер нута и доступ на Сервер прило жения работа ет База данных доступ на Создан ы тестов ые даннны е настро ены права д оступа	Таблицы и списки отобража ются корректно Функцио нальность работает Данные актуальн ы Состояни е системы соответст вует результат у теста	Pas s	
7				Список заявок: URL: http://localhost:8000/requests/	Список заявок: Заявки отобра жаются Сортировка	Список заявок: Заявки отобра жаются Сортировка	Систем а развер нута и	Таблицы и списки отобража ются	Pas s	

					Элементы: ID Тип Статус Дата Исполнитель Приоритет	работает Статусы доступны Управление фу нкционирует	работает Статусы доступны Управление фу нкционирует	доступ на Сервер прило жения работа ет База данных доступ на Создан ы тестов ые данные Настро ены права д оступа	корректно Функцио нальность работает Данные актуальн ы Состояни е системы соответст вует результат у теста	
--	--	--	--	--	--	---	---	---	--	--

Таблица 15 – ТС6 - Проверка функции авторизации пользователей:

TC_AUTH_FUNCTION	Приоритет теста	Название тестирования	Резюме испытания	Шаги тестированния	Данные тестирования	Ожидаемый результат	Фактический результат	Предпосылки	Постуловия	Статус (Pass/Fail)	Комментарии
1	Высокий	Проверка функции авторизации пользователей	Проверка корректной работы функции авторизации	Проверить вход в систему Проверить валидацию данных Проверить обработку о	Проверить вход с корректными данными: URL: http://localhost:8000/login/ Тестовые данные:	Вход с корректными данными: Перенаправление на главную Создание	Вход с корректными данными: Перенаправление на главную Создание	Система развернута и доступна Сервер приложения работает База данных доступна	Авторизация работает корректно Безопасность обеспече	Pass	

					Email: test@example.co m Пароль: Test123! Ожидаемый результат: Успешный вход	сессии Отображен ие профиля Доступ к функциям	сессии Отображен ие профиля Доступ к функциям	Созданы тестовые акк аунты Настроены п рава доступа	на Сессии управля ются Состоян ие системы соответс твует результату теста		
2		пользова телей в системе	шибок Проверить безопаснос ть		Вход с неверным email: URL: http://localhost:8 000/login/ Тестовые данные: Email: wrong@example .com Пароль: Test123! Ожидаемый результат: Ошибка авторизации	Вход с неверным email: Сообщени е об ошибке Сохранени е email Очистка пароля Возможнос ть повтора	Вход с неверным email: Сообщени е об ошибке Сохранени е email Очистка пароля Возможнос ть повтора	Система развернута и доступна Сервер прил ожения работает База данных доступна Созданы тестовые акк аунты Настроены п рава доступа	Авториз ация работает коррект но Безопасн ость обеспече на Сессии управля ются Состоян ие системы соответс твует результату теста	Pass	
					Вход с неверным паролем: URL:	Вход с неверным паролем: Сообщени	Вход с неверным паролем: Сообщени	Система развернута и доступна Сервер прил ожения работает База данных доступна Созданы тестовые акк аунты Настроены п рава доступа	Авториз ация работает коррект но Безопасн ость обеспече на Сессии управля ются Состоян ие системы соответс твует результату теста		
										Pass	

					<p>http://localhost:8000/login/</p> <p>Тестовые данные:</p> <p>Email: test@example.com</p> <p>Пароль: WrongPass123!</p> <p>Ожидаемый результат:</p> <p>Ошибка авторизации</p>	<p>е об ошибке Сохранение email</p> <p>Очистка пароля Счетчик попыток</p>	<p>е об ошибке Сохранение email</p> <p>Очистка пароля Счетчик попыток</p>	<p>ожения работает База данных доступна</p> <p>Созданы тестовые аккаунты</p> <p>Настроены права доступа</p>	<p>но Безопасность обеспечена</p> <p>Сессии управляются</p> <p>Состояние системы соответствует результату теста</p>	
4					<p>Вход с пустыми полями:</p> <p>URL: http://localhost:8000/login/</p> <p>Тестовые данные:</p> <p>Email: (пусто)</p> <p>Пароль: (пусто)</p> <p>Ожидаемый результат:</p> <p>Валидация полей</p>	<p>Вход с пустыми полями: Подсветка полей Сообщения об ошибках Предотвращение отправки Валидация в реальном времени</p>	<p>Вход с пустыми полями: Подсветка полей Сообщения об ошибках Предотвращение отправки Валидация в реальном времени</p>	<p>Система развернута и доступна</p> <p>Сервер приложении работает</p> <p>База данных доступна</p> <p>Созданы тестовые аккаунты</p> <p>Настроены права доступа</p>	<p>Авторизация работает корректно</p> <p>Безопасность обеспечена</p> <p>Сессии управляются</p> <p>Состояние системы соответствует результату теста</p>	Pass

Таблица 16 – ТС7 - Проверка функции регистрации новых пользователей

TC_REGISTRATION_FUNCTION	Приоритет теста	Название тестированья	Резюме испытания	Шаги тестирования	Данные тестирования	Ожидаемый результат	Фактический результат	Предпосылки	Постусловия	Статус (Pass/Fail)	Комментарии
1	Высокий	Проверка функции регистрации новых пользователей	Проверка корректной работы функции регистрации новых пользователей в системе	Проверить форму регистрации Проверить валидацию данных Проверить создание аккаунта	Регистрация с корректными данными: URL: http://localhost:8000/register/ Тестовые данные: Имя: Иван Фамилия: Иванов Email: ivan@example.com Пароль: Test123! Подтверждение пароля: Test123! Роль: Сотрудник	Регистрация с корректными данными: Имя: Иван Фамилия: Иванов Email: ivan@example.com Пароль: Test123! Подтверждение пароля: Test123! Роль: Сотрудник	Регистрация с корректными данными: Создание аккаунта Отправка email Перенаправление Сообщение об успехе	Система развернута и доступна Сервер приложения работает База данных доступна Настроена отправка email Настроены права доступа	Регистрация работает корректно Email отправляется Аккаунты создаются Состояние системы соответствует результату теста	Pass	
2					Регистрация с существующим email: URL: http://localhost:8000/register/	Регистрация с существующим email: Сообщен	Регистрация с существующим email: Сообщен	Система развернута и доступна Сервер приложения работает	Регистрация работает корректно Email отправляется	Pass	

					Тестовые данные: Имя: Петр Фамилия: Петров Email: test@example.com (существующий) Пароль: Test123! Подтверждение пароля: Test123! Роль: Сотрудник	ие об ошибке Сохранение данных Подсветка поля Возможность исправления	ие об ошибке Сохранение данных Подсветка поля Возможность исправления	База данных доступна Настроена отправка email Настроены права доступа	Аккаунты создаются Состояние системы соответствует результату теста	
3					Регистрация с несовпадающими паролями: URL: http://localhost:8000/register/ Тестовые данные: Имя: Сергей Фамилия: Сергеев Email: sergey@example.com Пароль: Test123! Подтверждение пароля: Test456!	Регистрация с несовпадающими паролями: Сообщение об ошибках Очистка паролей Подсветка полей Валидация в реальном времени	Регистрация с несовпадающими паролями: Сообщение об ошибках Очистка паролей Подсветка полей Валидация в реальном времени	Система развернута и доступна Сервер приложения работает База данных доступна Настроена отправка email Настроены права доступа	Регистрация работает корректно Email отправляется Аккаунты создаются Состояние системы соответствует результату теста	Pas s

4			Роль: Сотрудник				
5			<p>Регистрация с некорректным email:</p> <p>URL: http://localhost:8000/register/</p> <p>Тестовые данные:</p> <p>Имя: Алексей</p> <p>Фамилия: Алексеев</p> <p>Email: invalid-email</p> <p>Пароль: Test123!</p> <p>Подтверждение пароля: Test123!</p> <p>Роль: Сотрудник</p>	<p>Регистрация с некорректным email:</p> <p>Сообщение об ошибке Подсветка поля Валидация формата Подсказка формата</p>	<p>Регистрация с некорректным email:</p> <p>Сообщение об ошибке Подсветка поля Валидация формата Подсказка формата</p>	<p>Система развернута и доступна Сервер приложения работает База данных доступна Настроена отправка email Настроены права доступа</p>	<p>Регистрация работает корректно Email отправляется Аккаунты создаются Состояние системы соответствует результату теста</p>

					Пароль: (пусто) Подтверждение пароля: (пусто) Роль: (не выбрана)	отправки Валидации в реальном времени	отправки Валидации в реальном времени	Настроены права доступа	результату теста			
6					Регистрация с недопустимыми символами: URL: http://localhost:8000/register/ Тестовые данные: Имя: Иван@#\$% Фамилия: Иванов@#\$% Email: ivan@example.com Пароль: Test123! Подтверждение пароля: Test123! Роль: Сотрудник	Регистрация с недопустимыми символами: Сообщение об ошибке Подсветка полей Валидация символов Подсказка допустимых символов	Регистрация с недопустимыми символами: Сообщение об ошибке Подсветка полей Валидация символов Подсказка допустимых символов	Система развернута и доступна Сервер приложения работает База данных доступна Настроена отправка email Настроены права доступа	Регистрация работает корректно Email отправляется Аккаунты создаются Состояние системы соответствует результату теста	Pas s		

7. ПРОТОКОЛ ТЕСТОВЫХ ИСПЫТАНИЙ

Заказчик: Кириллин К.М.

Объект испытаний: программа «BustCom»

Изготовитель: Кириллин К.М.

Дата проведения испытаний: 28.03.2025

Таблица 17. Результаты испытаний

№ п.п.	Наименование тест кейса	Позитивный	Негативный	Описание
1	2	3	4	5
Тестирование интерфейса				
1	TC1 - Проверка доступности главной страницы системы	8/8	Не проводилось	
2	TC2 - Проверка структуры окна "Авторизация"	10/10	Не проводилось	
3	TC3 - Проверка структуры панели администратора	7/7	Не проводилось	
4	TC4 - Проверка отображения форм ввода данных	6/6	Не проводилось	
5	TC5 - Проверка отображения таблиц с данными	7/7	Не проводилось	
Итого по разделу		38/38	-	
Тестирование функциональности				
1	TC6 - Проверка функции авторизации пользователей	4/4	Не проводилось	
2	TC7 - Проверка функции регистрации новых пользователей	6/6	Не проводилось	
Итого по разделу		10/10	-	
Тестирование безопасности				
1	TC8 - Проверка защиты от несанкционированного доступа	8/8	Не проводилось	
Итого по разделу		8/8	-	

ЗАКЛЮЧЕНИЕ: Представленный образец соответствует требованиям по проверяемым показателям.

ПРИЛОЖЕНИЕ Д. Руководство пользователя

АННОТАЦИЯ

Настоящее руководство пользователя предназначено для ознакомления с программой BustCom (далее – программа). Приложение предназначено для автоматизации процесса повышения квалификации сотрудников. Оно включает в себя учет данных о пройденных курсах, регистрацию на новые программы обучения, доступ к учебным материалам, мониторинг прогресса сотрудников, обратную связь по курсам, а также анализ эффективности обучения.

Содержание

1.	Назначение программы.....	3
1.1.	Назначение.....	3
1.2.	Основные возможности программы.....	3
2.	Условия выполнения программы.....	4
2.1.	Уровень подготовки пользователя.....	4
2.2.	Выполняемые функции.....	4
2.3.	Программные и аппаратные требования к системе.....	5
3.	Выполнение программы.....	7
3.1.	Подготовка к работе.....	7
3.2.	Описание функций программы.....	8
3.3.	Основные функции программы.....	8
4.	Возможные неполадки и способы устранения.....	22

1. Назначение программы

1.1. Назначение

Настоящее руководство предназначено для ознакомления с системой управления обучением BustCom. Программа предназначена для организации, контроля и управления процессом обучения сотрудников компании. Руководство содержит описание работы и сведения о функциях программы.

1.2. Основные возможности программы

- управление учебными курсами и материалами через административную панель;
- отслеживание прогресса обучения сотрудниками HR-отделом;
- создание и управление планами обучения менеджерами;
- проведение оценки эффективности обучения;
- управление тестированием с настройкой уровней сложности;
- автоматическая генерация и выдача сертификатов по завершении курсов;
- создание резервных копий данных и их восстановление;
- отслеживание производительности сотрудников;
- назначение курсов сотрудникам и контроль их выполнения.

Примечание: Система построена на основе веб-технологий с использованием фреймворка Django, что обеспечивает:

- многопользовательский доступ с разграничением прав (администратор, HR-менеджер, менеджер обучения, сотрудник);

- централизованное хранение данных в базе SQLite;
- возможность масштабирования системы под растущие потребности компании.

2. Условия выполнения программы

2.1. Уровень подготовки пользователя

Пользователь программы BustCom должен быть ознакомлен с данным руководством пользователя и иметь следующие навыки:

- базовые навыки работы с веб-браузером;
- понимание основ работы с системами управления обучением;
- навыки работы с документами в электронном формате;
- базовое понимание принципов корпоративного обучения.

Для администраторов системы дополнительно требуется:

- понимание принципов управления пользователями и правами доступа;
- навыки работы с системами резервного копирования данных;
- базовое понимание принципов работы веб-приложений.

2.2. Выполняемые функции

Программа выполняет следующие основные функции: Для администраторов:

- управление пользователями и их ролями;
- настройка параметров системы;
- управление резервными копиями;
- мониторинг активности пользователей.

Для HR-менеджеров:

- управление профилями сотрудников;
- назначение курсов;
- отслеживание прогресса обучения;
- формирование отчетов по обучению.

Для менеджеров обучения:

- создание и редактирование учебных материалов;
- настройка тестов и их сложности;
- формирование учебных планов;
- анализ результатов обучения.

Для сотрудников:

- прохождение назначенных курсов;
- выполнение тестов;
- получение сертификатов;
- отслеживание личного прогресса.

2.3. Программные и аппаратные требования к системе

2.3.1. Требования к программному обеспечению

Для работы с системой требуется:

- Операционная система: Windows 7/8/10/11, macOS 10.12 или новее, Linux с графическим интерфейсом
- Веб-браузер:
- Google Chrome (версия 90 или выше)
- Mozilla Firefox (версия 88 или выше)
- Microsoft Edge (версия 90 или выше)
- Safari (версия 14 или выше)
- Поддержка JavaScript в браузере
- Поддержка cookies в браузере

2.3.2. Требования к аппаратному обеспечению

Минимальные требования для клиентских рабочих мест:

- Процессор: двухъядерный процессор с частотой 2.0 ГГц или выше
- Оперативная память: 4 ГБ или более
- Свободное место на диске: 1 ГБ
- Разрешение экрана: 1366x768 или выше
- Подключение к локальной сети или интернету: от 1 Мбит/с

Рекомендуемые требования для сервера:

- Процессор: четырехъядерный процессор с частотой 2.5 ГГц или выше
- Оперативная память: 8 ГБ или более
- Свободное место на диске: 20 ГБ или более (зависит от количества учебных материалов)
- Сетевое подключение: 100 Мбит/с или выше

3. Выполнение программы

3.1. Подготовка к работе

3.1.1. Состав дистрибутива

Программа BustCom поставляется в виде веб-приложения, развернутого на сервере Django. Для работы требуется только веб-браузер и доступ к серверу.

3.1.2. Запуск программы

Перед запуском программы необходимо:

- убедиться в наличии подключения к серверу;
- иметь актуальные учетные данные для входа в систему.

После открытия программы в браузере на экране появится главная страница, показанная на рис.1 (внешний вид может отличаться в зависимости от настроек браузера).

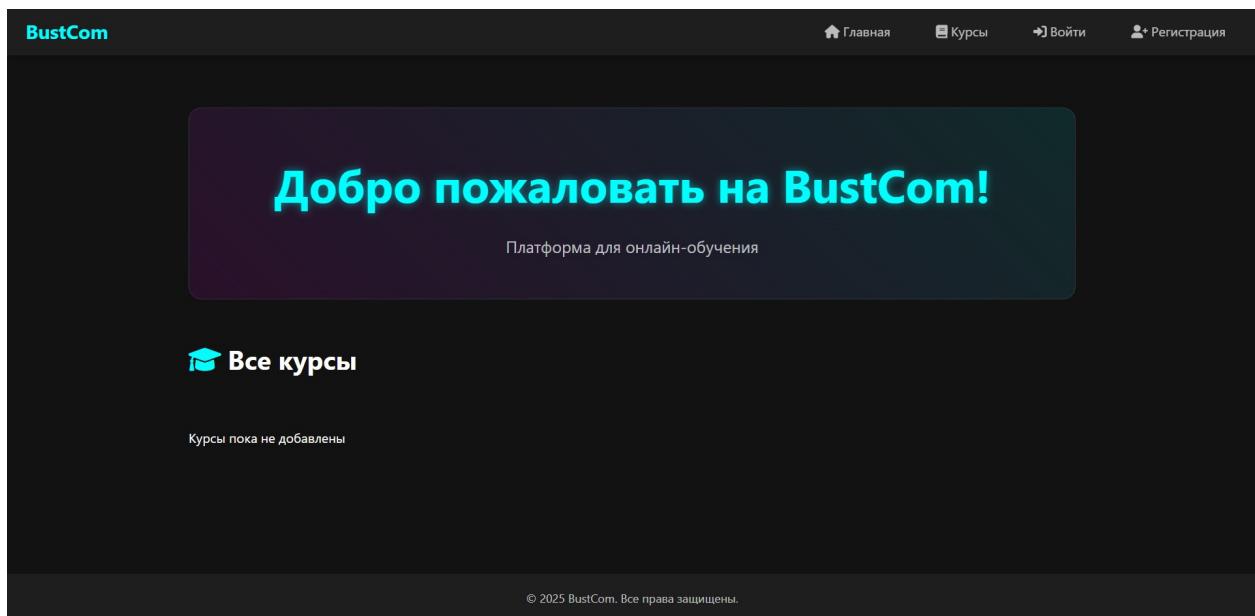


Рисунок 1 - главная страница

3.1.3. Проверка работоспособности программы

3.1.3.1. Проверяется корректность подключения к системе путем:

- успешной авторизации в системе;
- отображения главной панели управления;
- наличия доступа к назначенным модулям согласно роли пользователя.

3.1.3.2. В случае корректного подключения:

- в панели статуса отображается имя пользователя и его роль;
- доступны все назначенные функции;
- отображается актуальное состояние курсов и обучения.

3.1.3.3. В случае некорректного подключения:

- появляется сообщение об ошибке авторизации;
- отсутствует доступ к функциям системы;
- необходимо проверить правильность учетных данных и доступность сервера.

3.2. Описание функций программы

3.2.1. Элементы основного окна программы

Элементами основного окна программы являются в зависимости от роли пользователя (см. рис.2 - 6):

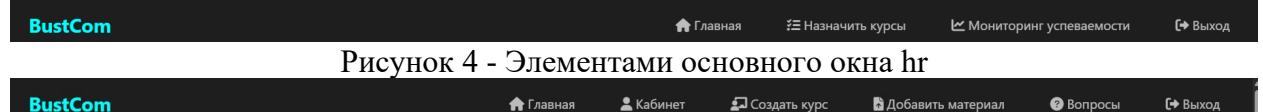


Рисунок 5 - Элементами основного окна преподователя



Рисунок 6 - Элементами основного окна менеджера

3.3. Основные функции программы

3.3.1. Описание основных элементов интерфейса

3.3.1.1. Панель администратора

На рис.7 представлена панель администратора, в которой пользователь с соответствующей ролью может взаимодействовать с базой данных, а так же создавать и загружать ее резервные копии

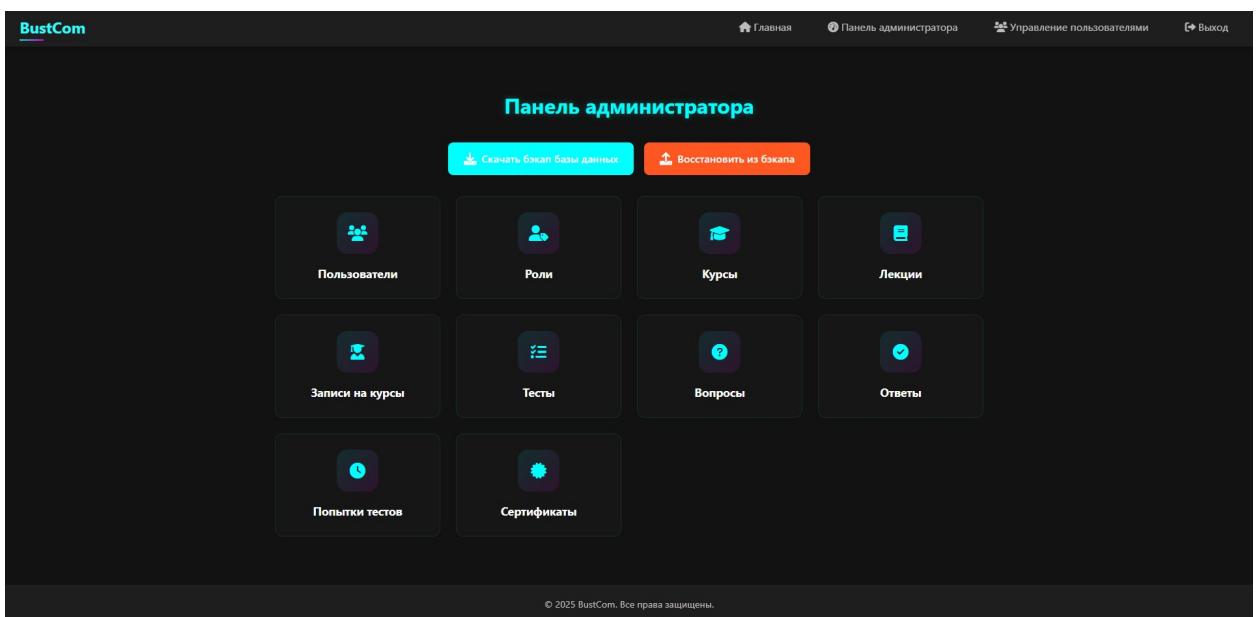


Рисунок 7 - панель администратора

На рис. 8 преставлен пример работы с таблицами бд. Пользователь может выбрать нужные данные и редактировать\удалить их. Так же можно добавить новые

roles		+ Добавить запись
id	name	Действия
1	hr	<input checked="" type="button"/> РЕДАКТИРОВАТЬ <input type="button"/> УДАЛИТЬ
2	admin	<input checked="" type="button"/> РЕДАКТИРОВАТЬ <input type="button"/> УДАЛИТЬ
3	teacher	<input checked="" type="button"/> РЕДАКТИРОВАТЬ <input type="button"/> УДАЛИТЬ
4	training_manager	<input checked="" type="button"/> РЕДАКТИРОВАТЬ <input type="button"/> УДАЛИТЬ
5	employee	<input checked="" type="button"/> РЕДАКТИРОВАТЬ <input type="button"/> УДАЛИТЬ
6	student	<input checked="" type="button"/> РЕДАКТИРОВАТЬ <input type="button"/> УДАЛИТЬ

Рисунок 8 - таблицы бд

На рис.9-10 приведены примеры добавления и редактирования данных в бд.

Добавить в roles

Name:

СОХРАНИТЬ

— Назад

© 2025 BustCom. Все права защищены.

Рисунок 9 - добавление

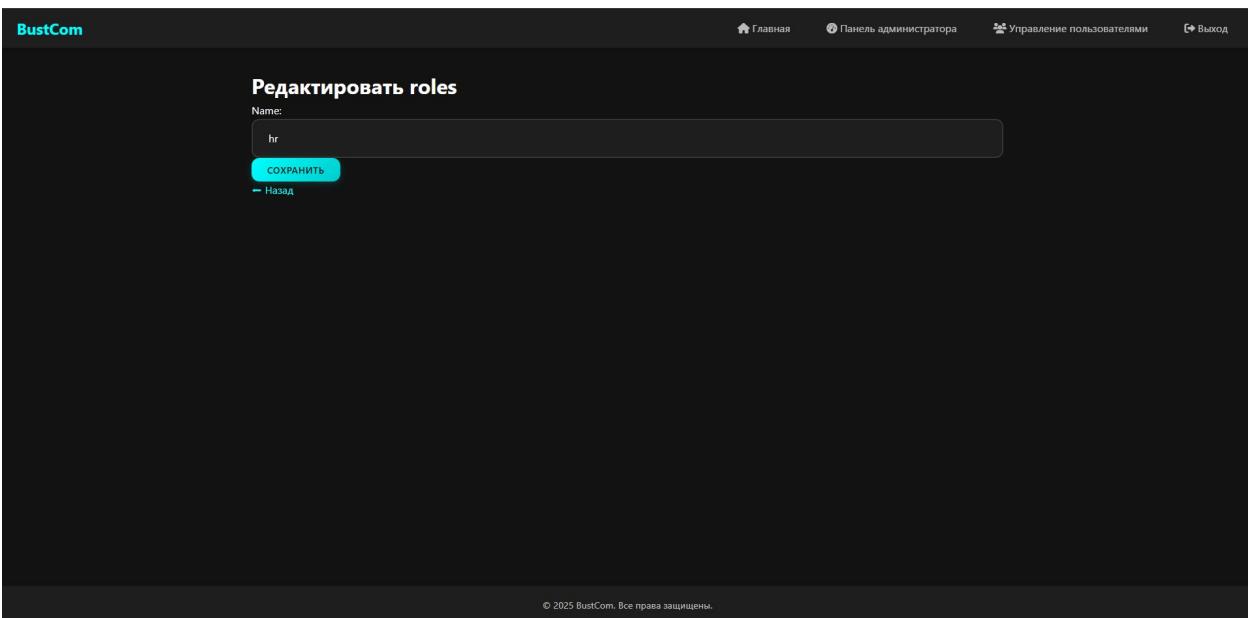


Рисунок 10 – редактирование

Сохранение бэкапа бд просиходит путем нажатия на кнопку сохранить бд, после чего файл скачивается на компьютер пользователя. В случае загрузки необходимо перейти на страницу восстановления и загрузить файл (рис. 11)

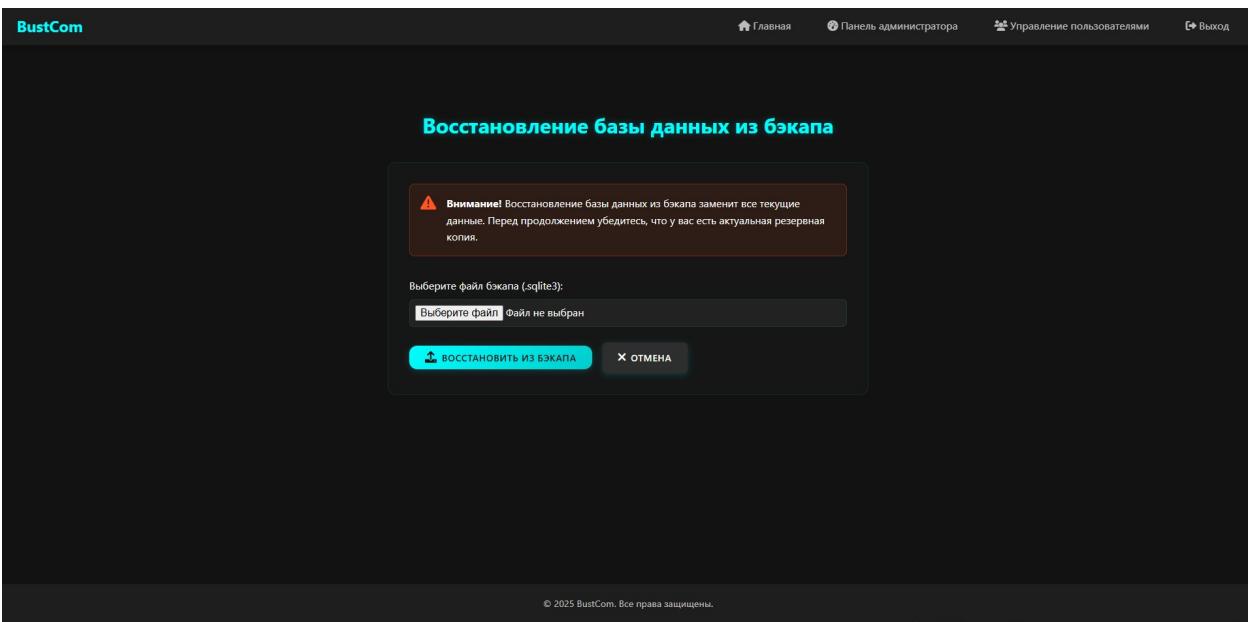


Рисунок 11 - восстановление бд

3.3.1.2. Панель сотрудника

На рис.12 представлена панель сотрудника, на которой отображены курсы и основная часть программы. Также отображается статистика и успеваемость.

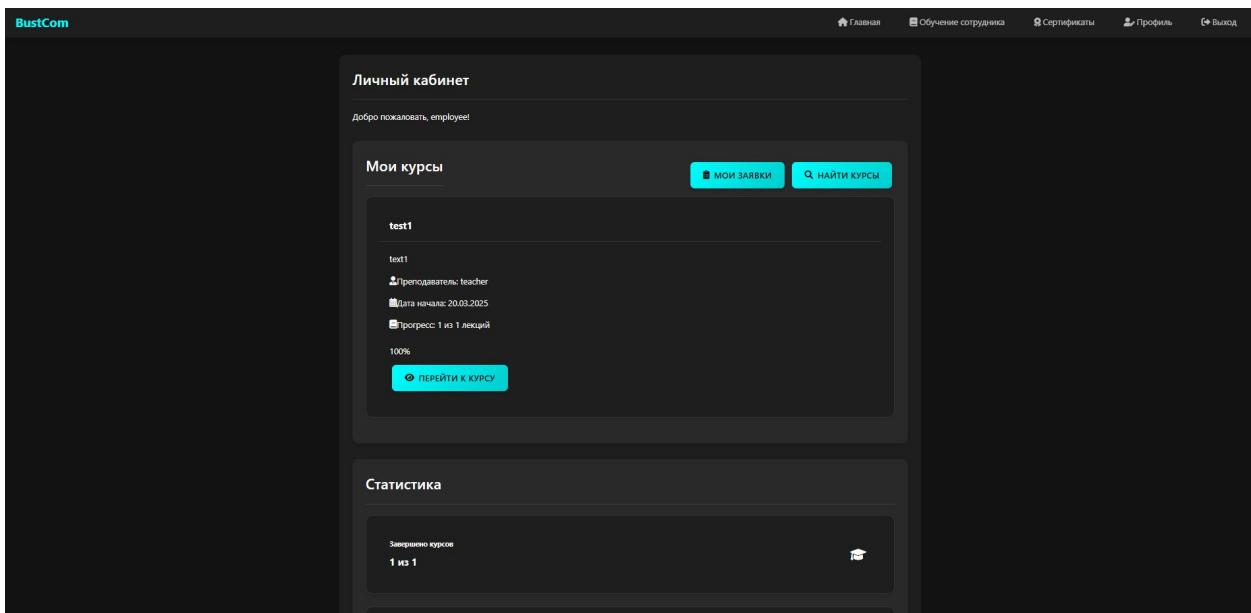


Рисунок 12 -панель сотрудника

Курс, на котором записан пользователь представлен на рис.13. На странице курса отображаются лекции, а также длительность прохождения курса в часах.

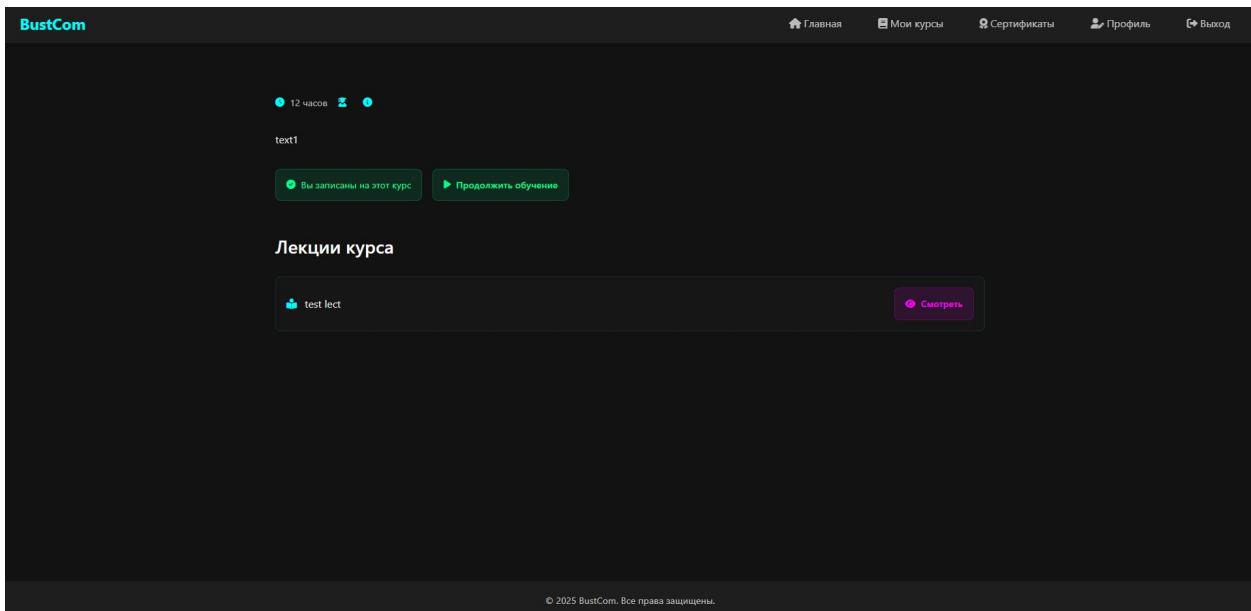


Рисунок 13 - странице курса

Страница просмотра лекций курса представлена на рис.14. На данной странице отображается информация существующих лекциях курса, где можно пройти повторно лекцию или тест к ней.

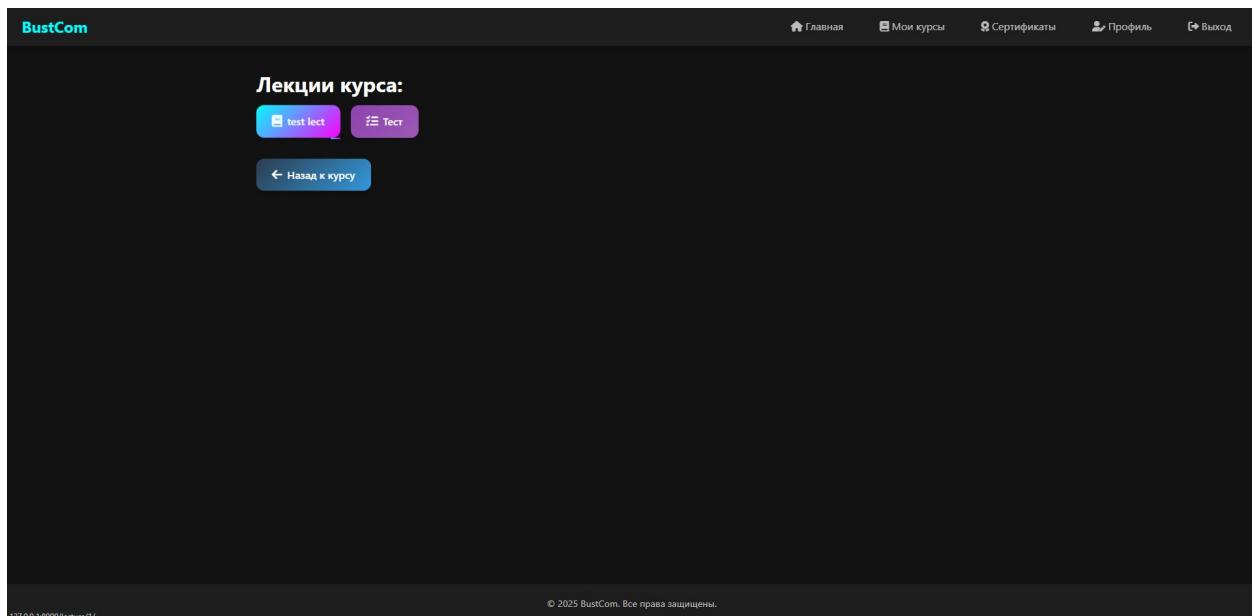


Рисунок 14 - Страница просмотра лекций курса

Сама лекция представлена на рис.15, где пользователю представлены материалы в виде ссылки на видео или текст, а также тест и возможность задать во прососы к лекции. Также присутствует кнопка «отметить как проходимую», благодаря которой пользователь отмечает что прошел и изучил материал.

The screenshot shows a dark-themed web page for a lecture titled "test lect". At the top, it says "Порядковый номер: 2" and "Материалы:". Below that are two blue buttons: "СМОТРЕТЬ ВИДЕО" (Watch Video) and "← НАЗАД К ЛЕКЦИЯМ" (Back to Lectures). A section titled "Тесты" contains a blue button labeled "ПРОЙТИ ТЕСТ "TEST"" (Take Test). The main content area is titled "Комментарии" (Comments). It displays several comments from users: "employee" (April 5, 2025, 11:33 a.m.) with the message "а как че делать ?"; "zzz" (April 5, 2025, 12:11 a.m.) with the message "asdasd"; "ilya" (April 4, 2025, 11:57 p.m.) with the message "asdasd"; and "student" (March 21, 2025, 7:30 a.m.) with the message "ну норм тест". Below the comments is a section titled "Добавить комментарий:" (Add comment:) with a text input field and a "Text:" placeholder. A blue button at the bottom of this section says "ОТПРАВИТЬ" (Send). At the very bottom is a blue button with a checkmark and the text "✓ ОТМЕТИТЬ КАК ПРОЙДЕННУЮ" (Mark as completed).

Рисунок 15 - страница лекции

На рисунке 16 представлена страница теста, в которой выгружаются вопросы и варианты ответов (есть три варианта: 1- один правильный, 2 – несколько вариантов правильные, 3 – ответ текстом)

The screenshot shows a test page from BustCom. At the top, there is a navigation bar with links for 'Главная', 'Мои курсы', 'Сертификаты', 'Профиль', and 'Выход'. The main content area has a dark background. It displays a question: 'кто устал?' with two options: 'я' and 'не я'. A blue button labeled 'ОПРАВИТЬ ОТВЕТЫ' is visible. Below the question is a section titled 'Ваши попытки:' containing a list of attempts. The list includes several entries from March 21, 2025, and one from March 22, 2025. Each entry shows the attempt time, the number of correct answers (баллы), and the percentage. At the bottom of the page is a copyright notice: '© 2025 BustCom. Все права защищены.'

Рисунок 16 - страница теста

На рисунке 17 отображен результат прохождения теста

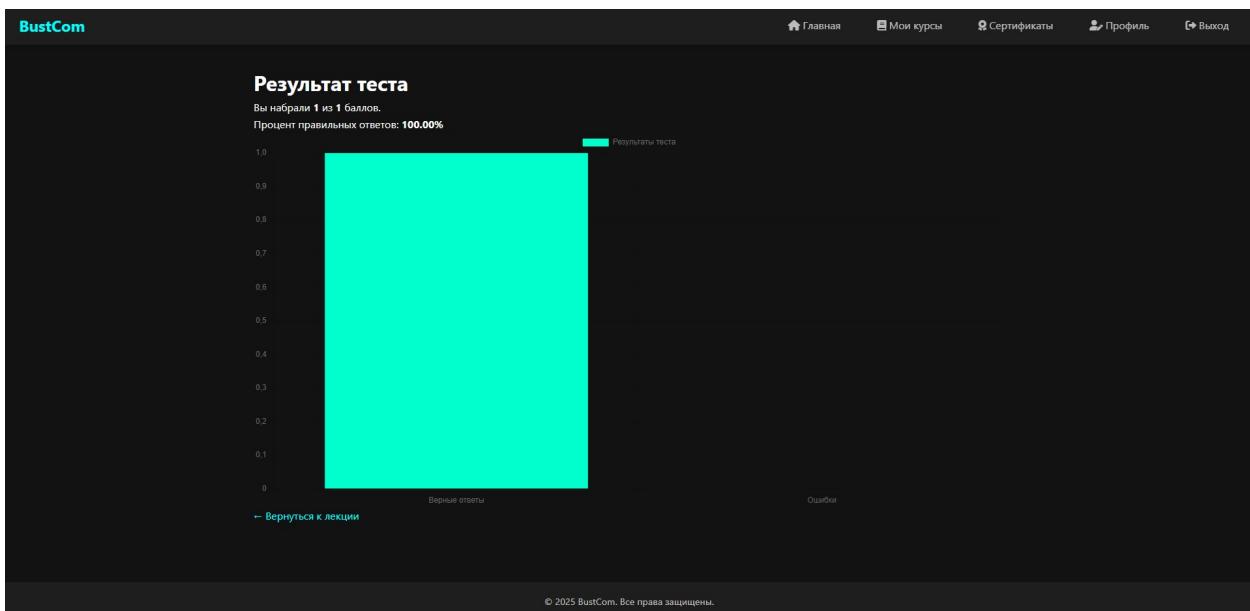


Рисунок 17 - страница с результатом прохождения тестирования

На рис.18 отображена страница с сертификатами пользователя. После прохождения лекции целиком пользователю выдается сертификат, информирующий об этом. На рис.19 отображен сам сертификат.

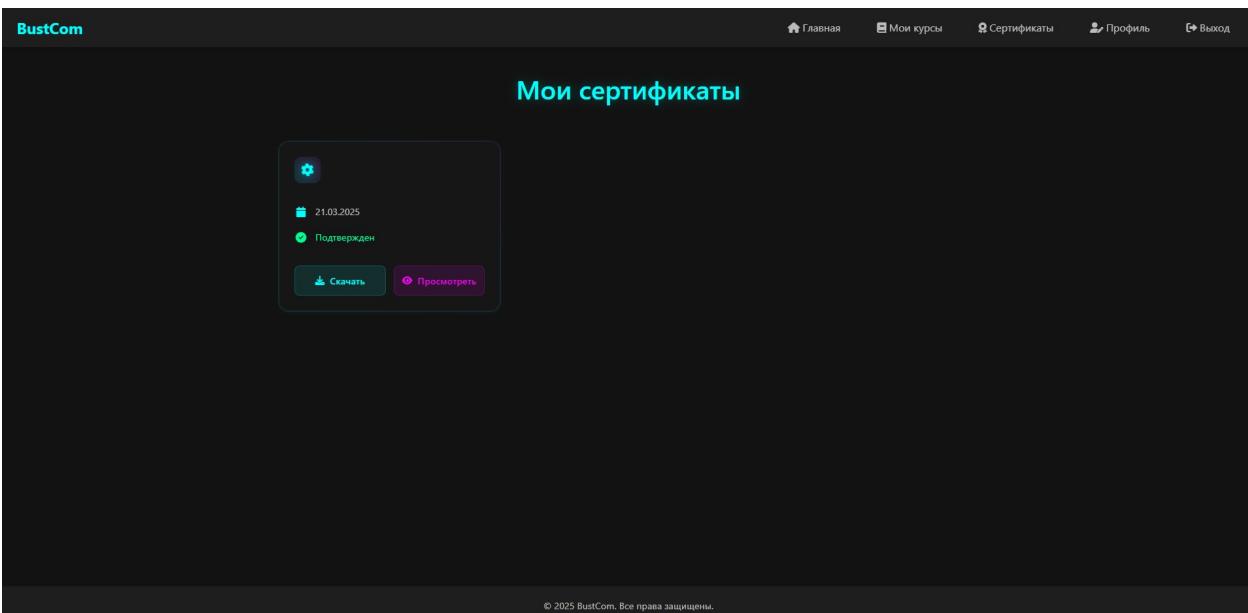


Рисунок 18 - страница с сертификатами

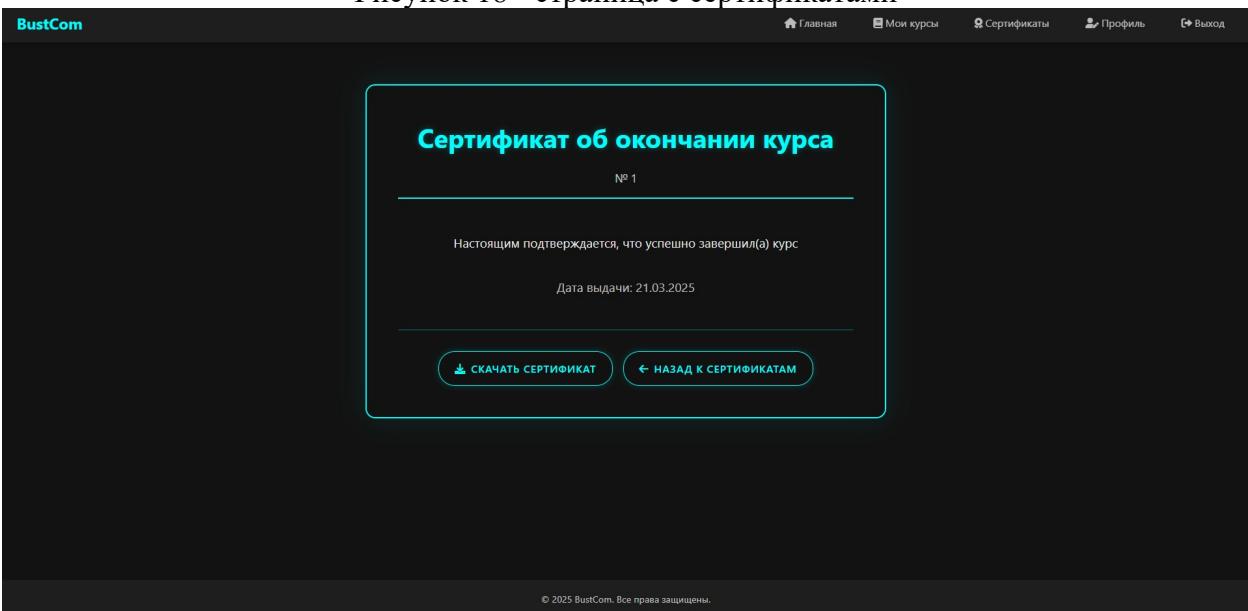


Рисунок 19 – сертификат

На рисунке 20 отображен профиль пользователя, в котором выводится информация о его роли, почте и имени, а также о его курсах

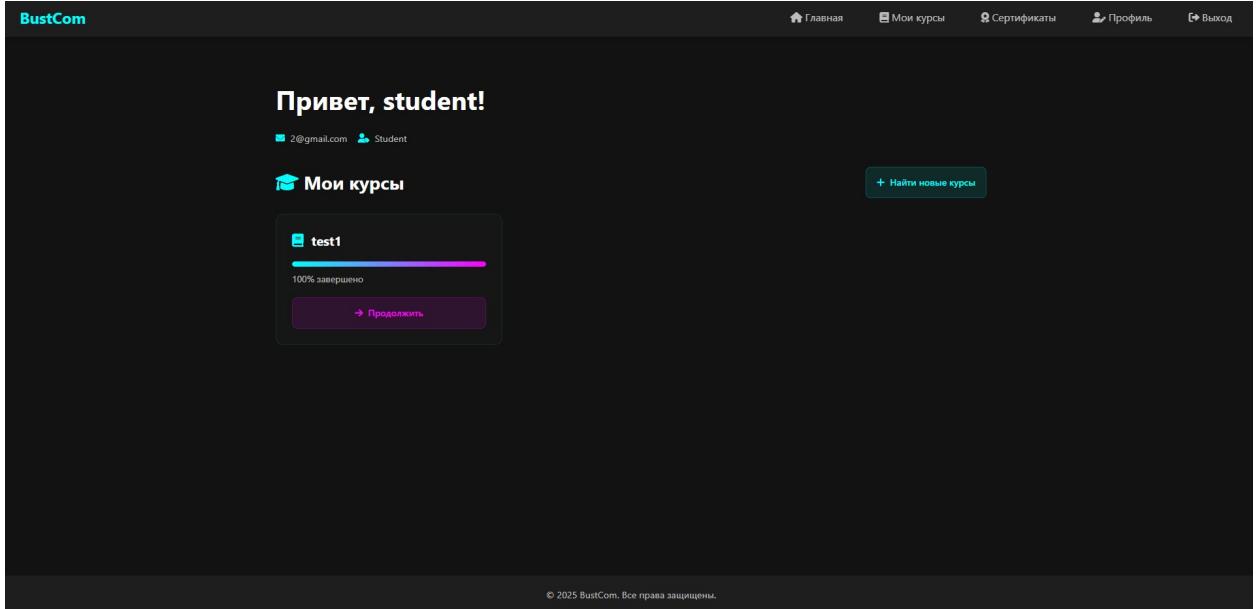


Рисунок 20 – профиль

3.3.1.3. Панель преподавателя.

На рисунке 21 отображена панель преподавателя. В ней присутствуют основные кнопки для управления лекциями и курсами.

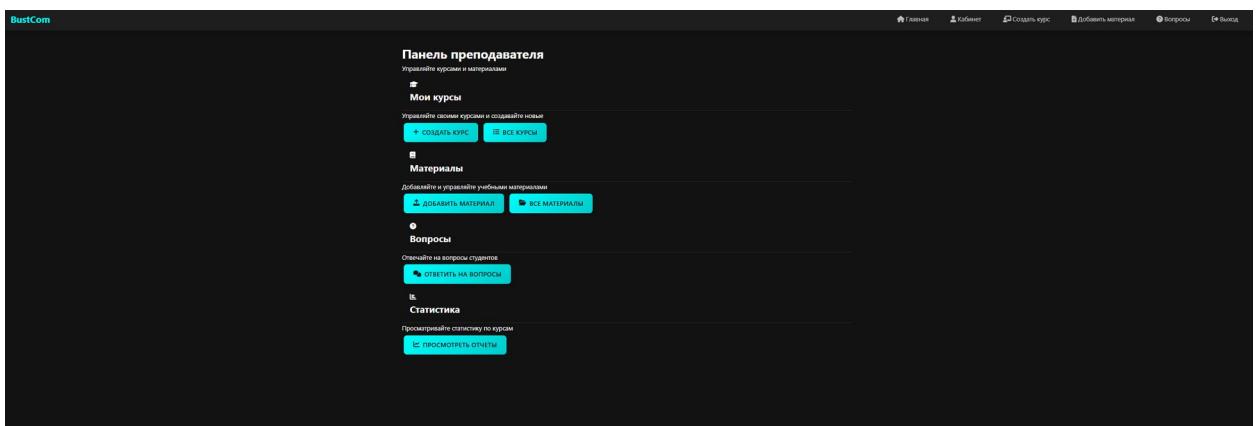


Рисунок 21 - панель преподавателя

На рис.22 отображена панель для создания новых курсов. Тут преподаватель выбирает название курса, его описание и продолжительность в часах, а также сложность курса. После нажатия кнопки «создать» курс будет отображаться для всех пользователей.

The screenshot shows a dark-themed web interface for creating a new course. At the top, there are navigation links: 'BustCom', 'Главная' (Home), 'Кабинет' (Cabinet), 'Создать курс' (Create course), 'Добавить материал' (Add material), 'Вопросы' (Questions), and 'Выход' (Logout). Below the header is a modal window titled 'Создать новый курс' (Create new course). The form fields include:

- Название курса**: A text input field with placeholder 'Введите название курса' (Enter course name).
- Описание**: A text input field with placeholder 'Введите описание курса' (Enter course description).
- Продолжительность (часов)**: A text input field with placeholder 'Продолжительность в часах' (Duration in hours) and a note below it: 'Укажите примерную продолжительность курса в часах' (Specify the approximate duration of the course in hours).
- Уровень сложности**: A dropdown menu set to 'Начальный' (Beginner) with a note below: 'Выберите уровень сложности курса' (Select the course difficulty level).

At the bottom of the modal are two buttons: 'СОЗДАТЬ КУРС' (Create course) and '× ОТМЕНА' (Cancel).

Рисунок 22 - панель для создания новых курсов

На рисунке 23 показана страница курсов, с которыми преподаватель может взаимодействовать. При нажатии на «подробнее» (рис.24-25) откроется панель для редактирования курсов, в которой преподаватель может менять основную информацию.

The screenshot shows a dark-themed web interface for managing courses. At the top, there are navigation links: 'BustCom', 'Главная' (Home), 'Кабинет' (Cabinet), 'Создать курс' (Create course), 'Добавить материал' (Add material), 'Вопросы' (Questions), and 'Выход' (Logout). Below the header is a section titled 'Список курсов' (List of courses). It displays two course cards:

Название	Продолжительность	Автор
text1	12 часов	astasid
astasid	1231 часов	astasid

Each card has a 'Подробнее' (More details) button and a 'Записаться' (Enroll) button.

Рисунок 23 - страница курсов

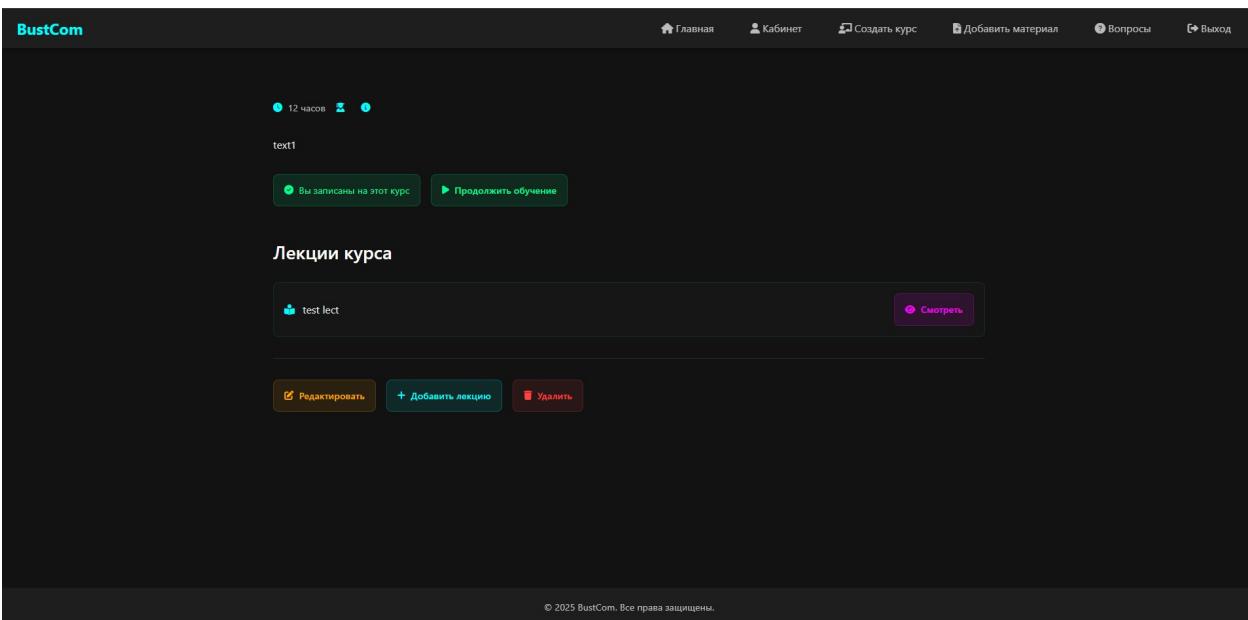


Рисунок 24 - панель для редактирования курсов

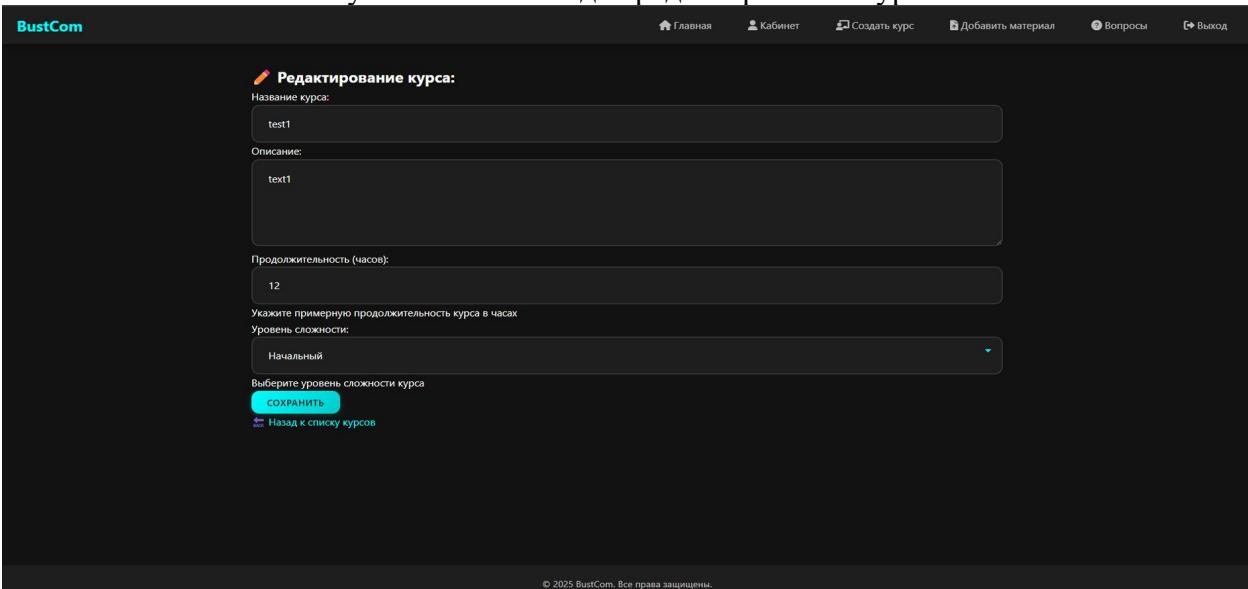


Рисунок 25 - панель для редактирования курсов

Преподаватель так же может добавлять лекции и материал для них. При нажатии «добавить материал» откроется страница, на которой можно выбрать лекцию, тип материала и ссылку на него (рис. 26)

ДОПОЛНИТЕЛЬНО

 Выход

Выберите лекцию

test1 - test lect

Тип материала

-- Выберите тип --

-- Выберите тип --

Файл

Видео

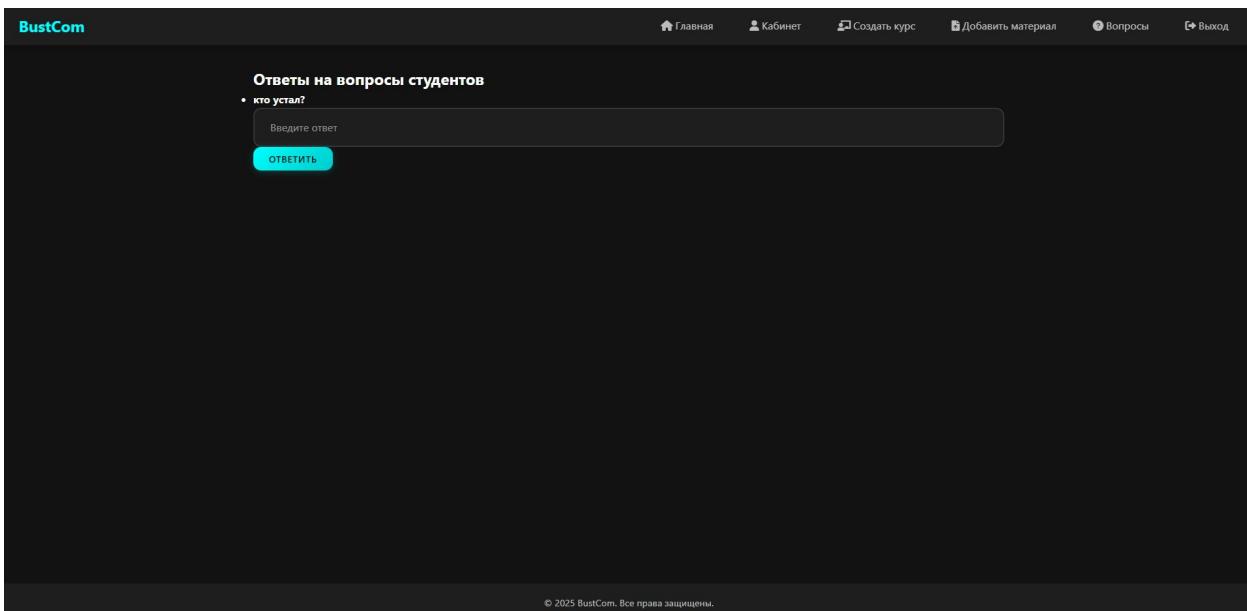
Введите URL файла или видео

Укажите прямую ссылку на файл или видео

 ДОБАВИТЬ МАТЕРИАЛ

Рисунок 26 - страница добавления материала

Преподаватель может отвечать на вопросы сотрудников. Для того чтобы открыть данную страницу надо нажать на «вопросы» в вверху панели преподавателя (рис.27)



© 2025 BustCom. Все права защищены.

Рисунок 27 - ответы на вопросы

Преподаватель также имеет права на редактирования тестов. Он может перейти на страницу теста в лекции и нажать «добавить вопрос» для добавления нового вопроса в тест к лекции (рис.28). Там он выбирает тест, вопрос, и способ ответа(рис.29).

The screenshot shows a dark-themed user interface for BustCom. At the top, there's a navigation bar with links for 'Главная', 'Кабинет', 'Создать курс', 'Добавить материал', 'Вопросы', and 'Выход'. Below the navigation, the main content area has a title 'test lect' and a subtitle 'Порядковый номер: 2'. Underneath, there's a section for 'Материалы:' with a link to 'Смотреть видео' and a 'Назад к лекциям' button. A 'Тесты' section contains a button to 'пройти ТЕСТ "TEST"'. Below this, there's a 'Тесты' list with a row for 'Пройти тест "test"' and a '+ Добавить вопрос' button. A 'Комментарии' section shows one comment from 'student' dated March 21, 2025, 7:30 a.m. A 'Добавить комментарий:' input field is present. At the bottom, there are several buttons: 'ОТПРАВИТЬ', 'ОТМЕНИТЬ КАК ПРОЙДЕННУЮ', 'РЕДАКТИРОВАТЬ ЛЕКЦИЮ', and 'УДАЛИТЬ ЛЕКЦИЮ'.

Рисунок 28 - страница лекции

This screenshot shows a 'Добавить вопрос в тест' (Add question to test) form. At the top, it says 'Test: test'. Below that is a 'Question text:' input field. Under 'Question type:', there's a dropdown menu with options: 'Один ответ' (One answer), 'Несколько ответов' (Multiple answers), and 'Ввод текста' (Text input). The 'Один ответ' option is currently selected. At the bottom of the form, there's a note: '© 2025 BustCom. Все права защищены.'

Рисунок 29 - добавление теста

После добавления вопроса преподаватель может добавить ответы. На рисунке 30 представлена страница добавления ответов, где выбирается вопрос, указывается текст ответа и выбор флагжка правильный ли он или нет.

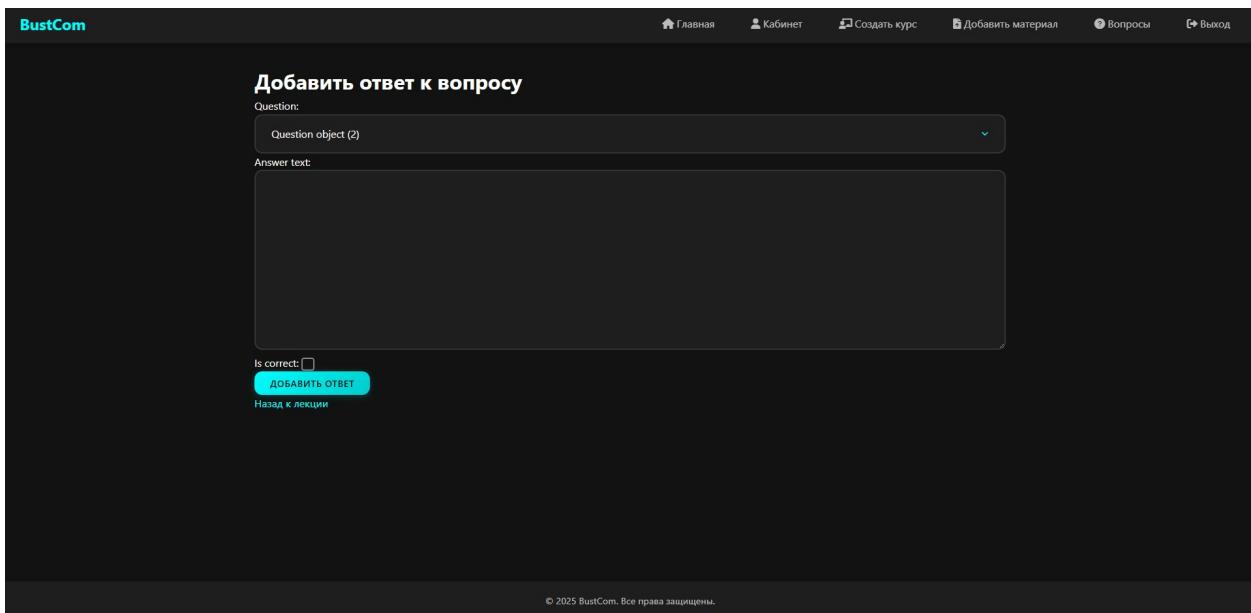


Рисунок 30 - добавление ответа

3.3.1.4. Панель hr.

На рисунке 31 представлена панель управления для hr. На ней пользователь с данной ролью может просматривать активность сотрудников, а также назначать курсы. Кроме этого, hr может еще рассматривать поданные заявки и давать ответ на них

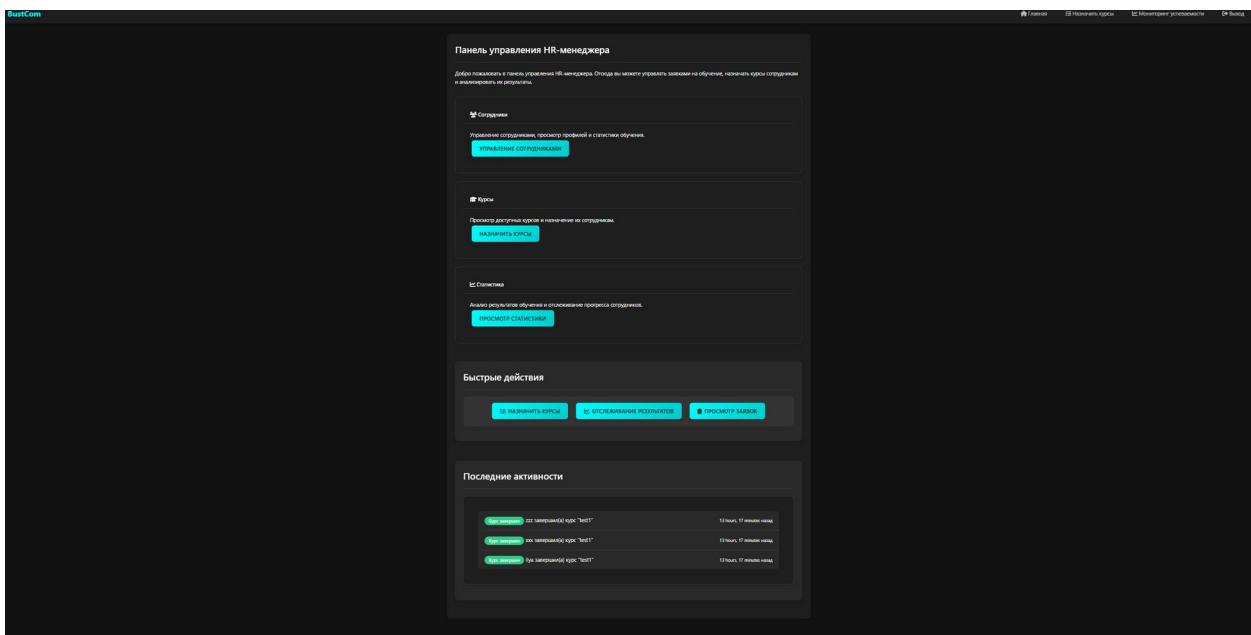


Рисунок 31 - панель управления для hr

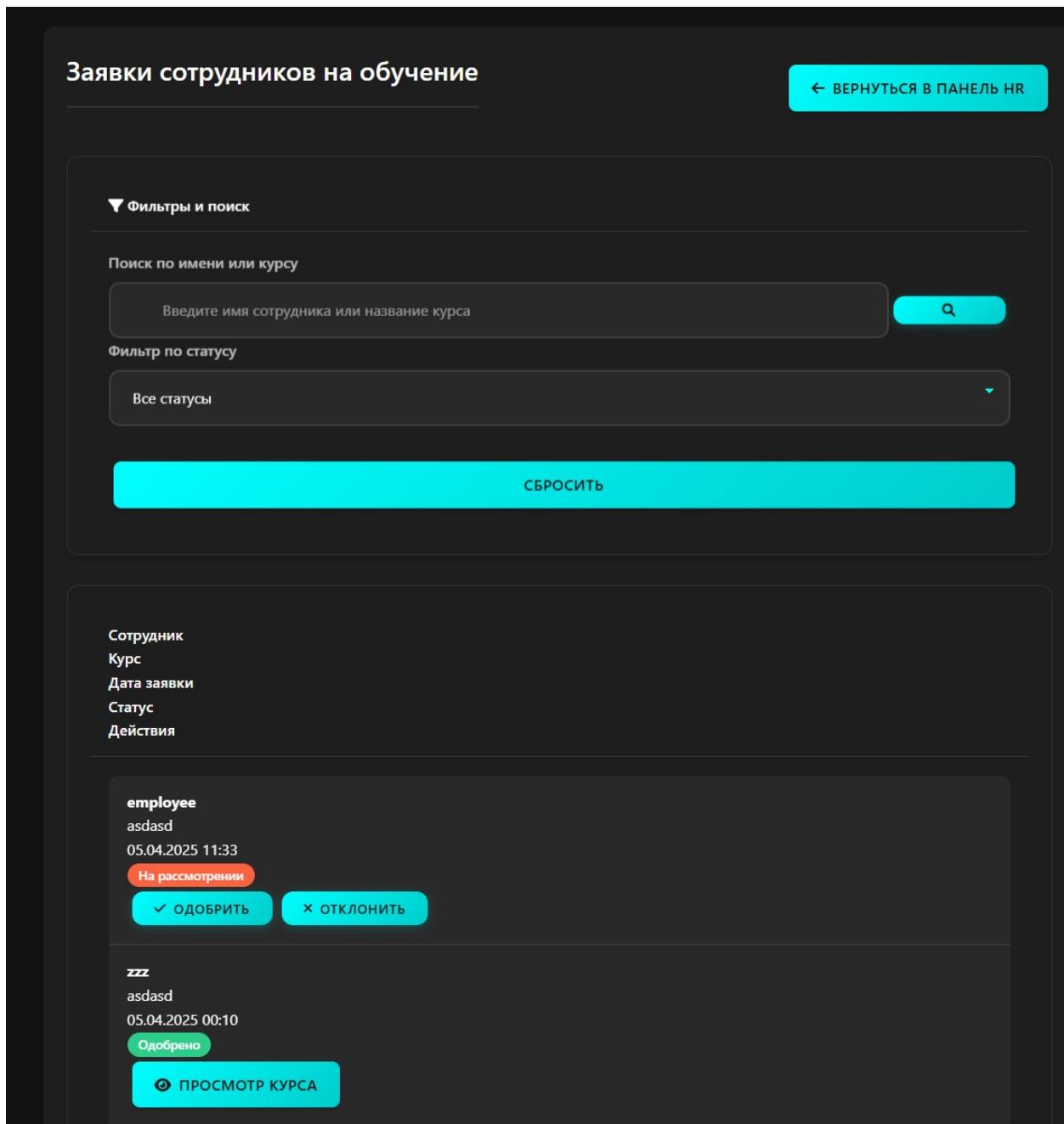


Рисунок 32 - заявки

Hr может назначать курсы. Для этого необходимо на главной странице нажать «назначить курсы». После чего откроется страница с выбором сотрудника и курсом. (рис. 33)

Назначение курсов

+ Назначить курс сотруднику

Выберите сотрудника

Выберите курс

+ НАЗНАЧИТЬ КУРС

☰ Текущие назначения

Сотрудник	Курс	Дата назначения	Статус	Действия
employee	test1	20.03.2025	Завершён	УДАЛИТЬ
student	test1	21.03.2025	Завершён	УДАЛИТЬ
teacher	test1	23.03.2025	В процессе	УДАЛИТЬ
zxc	test1	04.04.2025	Завершён	УДАЛИТЬ
zxc	asdasd	04.04.2025	В процессе	УДАЛИТЬ

Рисунок 33 - назначение курсов

Для просмотра успеваемости необходимо перейти в «мониторинг курсов», где отображается вся основная статистика по обучению.

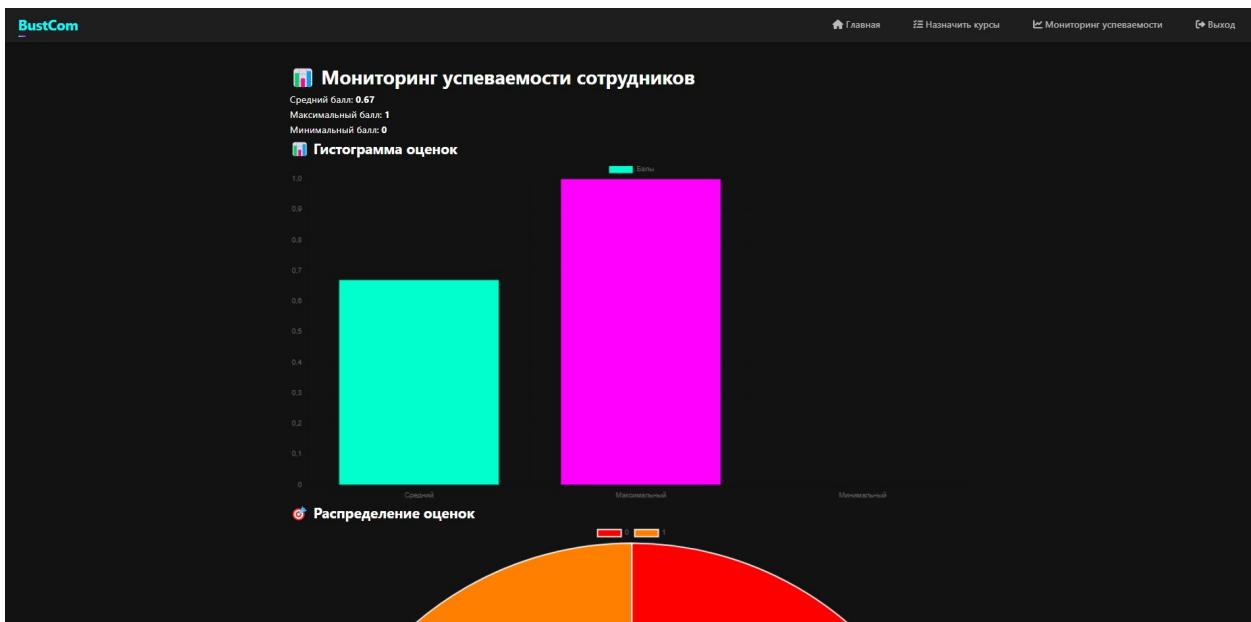


Рисунок 34 - мониторинг успеваемости

3.3.1.5. Панель менеджера

В панели менеджера пользователь может создавать планы обучения, выбирая курсы и пользователей, а также тоже просматривать заявки на обучения.

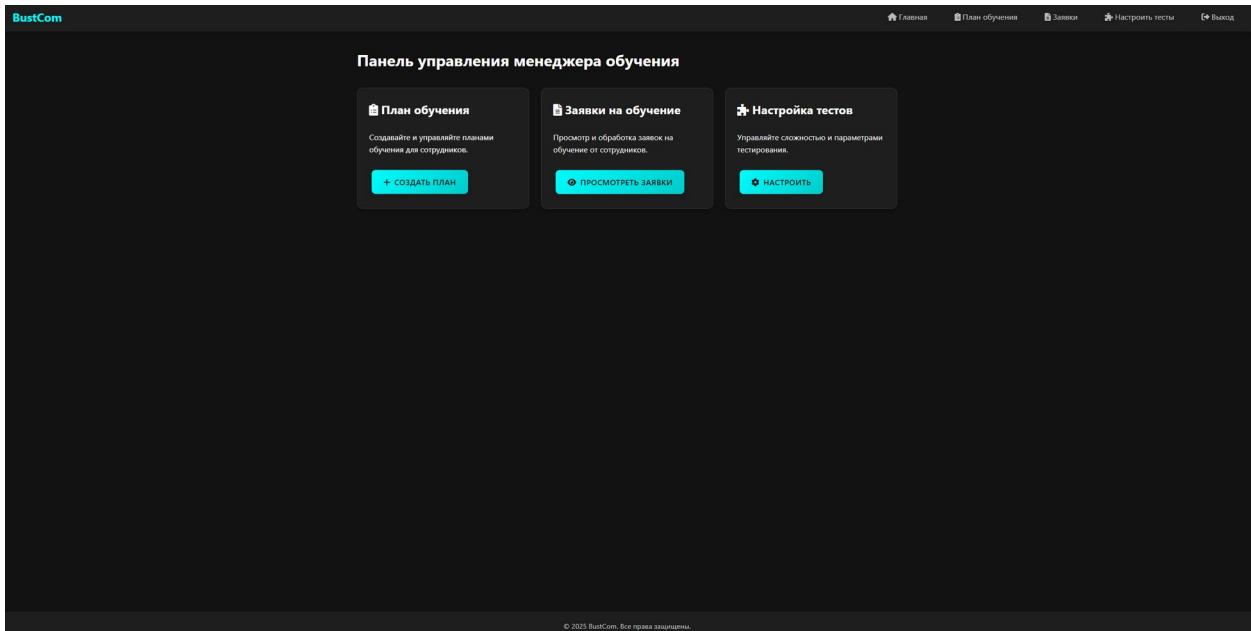


Рисунок 35 - панель менеджера

Просмотр заявок на обучение

Сотрудник	Курс	Дата заявки	Статус	Действия
employee	asdasd	05.04.2025	На рассмотрении	 
zzz	asdasd	05.04.2025	Одобрено	
zzz	test1	05.04.2025	Одобрено	
xxx	asdasd	05.04.2025	Одобрено	
xxx	test1	05.04.2025	Одобрено	
ilya	asdasd	04.04.2025	Одобрено	
ilya	test1	04.04.2025	Одобрено	
zxc	asdasd	04.04.2025	Одобрено	

Рисунок 36 - просмотр заявок

Для создания заявок на обучения необходимо заполнить форму с выбором сотрудников, дат, курсом.

Создание плана обучения

Новый план обучения

Название плана

Дата начала

 ДД.ММ.ГГГГ

Дата окончания

 ДД.ММ.ГГГГ

Статус плана

Черновик

Описание

Выберите курсы

- test1
- asdasd

Рисунок 37 - план обучения

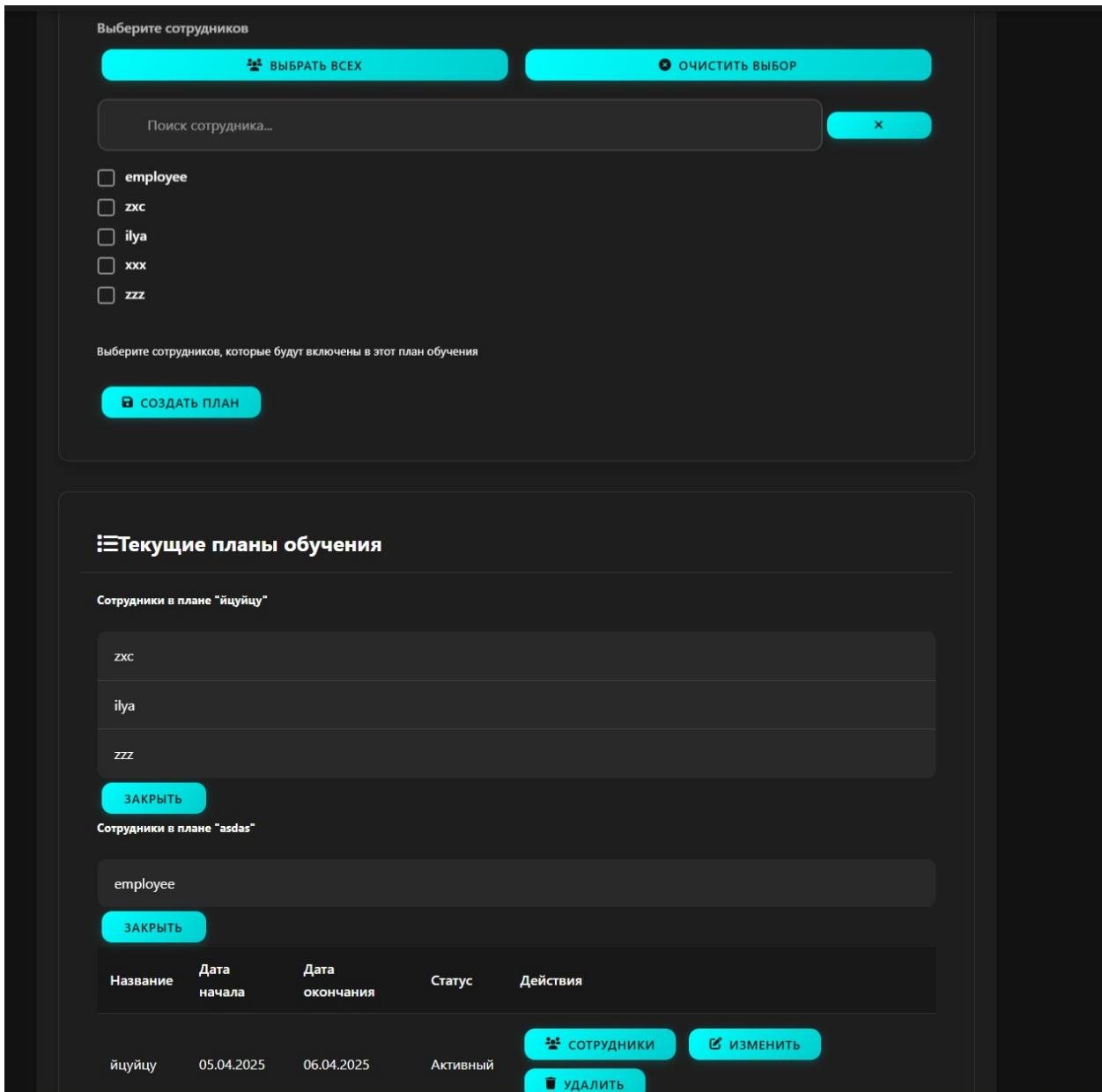


Рисунок 38 - план обучения

4. Возможные неполадки и способы устранения

Таблица 1 - Возможные неполадки и способы устранения

№	Неполадки	Способ устранения
1	Неудачное завершение установки встроенного ПО	Повторить установку. Актуализировать версию файла со встроенным ПО, повторить установку.
2	Прерывание интернет-соединения	Проверить подключение к интернету. 2. Перезагрузить страницу. 3. Если проблема сохраняется, проверить настройки сети.
3	Неверный формат email при регистрации	1. Проверить правильность ввода email. 2. Убедиться, что email содержит символ @ и домен. 3. Использовать корректный формат (например,

		user@domain.com).
4	Несовпадение паролей при регистрации	1. Проверить правильность ввода пароля в обоих полях. 2. Убедиться, что Caps Lock выключен. 3. Ввести пароль заново.
5	Превышение максимальной длины текста	1. Сократить текст до допустимой длины. 2. Разбить длинный текст на несколько частей. 3. Удалить лишние пробелы и переносы строк.
6	Загрузка файла недопустимого формата	1. Проверить список допустимых форматов файлов. 2. Конвертировать файл в поддерживаемый формат. 3. Уменьшить размер файла, если превышен лимит.
7	Истечение сессии пользователя	1. Войти в систему заново. 2. Сохранить данные перед выходом. 3. Увеличить время бездействия в настройках.
8	Ошибка валидации данных формы	1. Проверить все обязательные поля. 2. Исправить формат введенных данных. 3. Убедиться, что все поля заполнены корректно.
9	Ошибка при сохранении данных	1. Проверить подключение к интернету. 2. Попробовать сохранить данные позже. 3. Сохранить данные локально и повторить попытку.
10	Недоступность сервера	1. Проверить статус сервера. 2. Попробовать подключиться позже. 3. Обратиться в службу поддержки.
11	Конфликт версий данных	1. Обновить страницу. 2. Сохранить локальную копию данных. 3. Синхронизировать данные с сервером.
12	Ошибка при загрузке страницы	1. Очистить кэш браузера. 2. Попробовать другой браузер. 3. Проверить настройки безопасности браузера.

ПРИЛОЖЕНИЕ Д. СКРИПТ БАЗЫ ДАННЫХ

1. Описание структуры базы данных

База данных "db.sqlite3" разработана для хранения информации и управления данными системы корпоративного обучения. Структура базы данных включает несколько взаимосвязанных таблиц, каждая из которых выполняет отдельную роль для обеспечения функциональности системы.

Описание таблиц:

- User — Хранит данные о пользователях системы с различными ролями (администратор, HR-менеджер, преподаватель, менеджер по обучению, сотрудник).
- Role — Содержит информацию о доступных ролях в системе.
- Profile — Расширенная информация о пользователях, включая контактные данные и принадлежность к компании.
- Course — Хранит данные об учебных курсах, включая название, описание, продолжительность и уровень сложности.
- CourseTeacher — Связывает курсы с преподавателями, которые их ведут.
- Lecture — Содержит информацию о лекциях в рамках курсов.
- LectureResource — Хранит учебные материалы, прикрепленные к лекциям.
- Enrollment — Регистрирует записи пользователей на курсы и отслеживает их статус обучения.

- Progress — Отслеживает прогресс пользователей по каждой лекции курса.
- Test — Содержит информацию о тестах, их сложности и количестве попыток.
- Question — Хранит вопросы тестов различных типов (один ответ, несколько ответов, ввод текста).
- Answer — Содержит варианты ответов на вопросы и отметки о правильности.
- Attempt — Регистрирует попытки прохождения тестов пользователями.
- TestAttempt — Детальная информация о результатах тестирования.
- Certificate — Хранит данные о выданных сертификатах по завершении курсов.
- CourseLog — Журнал действий в системе, связанных с курсами (запись, прогресс, завершение, получение сертификата).
- CompletedLecture — Отмечает завершенные пользователями лекции.
- Notification — Система уведомлений для пользователей.
- TrainingRequest — Хранит заявки на обучение от сотрудников.
- TrainingPlan — Содержит планы обучения с привязкой к курсам и сотрудникам.

2. Структура базы данных

2.1. Логическая схема

Логическая модель базы данных построена с учетом нормализации данных до третьей нормальной формы, что позволяет снизить избыточность данных и поддерживать целостность. Взаимосвязи между таблицами реализованы через связи "один ко многим" и "многие ко многим", обеспечивая корректное хранение информации о пользователях, ролях, товарах, логов и др.

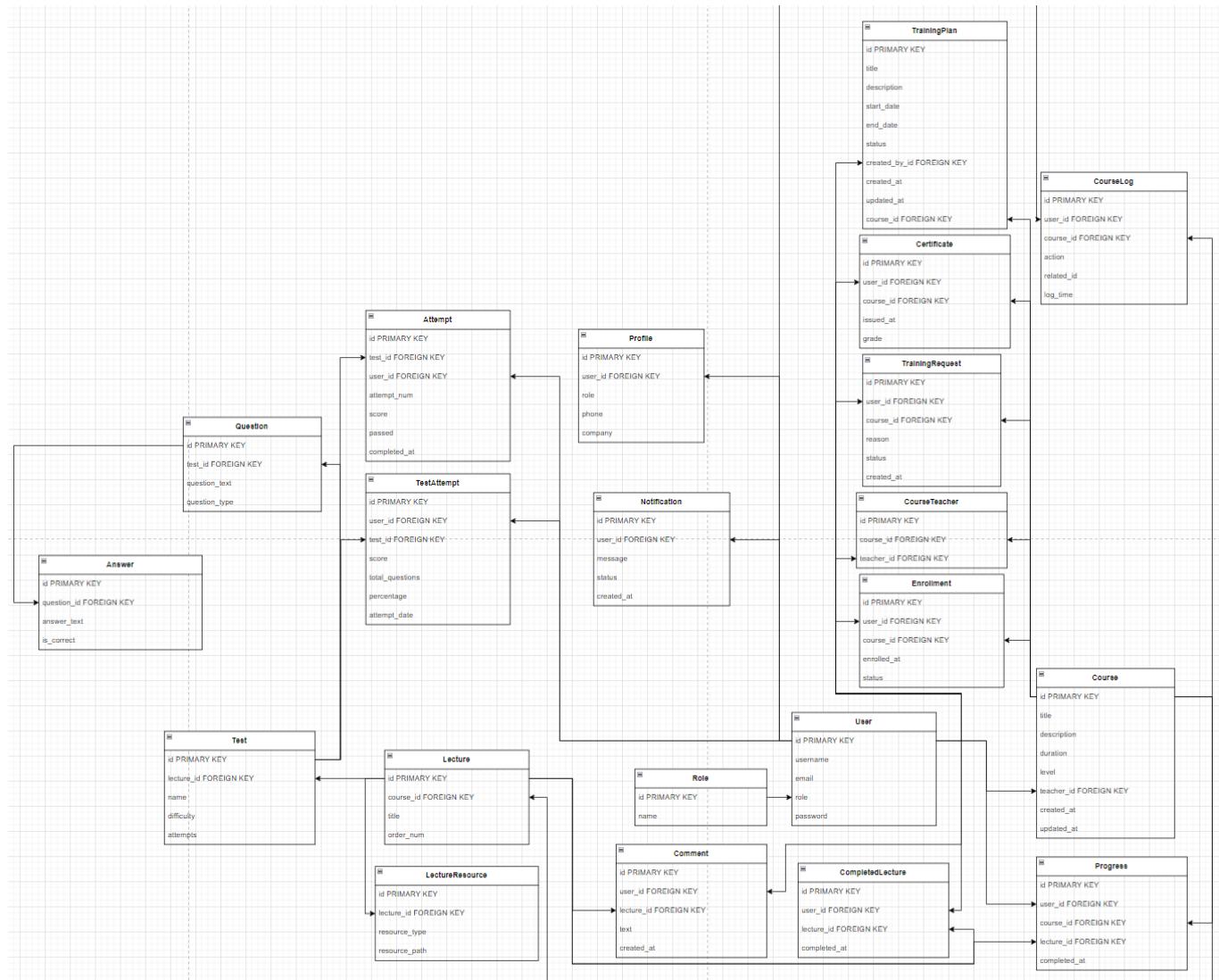


Рисунок 1 – логическая модель базы данных

2.2. Физическая схема

Физическая модель реализована в СУБД SQLite с учетом особенностей хранения и обработки данных.

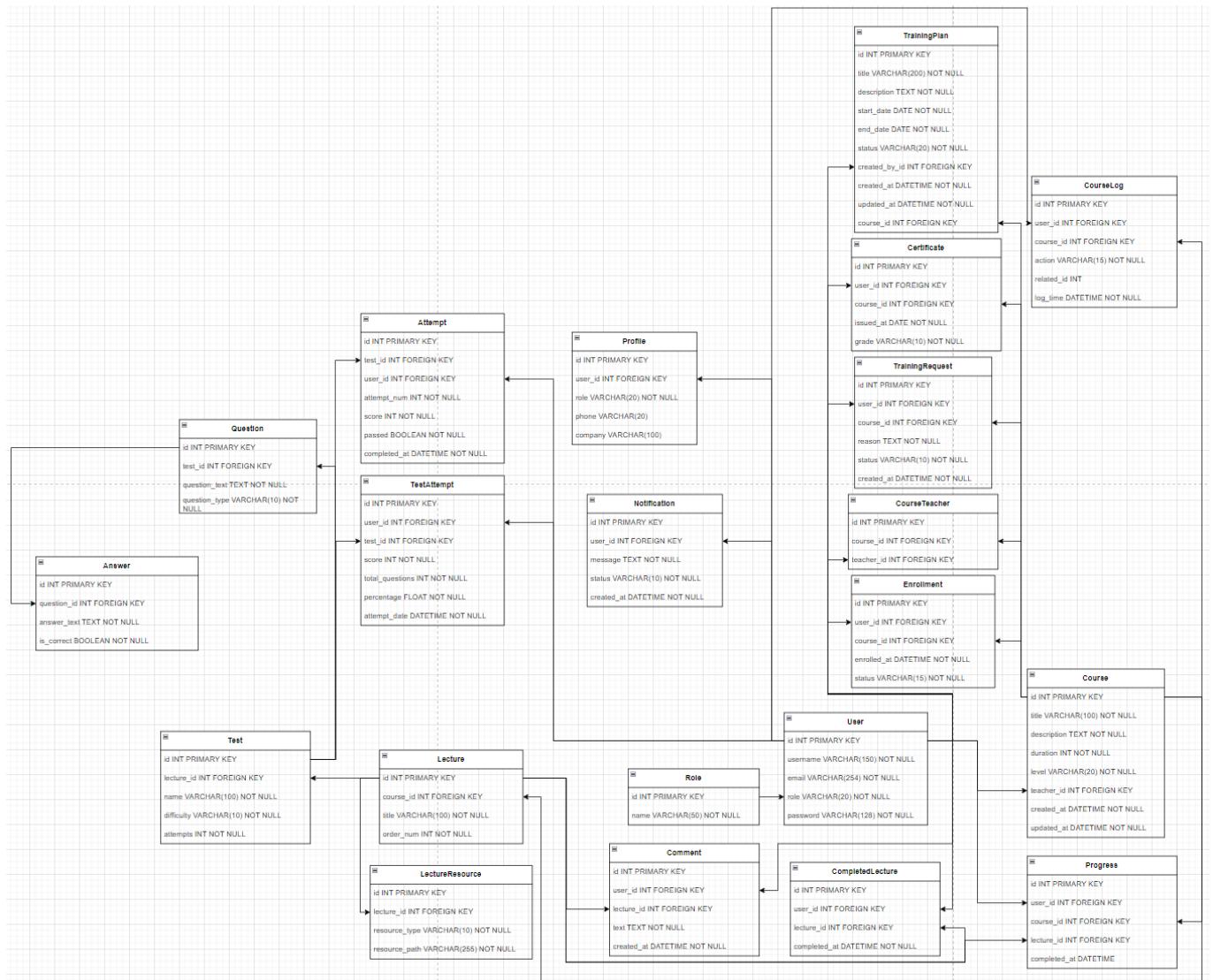


Рисунок 2 – физическая модель данных

2.3. Скрипт базы данных

```
from django.db import models
from django.conf import settings
from django.contrib.auth.models import AbstractUser, BaseUserManager
from django.db import models

class Role(models.Model):
    name = models.CharField(max_length=50, unique=True)

    def __str__(self):
        return self.name

class UserManager(BaseUserManager):
    def create_user(self, username, email, password=None, **extra_fields):
        if not email:
            raise ValueError("Email обязательен!")
        email = self.normalize_email(email)
        user = self.model(username=username, email=email, **extra_fields)
        user.set_password(password)
        user.save(using=self._db)
        return user

    def create_superuser(self, username, email, password=None, **extra_fields):
        extra_fields.setdefault('is_staff', True)
        extra_fields.setdefault('is_superuser', True)

        return self.create_user(username, email, password, **extra_fields)

class User(AbstractUser):
    ROLE_CHOICES = [
        ('admin', 'Администратор'),
        ('hr', 'HR-менеджер'),
        ('teacher', 'Преподаватель'),
        ('training_manager', 'Менеджер по обучению'),
        ('employee', 'Сотрудник')
    ]

    email = models.EmailField(unique=True)
    role = models.CharField(max_length=20, choices=ROLE_CHOICES, default='employee')

    objects = UserManager()

    groups = models.ManyToManyField(
        'auth.Group',
        related_name="bustcomproj_user_groups",
        blank=True
    )
    user_permissions = models.ManyToManyField(
        'auth.Permission',
        related_name="bustcomproj_user_permissions",
        blank=True
    )

    def is_hr(self):
        return self.role == 'hr'

    def is_teacher(self):
        return self.role == 'teacher'
```

```

def is_training_manager(self):
    return self.role == 'training_manager'

def is_employee(self):
    return self.role == 'employee'

def __str__(self):
    return f'{self.username} - {self.get_role_display()}'


class Course(models.Model):
    LEVEL_CHOICES = [
        ('beginner', 'Начальный'),
        ('intermediate', 'Средний'),
        ('advanced', 'Продвинутый')
    ]

    title = models.CharField(max_length=100)
    description = models.TextField()
    duration = models.IntegerField(help_text='Продолжительность в часах')
    level = models.CharField(max_length=20, choices=LEVEL_CHOICES, default='beginner')
    teacher = models.ForeignKey(User, on_delete=models.CASCADE, related_name='courses')
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.title


class CourseTeacher(models.Model):
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    teacher = models.ForeignKey(User, on_delete=models.CASCADE)

    class Meta:
        unique_together = ('course', 'teacher')


class Lecture(models.Model):
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    title = models.CharField(max_length=100)
    order_num = models.IntegerField()

    def __str__(self):
        return self.title


class LectureResource(models.Model):
    lecture = models.ForeignKey(Lecture, on_delete=models.CASCADE)
    resource_type = models.CharField(max_length=10, choices=[('file', 'Файл'), ('video', 'Видео')])
    resource_path = models.CharField(max_length=255)

    def __str__(self):
        return f'{self.lecture.title} - {self.resource_type}'


class Enrollment(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    enrolled_at = models.DateTimeField(auto_now_add=True)
    status_choices = [('в процессе', 'В процессе'), ('завершён', 'Завершён'), ('отчислен', 'Отчислен')]


```

```

status = models.CharField(max_length=15, choices=status_choices, default='в процессе')

class Progress(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    lecture = models.ForeignKey(Lecture, on_delete=models.CASCADE)
    completed_at = models.DateTimeField(null=True, blank=True)

class Test(models.Model):
    lecture = models.ForeignKey(Lecture, on_delete=models.CASCADE)
    name = models.CharField(max_length=100)
    difficulty_choices = [('низкая', 'Низкая'), ('средняя', 'Средняя'), ('высокая', 'Высокая')]
    difficulty = models.CharField(max_length=10, choices=difficulty_choices)
    attempts = models.IntegerField(default=3)

    def __str__(self):
        return self.name

class Question(models.Model):
    test = models.ForeignKey(Test, on_delete=models.CASCADE)
    question_text = models.TextField()
    question_type_choices = [('один', 'Один ответ'), ('несколько', 'Несколько ответов'), ('ввод', 'Ввод текста')]
    question_type = models.CharField(max_length=10, choices=question_type_choices)

class Answer(models.Model):
    question = models.ForeignKey(Question, on_delete=models.CASCADE)
    answer_text = models.TextField()
    is_correct = models.BooleanField()

class Attempt(models.Model):
    test = models.ForeignKey(Test, on_delete=models.CASCADE)
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    attempt_num = models.IntegerField()
    score = models.IntegerField()
    passed = models.BooleanField()
    completed_at = models.DateTimeField(auto_now_add=True)

class Certificate(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    issued_at = models.DateField(auto_now_add=True)
    grade = models.CharField(max_length=10)

class Notification(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    message = models.TextField()
    status_choices = [('новое', 'Новое'), ('прочитано', 'Прочитано')]
    status = models.CharField(max_length=10, choices=status_choices, default='новое')
    created_at = models.DateTimeField(auto_now_add=True)

class CourseLog(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    course = models.ForeignKey(Course, on_delete=models.CASCADE)
    action_choices = [('запись', 'Запись'), ('прогресс', 'Прогресс'), ('завершение', 'Завершение'), ('сертификат', 'Сертификат')]
    action = models.CharField(max_length=15, choices=action_choices)
    related_id = models.IntegerField(null=True, blank=True)
    log_time = models.DateTimeField(auto_now_add=True)

```

```

class CompletedLecture(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    lecture = models.ForeignKey(Lecture, on_delete=models.CASCADE)
    completed_at = models.DateTimeField(auto_now_add=True)

    class Meta:
        unique_together = ('user', 'lecture')

    def __str__(self):
        return f'{self.user.username} - {self.lecture.title}'


class Profile(models.Model):
    ROLE_CHOICES = [
        ('admin', 'Администратор'),
        ('hr', 'HR-менеджер'),
        ('teacher', 'Преподаватель'),
        ('training_manager', 'Менеджер по обучению'),
        ('employee', 'Сотрудник')
    ]
    user = models.OneToOneField(User, on_delete=models.CASCADE)
    role = models.CharField(max_length=20, choices=ROLE_CHOICES, default='employee')
    phone = models.CharField(max_length=20, blank=True, null=True)
    company = models.CharField(max_length=100, blank=True, null=True)

    def __str__(self):
        return f'{self.user.username} - {self.get_role_display()}'


from django.db import models
from django.contrib.auth.models import User

from django.conf import settings

class TestAttempt(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    test = models.ForeignKey(Test, on_delete=models.CASCADE)
    score = models.IntegerField()
    total_questions = models.IntegerField()
    percentage = models.FloatField()
    attempt_date = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f'{self.user.username} - {self.test.lecture.title} - {self.percentage}%'


class Comment(models.Model):
    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    lecture = models.ForeignKey(Lecture, on_delete=models.CASCADE, related_name='comments')
    text = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return f'{self.user.username} - {self.lecture.title} - {self.created_at}'


from django.db import models
from django.conf import settings

class TrainingRequest(models.Model):
    STATUS_CHOICES = [

```

```

('pending', 'Ожидание'),
('approved', 'Одобрено'),
('rejected', 'Отклонено')
]

user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
course = models.ForeignKey('Course', on_delete=models.CASCADE)
reason = models.TextField()
status = models.CharField(max_length=10, choices=STATUS_CHOICES, default='pending')
created_at = models.DateTimeField(auto_now_add=True)

def __str__(self):
    return f'{self.user.username} - {self.course.title} ({self.get_status_display()})'

from django.db.models.signals import post_save
from django.dispatch import receiver
from .models import Profile, User

@receiver(post_save, sender=User)
def create_user_profile(sender, instance, created, **kwargs):
    if created:
        Profile.objects.get_or_create(user=instance, defaults={'role': instance.role})

class TrainingPlan(models.Model):
    """Training plan for employees."""
    STATUS_CHOICES = (
        ('draft', 'Черновик'),
        ('active', 'Активный'),
        ('completed', 'Завершен'),
        ('cancelled', 'Отменен'),
    )

    title = models.CharField(max_length=200)
    description = models.TextField()
    start_date = models.DateField()
    end_date = models.DateField()
    status = models.CharField(max_length=20, choices=STATUS_CHOICES, default='draft')
    created_by = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now=True)

    courses = models.ManyToManyField('Course', related_name='training_plans')
    employees = models.ManyToManyField(settings.AUTH_USER_MODEL,
                                      related_name='assigned_training_plans')

    def __str__(self):
        return self.title

class ActivityLog(models.Model):
    """Log of user activities in the system."""
    ACTION_CHOICES = (
        ('запись', 'Запись на курс'),
        ('отчислен', 'Отчисление с курса'),
        ('завершение', 'Завершение курса'),
        ('прогресс', 'Прогресс по курсу'),
    )

    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)
    course = models.ForeignKey('Course', on_delete=models.CASCADE, null=True)
    action = models.CharField(max_length=20, choices=ACTION_CHOICES)
    details = models.TextField(blank=True, null=True)
    log_time = models.DateTimeField(auto_now=True)

```

```
class Meta:  
    ordering = ['-log_time']  
  
def __str__(self):  
    return f'{self.user.username} - {self.action} - {self.log_time.strftime("%d.%m.%Y %H:%M")}'
```

2.4. Словарь данных

Таблица 1 - Таблица Role (Роли пользователей)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор роли	PRIMARY KEY	1
name	VARCHAR(50)	Название роли	NOT NULL, UNIQUE	'teacher'

Описание полей:

- id: Уникальный идентификатор роли, основной ключ таблицы.

- name: Название роли, обязательное для заполнения, уникальное.

Таблица 2 - Таблица User (Пользователи)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор пользователя	PRIMARY KEY	1
username	VARCHAR(150)	Имя пользователя	NOT NULL, UNIQUE	'ivan.petrov'
email	VARCHAR(254)	Электронная почта	NOT NULL, UNIQUE	'ivan@example.com'
role	VARCHAR(20)	Роль пользователя	NOT NULL	'teacher'
password	VARCHAR(128)	Пароль	NOT NULL	'hashed_password'

Описание полей:

- id: Основной ключ, уникальный идентификатор пользователя.

- username: Имя пользователя, используется для входа в систему и должно быть уникальным.

- email: Адрес электронной почты, также уникален для каждого пользователя.

- role: Роль пользователя, выбирается из предопределенных значений.

- password: Пароль пользователя.

Таблица 3 - Таблица Course (Курсы)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор курса	PRIMARY KEY	1
title	VARCHAR(100)	Название курса	NOT NULL	'Основы Python'
description	TEXT	Описание курса		'Описание курса'
duration	INTEGER	Продолжительность курса (в часах)	NOT NULL	10
level	VARCHAR(20)	Уровень курса	NOT NULL	'beginner'
teacher_id	INTEGER	Преподаватель курса	FOREIGN KEY -> User.id	1
created_at	TIMESTAMP	Дата создания курса	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'
updated_at	TIMESTAMP	Дата обновления курса	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор курса.
- title: Название курса, обязательное для заполнения.
- description: Описание курса.
- duration: Продолжительность курса в часах.
- level: Уровень курса (например, 'beginner', 'intermediate', 'advanced').
- teacher_id: Ссылка на таблицу User, определяющая преподавателя курса.
- created_at: Дата и время создания курса.
- updated_at: Дата и время последнего обновления курса.

Таблица 4 - Таблица CourseTeacher (Преподаватели курсов)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор записи	PRIMARY KEY	1
course_id	INTEGER	Курс	FOREIGN KEY -> Course.id	1
teacher_id	INTEGER	Преподаватель	FOREIGN KEY -> User.id	2

Описание полей:

- id: Уникальный идентификатор записи.
- course_id: Ссылка на таблицу Course, определяющая курс.
- teacher_id: Ссылка на таблицу User, определяющая преподавателя.

Таблица 5 - Таблица Lecture (Лекции)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор лекции	PRIMARY KEY	1
course_id	INTEGER	Курс	FOREIGN KEY -> Course.id	1
title	VARCHAR(100)	Название лекции	NOT NULL	'Введение в Python'
order_num	INTEGER	Порядковый номер лекции	NOT NULL	1

Описание полей:

- id: Уникальный идентификатор лекции.
- course_id: Ссылка на таблицу Course, определяющая курс.
- title: Название лекции, обязательное для заполнения.
- order_num: Порядковый номер лекции в курсе.

Таблица 6 - Таблица LectureResource (Ресурсы лекций)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор ресурса	PRIMARY KEY	1
lecture_id	INTEGER	Лекция	FOREIGN KEY -> Lecture.id	1
resource_type	VARCHAR(10)	Тип ресурса	NOT NULL	'file'
resource_path	VARCHAR(255)	Путь к ресурсу	NOT NULL	'/path/to/resource'

Описание полей:

- id: Уникальный идентификатор ресурса.
- lecture_id: Ссылка на таблицу Lecture, определяющая лекцию.
- resource_type: Тип ресурса (например, 'file', 'video').
- resource_path: Путь к ресурсу.

Таблица 7 - Таблица Enrollment (Записи на курсы)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор записи	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
course_id	INTEGER	Курс	FOREIGN KEY -> Course.id	1
enrolled_at	TIMESTAMP	Дата записи на курс	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'
status	VARCHAR(15)	Статус записи	NOT NULL	'в процессе'

Описание полей:

- id: Уникальный идентификатор записи.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- course_id: Ссылка на таблицу Course, определяющая курс.
- enrolled_at: Дата и время записи на курс.
- status: Статус записи (например, 'в процессе', 'завершён', 'отчислен').

Таблица 8 - Таблица Progress (Прогресс)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор прогресса	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
course_id	INTEGER	Курс	FOREIGN KEY -> Course.id	1
lecture_id	INTEGER	Лекция	FOREIGN KEY -> Lecture.id	1
completed_at	TIMESTAMP	Дата завершения лекции		'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор прогресса.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- course_id: Ссылка на таблицу Course, определяющая курс.
- lecture_id: Ссылка на таблицу Lecture, определяющая лекцию.
- completed_at: Дата и время завершения лекции.

Таблица 9 - Таблица Test (Тесты)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор теста	PRIMARY KEY	1
lecture_id	INTEGER	Лекция	FOREIGN KEY -> Lecture.id	1
name	VARCHAR(100)	Название теста	NOT NULL	'Тест 1'
difficulty	VARCHAR(10)	Сложность теста	NOT NULL	'низкая'
attempts	INTEGER	Количество попыток	DEFAULT 3	3

Описание полей:

- id: Уникальный идентификатор теста.
- lecture_id: Ссылка на таблицу Lecture, определяющая лекцию.
- name: Название теста, обязательное для заполнения.
- difficulty: Сложность теста (например, 'низкая', 'средняя', 'высокая').
- attempts: Количество попыток для прохождения теста.

Таблица 10 - Таблица Question (Вопросы)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор вопроса	PRIMARY KEY	1
test_id	INTEGER	Тест	FOREIGN KEY -> Test.id	1
question_text	TEXT	Текст вопроса	NOT NULL	'Что такое Python?'
question_type	VARCHAR(10)	Тип вопроса	NOT NULL	'один'

Описание полей:

- id: Уникальный идентификатор вопроса.
- test_id: Ссылка на таблицу Test, определяющая тест.
- question_text: Текст вопроса, обязательное для заполнения.
- question_type: Тип вопроса (например, 'один', 'несколько', 'ввод').

Таблица 11 - Таблица Answer (Ответы)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор ответа	PRIMARY KEY	1
question_id	INTEGER	Вопрос	FOREIGN KEY -> Question.id	1
answer_text	TEXT	Текст ответа	NOT NULL	'Python — это язык программирования.'
is_correct	BOOLEAN	Правильность ответа	NOT NULL	TRUE

Описание полей:

- id: Уникальный идентификатор ответа.
- question_id: Ссылка на таблицу Question, определяющая вопрос.
- answer_text: Текст ответа, обязательное для заполнения.
- is_correct: показывает, является ли ответ правильным.

Таблица 12 - Таблица Attempt (Попытки)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор попытки	PRIMARY KEY	1
test_id	INTEGER	Тест	FOREIGN KEY -> Test.id	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
attempt_num	INTEGER	Номер попытки	NOT NULL	1
score	INTEGER	Оценка	NOT NULL	80
passed	BOOLEAN	Показатель успешности	NOT NULL	TRUE
completed_at	TIMESTAMP	Дата завершения попытки	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор попытки.
- test_id: Ссылка на таблицу Test, определяющая тест.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- attempt_num: Номер попытки.
- score: Оценка за попытку.
- passed: Показывает, успешно ли пройден тест.
- completed_at: Дата и время завершения попытки.

Таблица 13 - Таблица Certificate (Сертификаты)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор сертификата	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
course_id	INTEGER	Курс	FOREIGN KEY -> Course.id	1
issued_at	DATE	Дата выдачи сертификата	DEFAULT CURRENT_DATE	'2024-10-15'
grade	VARCHAR(10)	Оценка	NOT NULL	'A'

Описание полей:

- id: Уникальный идентификатор сертификата.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- course_id: Ссылка на таблицу Course, определяющая курс.
- issued_at: Дата выдачи сертификата.
- grade: Оценка за курс.

Таблица 14 - Таблица Notification (Уведомления)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор уведомления	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
message	TEXT	Текст уведомления	NOT NULL	'Новое уведомление'
status	VARCHAR(10)	Статус уведомления	NOT NULL	'новое'
created_at	TIMESTAMP	Дата создания уведомления	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор уведомления.
- user_id: Ссылка на таблицу User, определяющая пользователя.

- message: Текст уведомления, обязательное для заполнения.
- status: Статус уведомления (например, 'новое', 'прочитано').
- created_at: Дата и время создания уведомления.

Таблица 15 - Таблица CourseLog (Логи курсов)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор лога	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
course_id	INTEGER	Курс	FOREIGN KEY -> Course.id	1
action	VARCHAR(15)	Действие	NOT NULL	'запись'
related_id	INTEGER	Связанный идентификатор		1
log_time	TIMESTAMP	Время записи лога	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор лога.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- course_id: Ссылка на таблицу Course, определяющая курс.
- action: Действие (например, 'запись', 'прогресс', 'завершение', 'сертификат').
- related_id: Связанный идентификатор (например, ID записи, прогресса и т.д.).
- log_time: Дата и время записи лога.

Таблица 16 - Таблица CompletedLecture (Завершенные лекции)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор завершенной	PRIMARY KEY	1

		лекции		
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
lecture_id	INTEGER	Лекция	FOREIGN KEY -> Lecture.id	1
completed_at	TIMESTAMP	Дата завершения лекции	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор завершенной лекции.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- lecture_id: Ссылка на таблицу Lecture, определяющая лекцию.
- completed_at: Дата и время завершения лекции.

Таблица 17 - Таблица Profile (Профили пользователей)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор профиля	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
role	VARCHAR(20)	Роль пользователя	NOT NULL	'teacher'
phone	VARCHAR(20)	Телефон		'+79991234567'
company	VARCHAR(100)	Компания		'Компания 1'

Описание полей:

- id: Уникальный идентификатор профиля.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- role: Роль пользователя (например, 'teacher', 'employee').
- phone: Телефон пользователя.
- company: Компания пользователя.

Таблица 18 - Таблица TestAttempt (Попытки тестов)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор попытки теста	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
test_id	INTEGER	Тест	FOREIGN KEY -> Test.id	1
score	INTEGER	Оценка	NOT NULL	80
total_questions	INTEGER	Общее количество вопросов	NOT NULL	10
percentage	FLOAT	Процент правильных ответов	NOT NULL	80.0
attempt_date	TIMESTAMP	Дата попытки	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор попытки теста.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- test_id: Ссылка на таблицу Test, определяющая тест.
- score: Оценка за попытку.
- total_questions: Общее количество вопросов в тесте.
- percentage: Процент правильных ответов.
- attempt_date: Дата и время попытки.

Таблица 19 - Таблица Comment (Комментарии)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор комментария	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
lecture_id	INTEGER	Лекция	FOREIGN KEY ->	1

			Lecture.id	
text	TEXT	Текст комментария	NOT NULL	'Отличная лекция!'
created_at	TIMESTAMP	Дата создания комментария	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор комментария.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- lecture_id: Ссылка на таблицу Lecture, определяющая лекцию.
- text: Текст комментария, обязательное для заполнения.
- created_at: Дата и время создания комментария.

Таблица 20 - Таблица TrainingRequest (Запросы на обучение)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор запроса	PRIMARY KEY	1
user_id	INTEGER	Пользователь	FOREIGN KEY -> User.id	1
course_id	INTEGER	Курс	FOREIGN KEY -> Course.id	1
reason	TEXT	Причина запроса	NOT NULL	'Хочу изучить Python'
status	VARCHAR(10)	Статус запроса	NOT NULL	'pending'
created_at	TIMESTAMP	Дата создания запроса	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор запроса.
- user_id: Ссылка на таблицу User, определяющая пользователя.
- course_id: Ссылка на таблицу Course, определяющая курс.

- reason: Причина запроса, обязательное для заполнения.
- status: Статус запроса (например, 'pending', 'approved', 'rejected').
- created_at: Дата и время создания запроса.

Таблица 21 - Таблица TrainingPlan (Планы обучения)

Поле	Тип данных	Описание	Ограничения	Пример данных
id	SERIAL	Уникальный идентификатор плана	PRIMARY KEY	1
title	VARCHAR(200)	Название плана	NOT NULL	'План обучения Python'
description	TEXT	Описание плана	NOT NULL	'Описание плана'
start_date	DATE	Дата начала плана	NOT NULL	'2024-10-15'
end_date	DATE	Дата окончания плана	NOT NULL	'2024-12-15'
status	VARCHAR(20)	Статус плана	NOT NULL	'draft'
created_by	INTEGER	Создатель плана	FOREIGN KEY -> User.id	1
created_at	TIMESTAMP	Дата создания плана	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'
updated_at	TIMESTAMP	Дата обновления плана	DEFAULT CURRENT_TIMESTAMP	'2024-10-15 12:45:00'

Описание полей:

- id: Уникальный идентификатор плана.
- title: Название плана, обязательное для заполнения.
- description: Описание плана.
- start_date: Дата начала плана.
- end_date: Дата окончания плана.
- status: Статус плана (например, 'draft', 'active', 'completed', 'cancelled').

- `created_by`: Ссылка на таблицу `User`, определяющая создателя плана.
- `created_at`: Дата и время создания плана.
- `updated_at`: Дата и время последнего обновления плана.