Частное учреждение

профессиональная образовательная организация

ТЕХНИКУМ «БИЗНЕС И ПРАВО»

ЗАЩИЩЕНО С ОЦЕНКОЙ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и ФИО руководителя)

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_г.

**ОТЧЕТ О ВЫПОЛНЕНИИ ИНДИВИДУАЛЬНОГО ЗАДАНИЯ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

| по ПМ.09 | Проектирование, разработка и оптимизация веб- приложений |
| --- | --- |
| Выполнил | Корниенко М.А. |
| Курс, группа | 4 курс, группа ИС-4 |
| Специальность | 09.02.07 Информационные системы и программирование |
| Руководитель практики | МартыненкоВ.А. |

г. Белореченск

2023 г.

# СОДЕРЖАНИе

[СОДЕРЖАНИЕ](#_Toc129177506)

[ВВЕДЕНИЕ 3](#_Toc129177507)

[1 ПРОЕКТИРОВАНИЕ ВЕБ ПРИЛОЖЕНИЯ 5](#_Оглавление000011)

[1.1 Анализ конкурентов 5](#_Toc129177509)

[1.2 Техническое задание 7](#_Toc129177510)

[1.3 Дизайн веб-проекта 7](#_Toc129177511)

[2 РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ВЕБ ПРИЛОЖЕНИЯ 8](#_Toc129177512)

[2.1 Оптимизация веб-приложения 8](#_Toc129177513)

[2.2 Тестирование веб-приложения 9](#_Toc129177514)

[2.3 Руководство пользователя 15](#_Оглавление000012)

[2.4 Руководство администратора 19](#_Оглавление000013)

[ЗАКЛЮЧЕНЕ 22](#_Toc129177516)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 23](#_Toc129177517)

[ПРИЛОЖЕНИЕ](#_Оглавление000014)

# ВВЕДЕНИЕ

PHP (Hypertext Preprocessor) - это один из наиболее популярных языков программирования для веб-разработки. Сегодня PHP используется на множестве сайтов во всем мире, включая такие известные как Facebook, Wikipedia и Yahoo. Учитывая эту популярность, многие люди хотят научиться программировать на PHP, чтобы создавать собственные веб-приложения и сайты.

При изучении языка программирования PHP может возникнуть множество сложностей, особенно для начинающих программистов. Важно понимать, что программирование - это не только умение писать код, но и понимание логики и принципов работы программ.

В этой исследовательской работе представлена разработка обучающего портала для изучения языка программирования PHP.

Предметом исследования данной работы является разработка обучающего портала для изучения языка программирования PHP.

Объектом исследования является процесс создания данного портала, включая анализ требований, выбор инструментов и технологий для разработки, проектирование и реализацию функциональности портала, а также оценку эффективности созданного продукта.

Целью данного проекта является создание платформы, которая поможет начинающим программистам освоить основы PHP и научиться создавать веб-приложения. Кроме того, обучающий портал будет содержать полезные материалы для более продвинутых программистов, чтобы они могли улучшить свои навыки и расширить свой опыт.

Задачи данной исследовательской работы включают:

* Изучение языка программирования PHP и его особенностей в контексте веб-разработки.
* Анализ требований к обучающему порталу и выбор функциональности, которая будет реализована.
* Изучение инструментов и технологий для разработки веб-приложений, которые будут использоваться при создании обучающего портала.
* Проектирование обучающего портала, включая разработку архитектуры, базы данных, интерфейса пользователя и функциональности.
* Реализация обучающего портала на основе выбранных инструментов и технологий.
* Тестирование и отладка разработанного портала.
* Предложение рекомендаций для дальнейшего улучшения обучающего портала и расширения его функциональности.

В заключении рассмотрим результаты работы, оценим эффективность созданного обучающего портала и предложим рекомендации для его дальнейшего улучшения.

# 1 ПРОЕКТИРОВАНИЕ ВЕБ ПРИЛОЖЕНИЯ

## 1.1 Анализ конкурентов

У данного проекта есть несколько конкурентов, 3 из которых я решил привести в пример.

Список конкурентов с их сильными и слабыми сторонами:

* PHP.net – оффициальный веб-сайт для PHP предоставляющий различную документацию по языку и его функции, как и туториалы, и статьи для разработчиков на PHP. Как официальный источник документации по PHP, сайт имеет хорошую репутацию и доверие среди PHP-сообщества. Но документация на сайте может быть технической и с большим количеством текста, который может не подойти для новичков.

Сильные стороны: как официальный веб-сайт по PHP, PHP.net имеет большое количество информации и ресурсов для разработчиков. Сайт имеет большое доверие, как ресурс для документации по PHP, статьи и туториалы на котором очень полные и детальные. PHP.net так же имеет активный форум сообщества, где разработчики могут задать вопрос и отвечать на чужие.

Слабые стороны: техническая натура документации PHP.net может помешать новичкам понять материал. Дизайн сайта очень устарел, что может создать проблемы в навигации и нахождении нужной информации.

* Codecademy – популярная платформа онлайн-обучения, которая предлагает различные курсы программирования, включая PHP. Codecademy хорошо известна своим интерактивным, практическим подходом к обучению, который включает в себя написание и тестирование кода непосредственно в браузере.Их курс PHP охватывает основные понятия, такие как переменные, типы данных и функции, а также более продвинутые темы, такие как объектно-ориентированное программирование и подключение к базе данных. Курсы Codecademy предназначены для начинающих, поэтому это может быть хорошим вариантом для людей, которые новички в программировании.

Сильные стороны: интерактивный подход Codecademy к обучению хорошо подходит для начинающих, так как позволяет им писать и тестировать код прямо в браузере. Платформа также обеспечивает мгновенную обратную связь, которая может помочь учащимся быстро выявлять и исправлять ошибки. Курс PHP Codecademy охватывает широкий спектр тем, от основ до более продвинутых концепций.

Слабые стороны: курсы Codecademy могут быть несколько поверхностными, поскольку они сосредоточены на предоставлении широкого обзора темы, а не на глубоких знаниях. Кроме того, бесплатные курсы платформы ограничены, и пользователям необходимо платить за доступ к более продвинутым курсам или функциям.

* Udemy - ещё одна популярная платформа онлайн-обучения, которая предлагает широкий спектр курсов, включая несколько курсов PHP. Курсы Udemy создаются индивидуальными инструкторами, поэтому качество и стиль преподавания могут варьироваться. Тем не менее, Udemy предлагает большой выбор курсов PHP, начиная от курсов начального уровня до более продвинутых тем, таких как создание веб-приложений с фреймворками PHP, такими как Laravel или Symfony. Курсы Udemy часто стоят недорого, что делает их доступными для широкого круга учащихся..

Сильные стороны: Udemy предлагает широкий спектр курсов PHP, созданных отдельными инструкторами, что означает, что учащиеся могут выбирать курсы, которые соответствуют их стилю обучения и потребностям. Многие из курсов оцениваются по доступной цене, что делает их доступными для широкого круга учащихся. Кроме того, Udemy предлагает пожизненный доступ к курсам, поэтому учащиеся могут вернуться и просмотреть материал курса в любое время.

Слабые стороны: качество курсов Udemy может варьироваться в широких пределах, так как курсы создаются индивидуальными инструкторами. Некоторые курсы могут быть плохо структурированы или обеспечивать всестороннее освещение темы. Кроме того, курсы Udemy могут быть не такими интерактивными, как другие платформы, что может затруднить учащимся оставаться вовлеченными.

## 1.2 Техническое задание

Техническое задание расположено в приложении А.

## 1.3 Дизайн веб-проекта

Дизайн сайта выполнен в цветах:

* Основной – зелёный;
* Фон – белый

Страницы сайта расположены в приложении Б.

# 2 РАЗРАБОТКА И ДОКУМЕНТИРОВАНИЕ ВЕБ ПРИЛОЖЕНИЯ

## 2.1 Оптимизация веб-приложения

Есть несколько шагов, которые можно предпринять для оптимизации сайта. Вот несколько советов:

* Внутренняя SEO-оптимизация: я убедился, что они оптимизированы для поисковых систем, что каждая страница имеет уникальные описательные заголовки и мета-описания, а заголовки соответствующим образом помечены тегами H1, H2 и H3.
* Создание семантического ядра: провёл исследования ключевых слов для определения наиболее релевантных и объемных поисковых терминов для веб-сайта. Использовал такие инструменты, как Google Keyword Planner или SEMrush, чтобы определить ключевые слова и фразы, которые наиболее актуальны для моего сайта.
* Кластеризация ключевых слов: организовал ключевые слова в кластеры на основе их релевантности и создал контент вокруг этих кластеров. Это поможет создать сплоченный и структурированный сайт, оптимизированный для поисковых систем.
* Поиск и устранение дубликатов: использовал такие инструменты, как Screaming Frog или Google Search Console, чтобы определить дублированный контент на данном сайте. Устраненил дубликаты с помощью перенаправления 301 или канонических тегов для консолидации контента.
* Оптимизация основных тегов: оптимизировал теги заголовка, мета-описания и заголовка на каждой странице, используя соответствующие ключевые слова. Это поможет поисковым системам понять содержание и назначение каждой страницы..
* Поиск и удаление неработающих ссылок: использовал такие инструменты, как Xenu Link Sleuth или Google Search Console, чтобы идентифицировать неработающие ссылки на сайте. Удаление или замена неработающих ссылок для улучшения пользовательского опыта и поисковой оптимизации.
* Очистка и оптимизация кода: оптимизировал код веб-сайта для повышения скорости страницы и пользовательского опыта. Это может включать минимизацию кода, оптимизацию изображений и использование кэширования для сокращения времени загрузки..
* Контент-маркетинг: создал высококачественный, информативный контент, оптимизированный для поисковых систем и соответствующий целевой аудитории. Это поможет привлечь посетителей на сайт и со временем повысить рейтинг в поисковых системах..

## 2.2 Тестирование веб-приложения

Ниже приведены результаты тестирования сайта на различные параметры, такие как скорость загрузки, работоспособность, безопасность и т.д. Данные результаты помогут оценить эффективность оптимизации сайта и выявить возможные проблемы, которые могут быть устранены для улучшения пользовательского опыта.

Таблица 1 – Тестирование сайта

| № Тест-кейса | Название проверки | Результат | Замечания и рекомендации |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 1 | Приложение корректно отображается и функционирует во всех актуальных браузерах (IE, Edge, Chrome, Firefox, Safari,...). | Положительно |  |
| 2 | Приложение корректно отображается и функционирует во всех актуальных версиях браузеров. | Положительно |  |
| 3 | Корректная обработка тайм-аутов | Положительно |  |

Продолжение таблицы 1

| 1 | 2 | 3 | 4 |
| --- | --- | --- | --- |
| 4 | Работоспособность при удалении cookies во время использования приложения. | Положительно |  |
| 5 | Работоспособность при удалении cookies после использования приложения | Положительно |  |
| Работа с формами | | | |
| 6 | Обязательные поля должны быть заполнены до подтверждения ввода данных. | Положительно |  |
| 7 | Обязательные поля должны быть явно обозначены. | Положительно |  |
| 8 | Если пользователь нажал кнопку подтверждения ввода данных с незаполненными обязательными полями, эти поля должны быть подсвечены и отображено сообщение об ошибке. | Положительно |  |
| 9 | Сообщение об ошибке не отображается, если не заполнены необязательные поля | Положительно |  |
| 10 | Текстовые поля допускают ввод специальных символов | Положительно |  |
| 11 | При вводе количества знаков меньше минимального отображается сообщение об ошибке | Положительно |  |
| 12 | При вводе количества знаков больше максимального отображается сообщение об ошибке. | Положительно |  |
| 13 | Если обязательное поле заполнено не полностью, отображается сообщение об ошибке | Положительно |  |

Продолжение таблицы 1

| 1 | 2 | 3 | 4 |
| --- | --- | --- | --- |
| 14 | Сообщение об ошибке отображается при попытке ввести недопустимые символы (например, ввод текста в поле, принимающее только числа). | Положительно |  |
| Авторизация и работа с профилем | | | |
| 15 | Создайте новый аккаунт. | Положительно |  |
| 16 | Авторизуйтесь с существующим аккаунтом | Положительно |  |
| 17 | Восстановите/сбросьте забытый пароль. | Положительно |  |
| 18 | При вводе неверного пароля отображается сообщение об ошибке и ссылка на страницу восстановления пароля. | Положительно |  |
| 19 | Отредактируйте профиль пользователя | Положительно |  |
| 20 | Зарегистрируйте нового пользователя, используя аккаунт социальных сетей | Функционал не реализован |  |
| 21 | Авторизуйтесь, используя аккаунт социальных сетей | Функционал не реализован |  |
| 22 | Загрузите в профиль аватар/картинку | Функционал не реализован |  |
| 23 | Отредактируйте аватар/картинку. | Функционал не реализован |  |
| 24 | Удалите аватар/картинку. | Функционал не реализован |  |
| 25 | При регистрации нового пользователя ему приходит подтверждение по электронной почте. | Функционал не реализован |  |
| 26 | При оформлении подписки на уведомления пользователю приходит подтверждение по электронной почте. | Функционал не реализован |  |

Продолжение таблицы 1

| 1 | | 2 | | 3 | | 4 |
| --- | --- | --- | --- | --- | --- | --- |
| 27 | | При смене пароля пользователю приходит подтверждение по электронной почте | | Функционал не реализован | |  |
| Навигация | | | | | | |
| 28 | При выполнении действий (например, выход из аккаунта) и манипуляциях с 28 данными (например, удаление файла) отображается сообщение с подтверждением. | | Функционал не реализован | |  | |
| 29 | Корректность работы сортировки. | | Функционал не реализован | |  | |
| 30 | Корректность работы фильтрации | | Функционал не реализован | |  | |
| 31 | Корректность работы кнопок и ссылок в меню навигации. | | Положительно | |  | |
| 32 | Отображение страницы с ошибкой 404, если пользователь указал неверный путь к файлу или URL. | | Функционал не реализован | |  | |
| 33 | Осуществляется проверка на корректность дат | | Функционал не реализован | |  | |
| 34 | Корректная обработка високосного года, отсутствует сообщение об ошибке | | Функционал не реализован | |  | |
| 35 | Числовые поля не допускают ввода текста, отображается сообщение об ошибке. | | Функционал не реализован | |  | |
| 36 | При попытке ввода отрицательных чисел, отображается сообщение об ошибке, если предусмотрен ввод строго положительных чисел. | | Функционал не реализован | |  | |
| 37 | При попытке деления на ноль отображается сообщение об ошибке. | | Функционал не реализован | |  | |

Продолжение таблицы 1

| 1 | 2 | 3 | 4 |
| --- | --- | --- | --- |
| 38 | Поля специальных форматов (например, процентные, денежные) отображаются корректно. | Функционал не реализован |  |
| 39 | Всплывающие уведомления об ошибке и предупреждения работают корректно. | Положительно |  |
| Интеграция с социальными сетями | | | |
| 40 | Корректное отображение кнопок социальных сетей. | Функционал не реализован |  |
| 41 | При нажатии на кнопку "поделиться отображается всплывающее окно, в котором доступна функция редактирования сообщения перед отправкой. | Функционал не реализован |  |
| 42 | Корректное отображение окна авторизации с помощью аккаунта социальных сетей. | Функционал не реализован |  |
| 43 | Корректное отображение аватара пользователя из профиля социальных сетей. | Функционал не реализован |  |
| 44 | Пользователь может оставлять комментарии на вашем сайте, используя аккаунт социальных сетей. | Функционал не реализован |  |
| Загрузка и скачивание файлов | | | |
| 45 | Файлы успешно загружаются | Функционал не реализован |  |
| 46 | Слишком тяжелый файлы не загружаются, и отображаются сообщения об ошибке | Функционал не реализован |  |
| 48 | Файлы ошибочного формата не загружаются, и отображается сообщение об ошибке. | Функционал не реализован |  |
| 49 | Файлы недопустимого формата не загружаются, и отображается сообщение об ошибке. | Функционал не реализован |  |
| 50 | Файлы успешно скачиваются. | Функционал не реализован |  |

## 

## 2.3 Руководство пользователя

Чтобы пользователю начать пользоваться сайтом, ему нужно:

1. Если пользователь не имеет аккаунта на сайте:
   1. придумать логин и пароль;
   2. зайти на сайт;
   3. перейти на вкладку войти, изображённую на рисунке 1;



Рисунок 1 – Руководство пользователя - кнопка «войти».

* 1. Перейти по ссылке, указанной на рисунке 2;

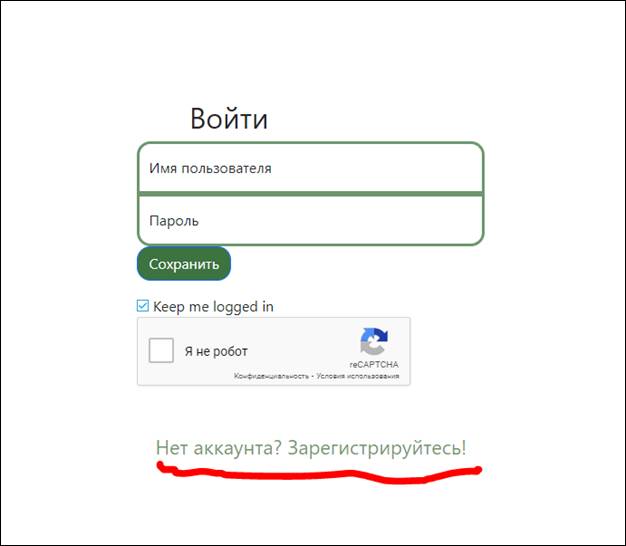


Рисунок 2 – Руководство пользователя – переход на страницу регистрации.

* 1. Ввести придуманные логин и пароль в форму, заполнив все поля, включая проверку на робота, и записать их где-нибудь. Форма изображена на рисунке 3;

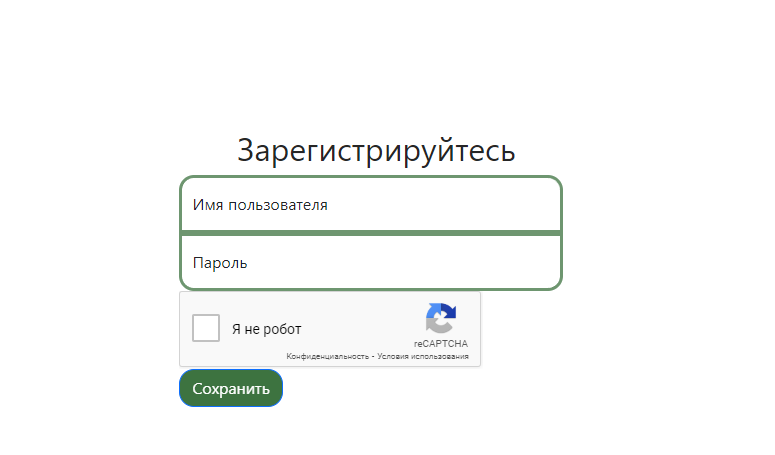


Рисунок 3 – Руководство пользователя – форма регистрации.

* 1. После ввода логина и пароля, подтверждения проверки на робота, и отправки формы, пользователя перекинет на главную страницу.

1. Если пользователь уже имеет аккаунт на сайте:
   1. зайти на сайт;
   2. перейти на вкладку войти, изображённую на рисунке 1;
   3. ввести в поля ваши данные, придуманные ранее, пройти тест на робота и, по желанию, отметить галочку «Оставаться в сети». Эта галочка позволит вам не вводить данные снова, даже спустя время. На данном сайте вы будете в сети до тех пор, пока не пройдёт неделя, или вы не выйдете из аккаунта. Форма авторизации показана на рисунке 4;

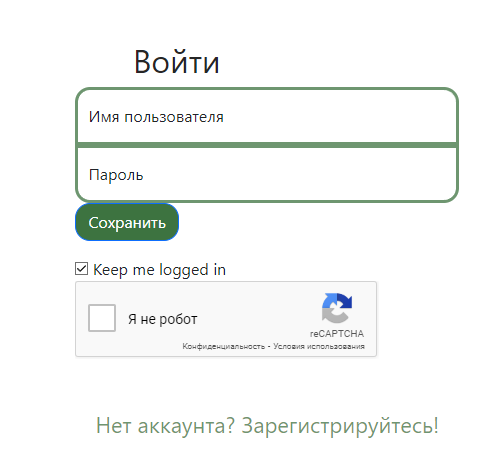


Рисунок 4 – Руководство пользователя – форма авторизации.

* 1. Если данные верны, вас перекинет на главную страницу, иначе, выведет ошибку.

После авторизации, пользователю становится доступны:

* личный кабинет;
* тесты.

Чтобы прочитать лекцию, надо:

1. На главной странице выбрать в меню пункт «Лекции». Пункт изображён на рисунке 5;

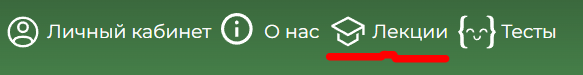


Рисунок 5 – Руководство пользователя – пункт «Лекции».

1. По переходу на страницу, появляется список лекций. Теперь вы можете выбрать любую из них и изучить материал внутри. Просто нажмите на ссылку. Ссылка показана на рисунке 6.

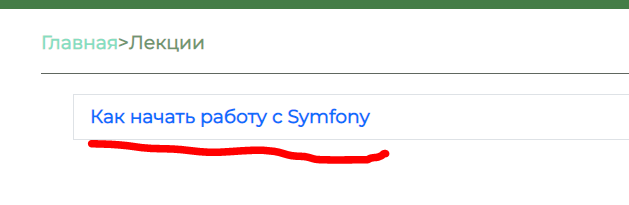


Рисунок 6 – Руководство пользователя – ссылка на лекцию.

1. После перехода по ссылке откроется страница с лекцией, которую можно изучить вне зависимости от того, есть у пользователя аккаунт или нет.

Чтобы воспользоваться тестами, нужно:

1. Переходим по вкладке в меню «Тесты», изображённой на рисунке 7;



Рисунок 7 – Руководство пользователя – пункт «Тесты».

1. На странице будет отображён список тестов, которые вы можете пройти. Список тестов отображён на рисунке 8

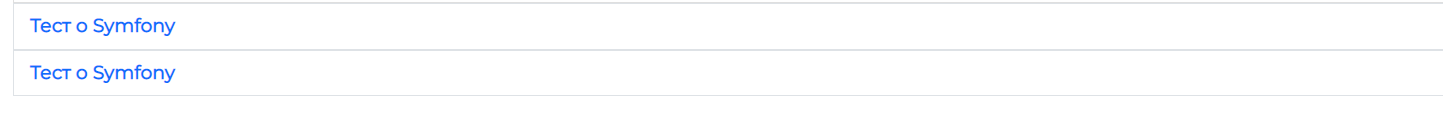


Рисунок 8 - Руководство пользователя - список Тестов.

1. При нажатии на любой из тестов у вас отоброизтся тест с простой формой где есть вопрос, а под ним ответы. С помощью галочки вы можете пометить ответ как правильный;

## 2.4 Руководство администратора

Помимо пользователей есть две роли:

* редакторы;
* администраторы.

Редакторы могут:

* создавать лекции и тесты;
* редактировать лекции;
* удалять лекции и тесты.

Чтобы получить доступ ко всем средства нужно войти в систему.

На рисунке 9 показан список лекций от лица редактора.

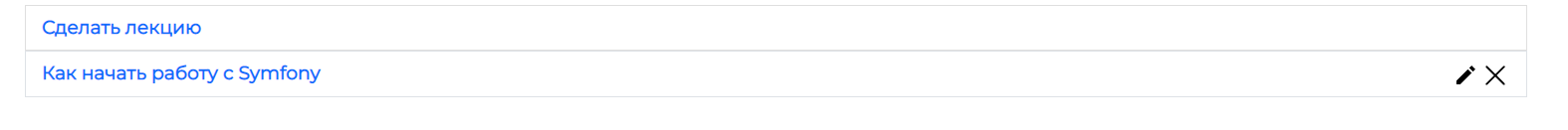


Рисунок 9 - Руководство администратора - создание, редактирование и удаление лекции.

* ссылка «создать лекцию» - создаёт лекцию;
* иконка карандаша - редактирует лекцию;
* иконка крестика - удаляет лекцию.

На рисунке 10 изображена форма создания и редактирования лекции. Форма состоит из двух обязательных полей:

* названия;
* наполнения.

После заполнения этих полей любым количество информации нажмите на кнопку «сохранить» чтобы сохранить или отредактировать лекцию в базу данных (далее БД).

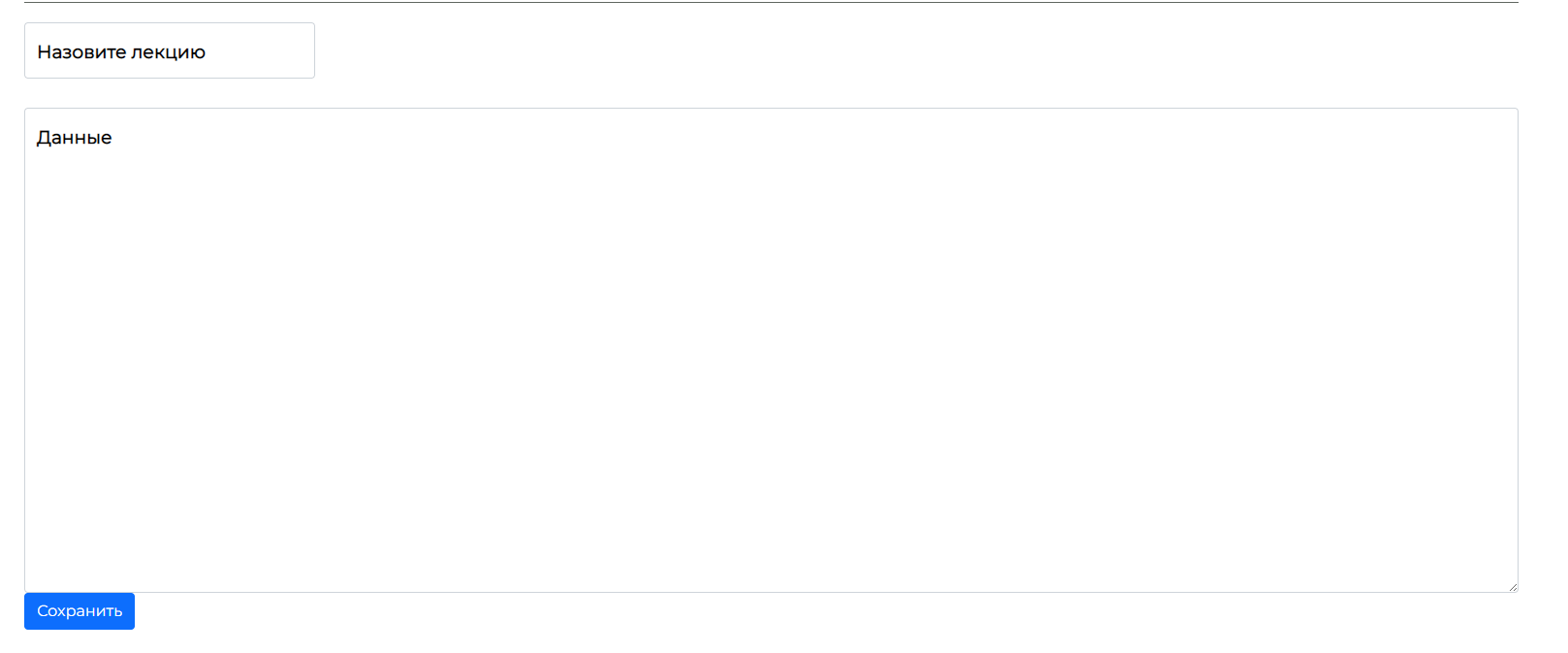


Рисунок 10 - Руководство администратора - Форма создания/редактирования лекции.

На рисунке 11 изображен список тестов от лица редактора. На нём:

* ссылка «сделать тест» - создать новый тест;
* иконка крестика - удалить тест.

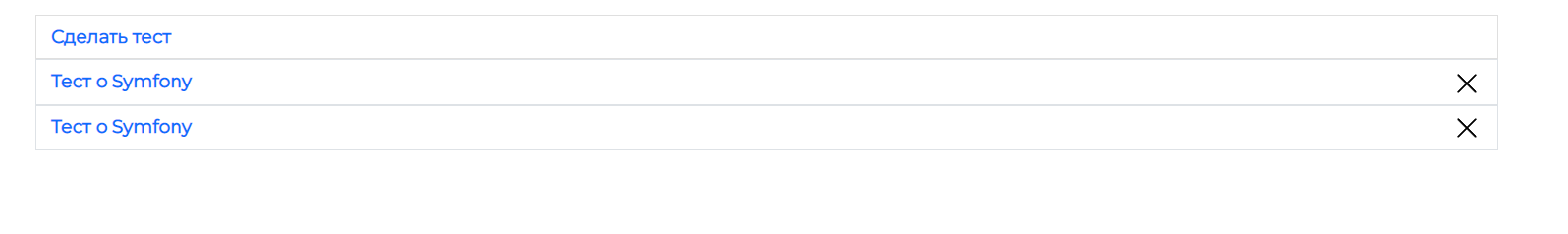


Рисунок 11 - Руководство администратора - создание и удаление тестов.

На рисунке 12 показана форма создания теста. На ней:

* кнопка «добавить вопрос» - добавить вопрос в форму;
* кнопка «добавить ответ» - добавить ответ к вопросу;

Поля «Название теста», «Вопрос», «Ответ» обязательны к заполнению любым количеством информации. пункт «Правильный» может быть у нуля или более вопросов.

После нажатия кнопки сохранить тест будет добавлен в БД.

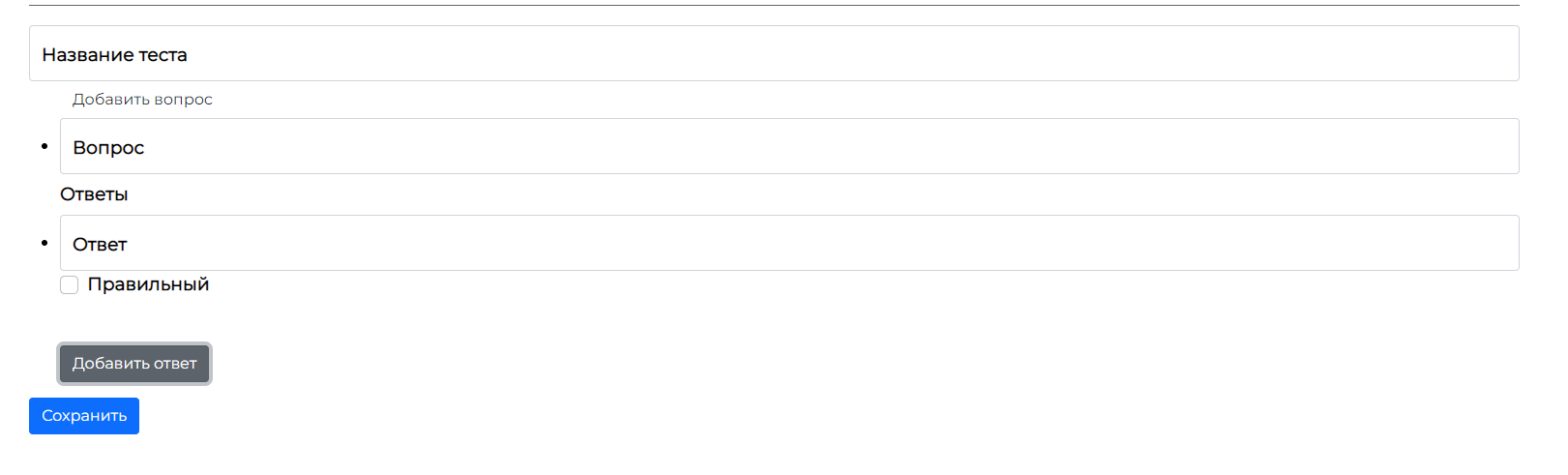


Рисунок 12 - Руководство администратора - создание теста.

# ЗАКЛЮЧЕНЕ

В заключение, разработка обучающего портала "язык программирования PHP" является важным и актуальным проектом, который поможет людям изучать язык программирования PHP и повышать свои навыки в этой области.

При разработке портала были учтены требования и потребности современных пользователей, что позволило создать удобный и интуитивно понятный интерфейс, а также обеспечить разнообразные методы обучения и проверки знаний.

В процессе создания портала были использованы современные технологии и методы разработки, что обеспечило высокую функциональность и эффективность работы.

Исходя из результатов тестирования портала, можно сделать вывод, что он полностью соответствует заданным требованиям и является эффективным инструментом для изучения языка программирования PHP.

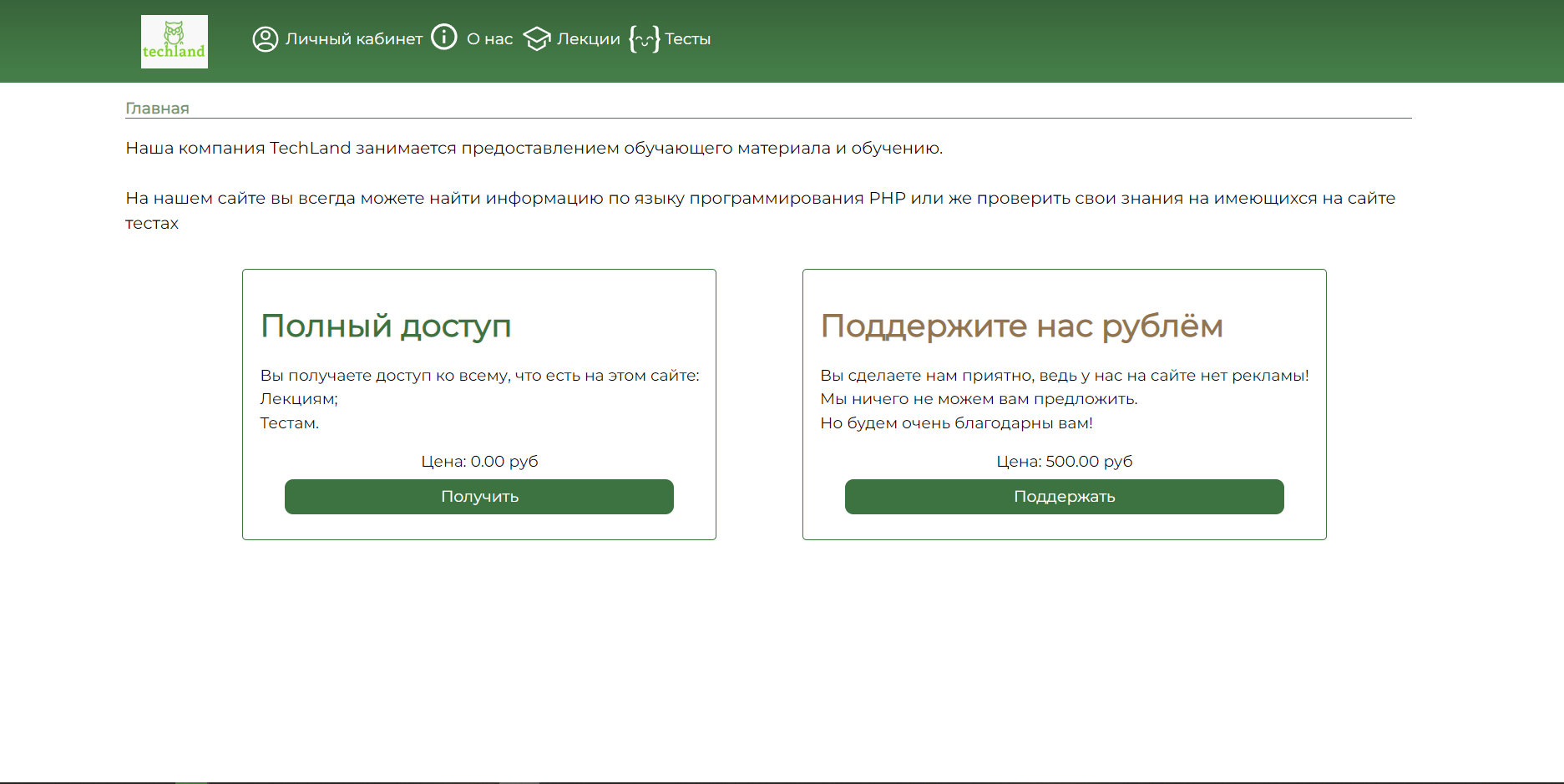
Кроме того, портал разработан с учетом потребностей разных категорий пользователей. Для начинающих пользователей предусмотрены простые и понятные материалы, которые позволяют освоить основы языка программирования PHP. Для опытных пользователей предусмотрены более сложные темы и задания, которые помогают углубить знания и навыки в области программирования.

В целом, разработка обучающего портала "язык программирования PHP" является успешным проектом, который может быть полезен как начинающим, так и опытным программистам в их профессиональном развитии.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. PHP: Hypertext Preprocessor. Официальный сайт языка PHP.
2. W3Schools. PHP 7 Tutorial.
3. TutorialsPoint. PHP Tutorial.
4. SitePoint. PHP Tutorial for Beginners.
5. Codecademy. Learn PHP.
6. Udemy. PHP for Beginners - Become a PHP Master.
7. Lynda.com. PHP Essential Training.
8. Packt Publishing. Mastering PHP 7.
9. Zend Technologies. Zend PHP Certification Study Guide.
10. O'Reilly Media. Learning PHP, MySQL & JavaScript.
11. PHP.net. PHP Manual.
12. PHP: The Right Way. Best Practices for Modern PHP Development.
13. Laracasts. PHP Tutorials.
14. PHP Academy. PHP Tutorials.
15. PHP & MySQL: Novice to Ninja. A step by step guide to building dynamic websites.
16. A Beginner's Guide to PHP Development. A comprehensive tutorial on PHP development.
17. PHP: Learn PHP Programming Quick & Easy. A beginner's guide to PHP programming.
18. PHP Programming: Master's Handbook. A comprehensive guide to mastering PHP programming.
19. The PHP Practitioner. A video course on PHP development.
20. PHP 7 Programming Cookbook. Over 80 recipes for PHP 7 programming.

| ПРИЛОЖЕНИЕ |
| --- |



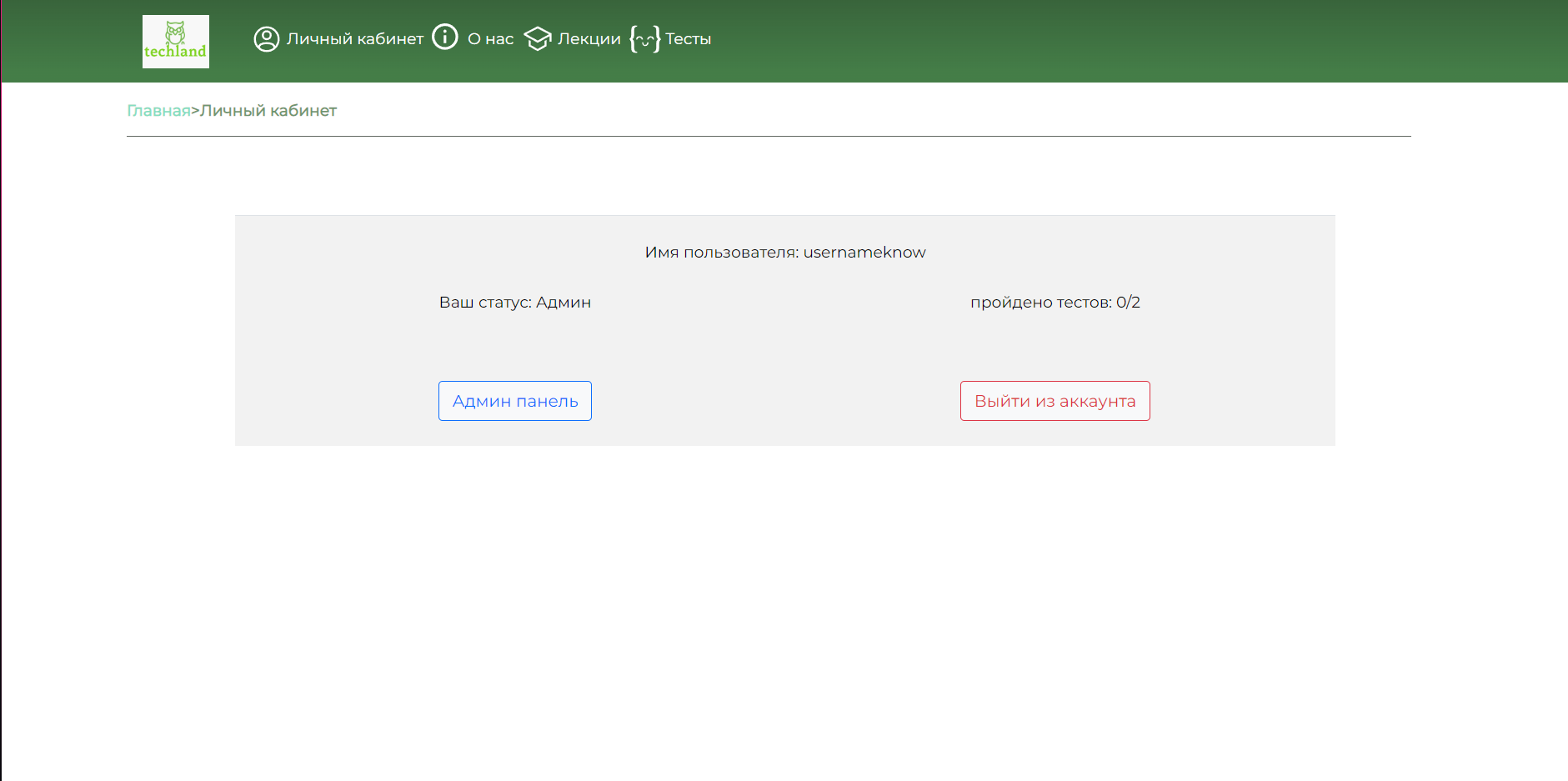
Рисунок 13 - Главная страница.

Рисунок 14 - Личный кабинет пользователя.

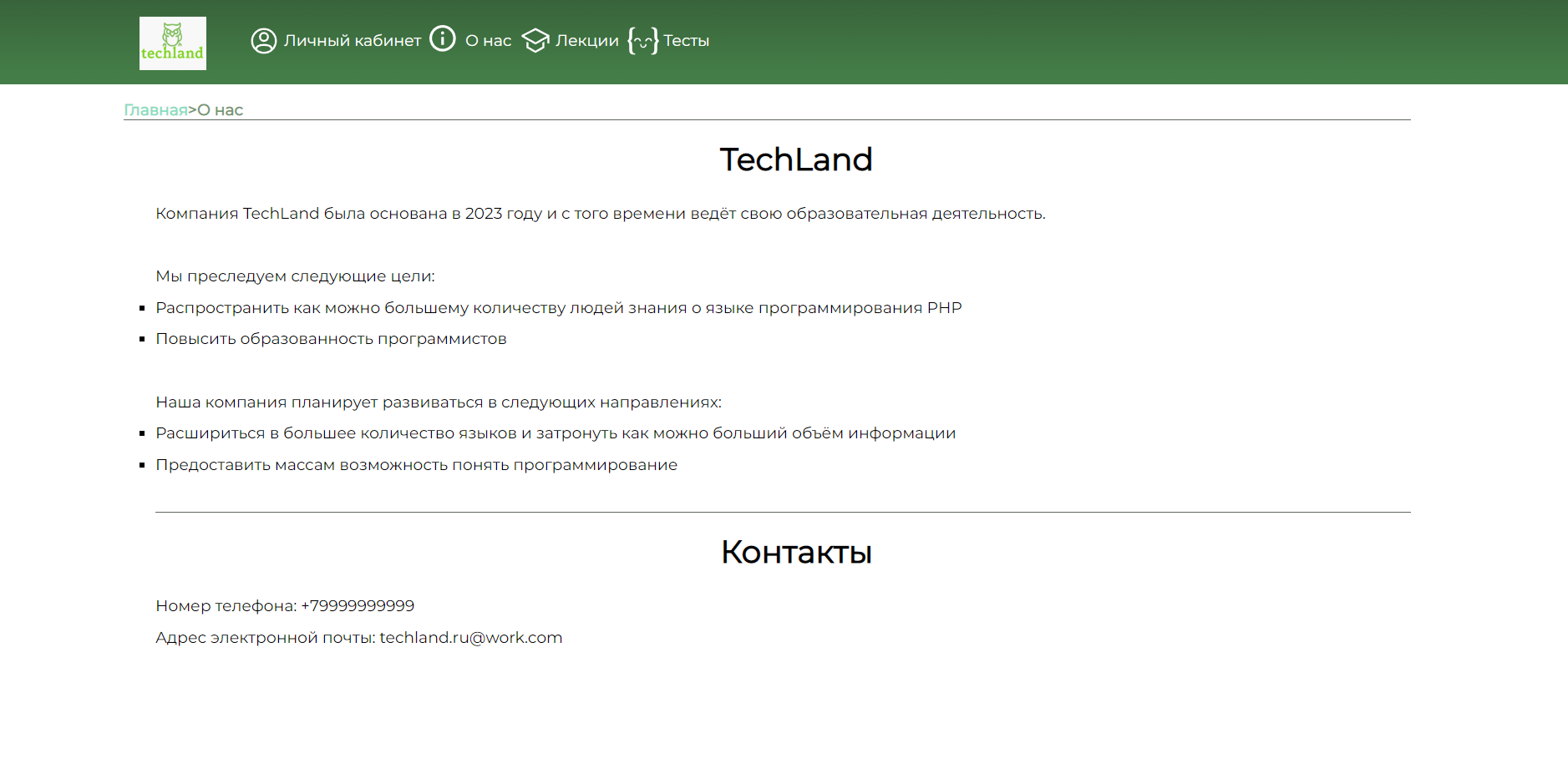


Рисунок 15 - О нас.

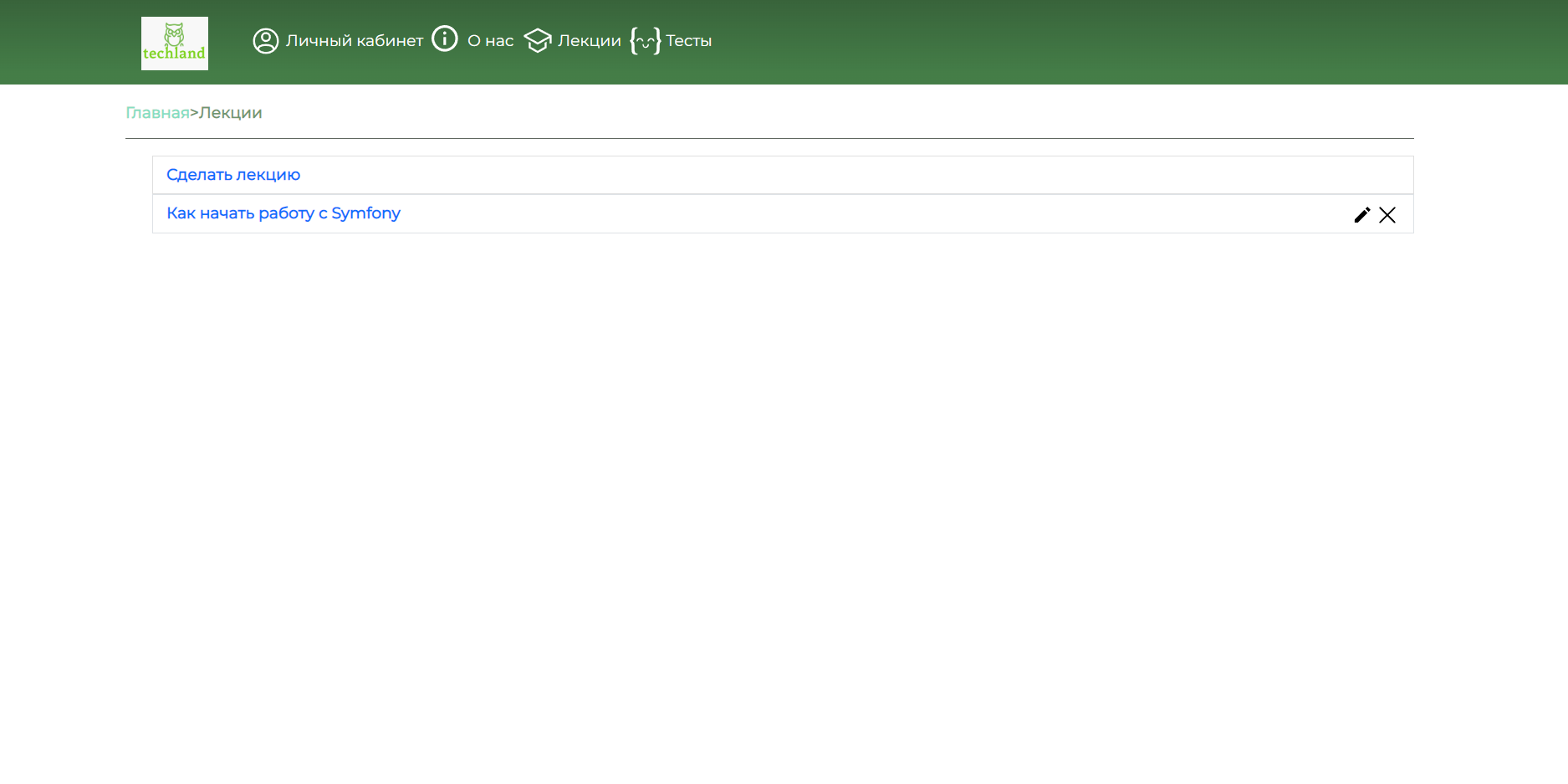


Рисунок 16 - Список лекций у редактора\админа.

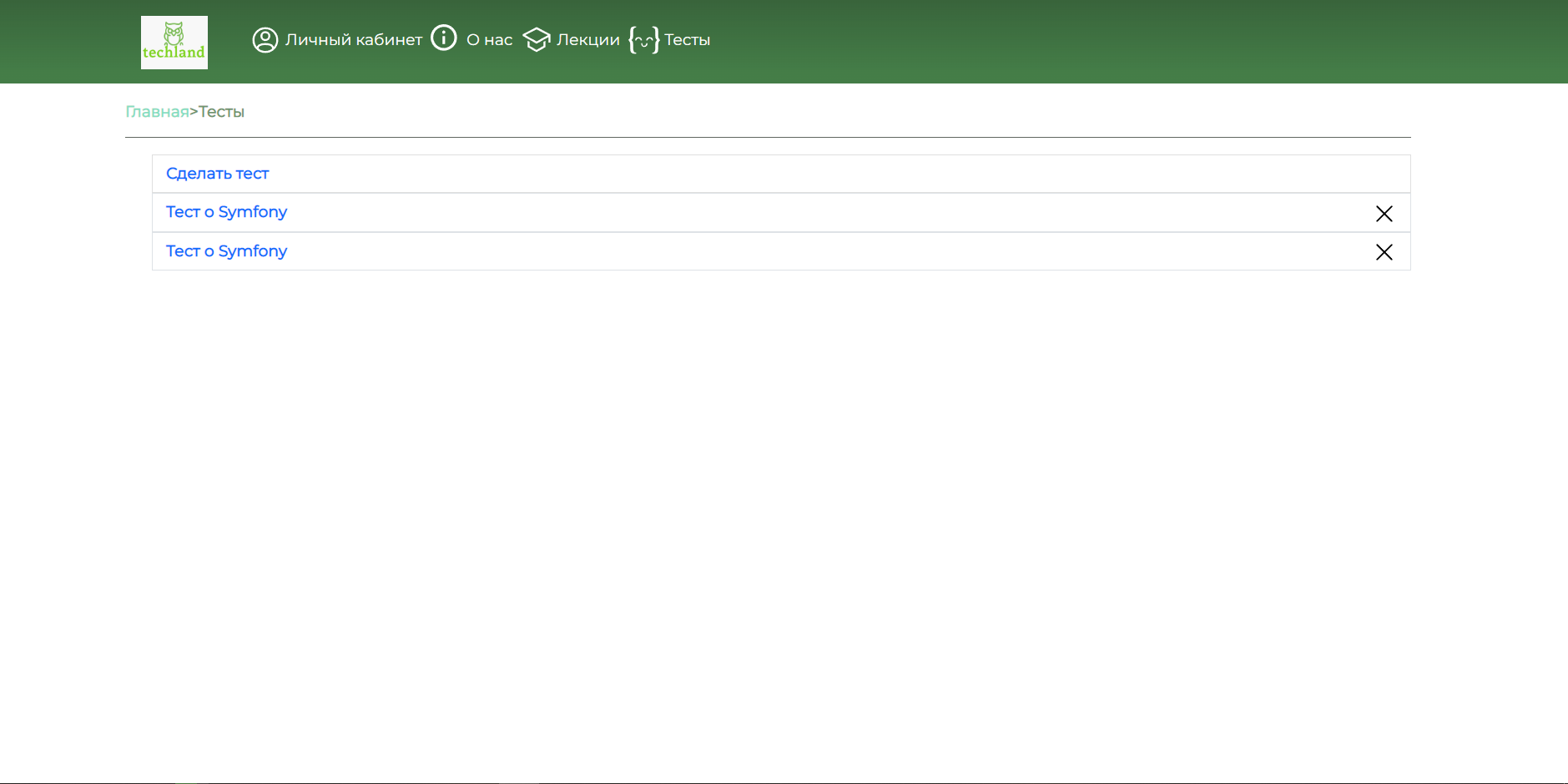


Рисунок 17 - Список тестов у редактора\админа.

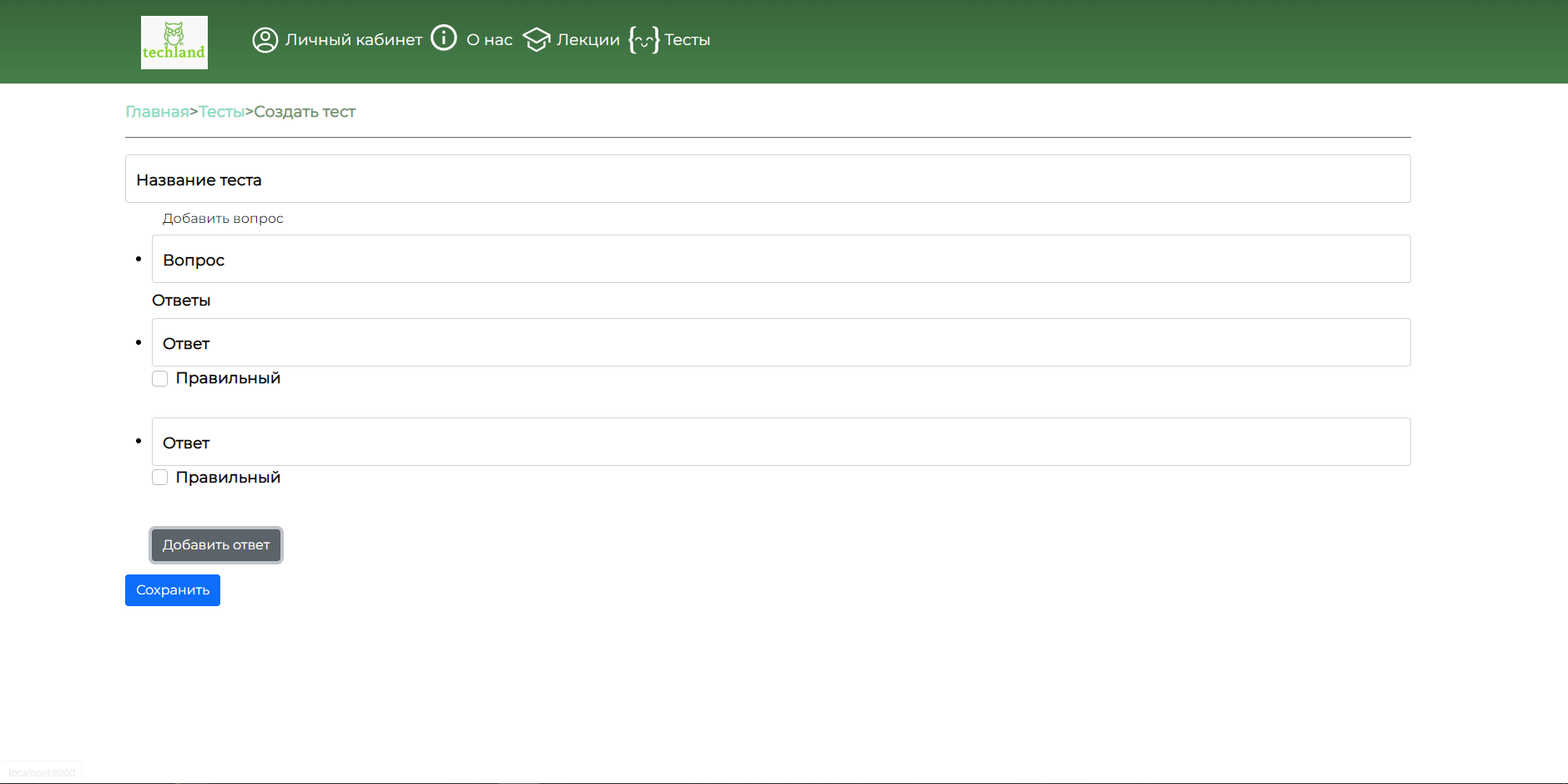


Рисунок 18 - Создание теста.

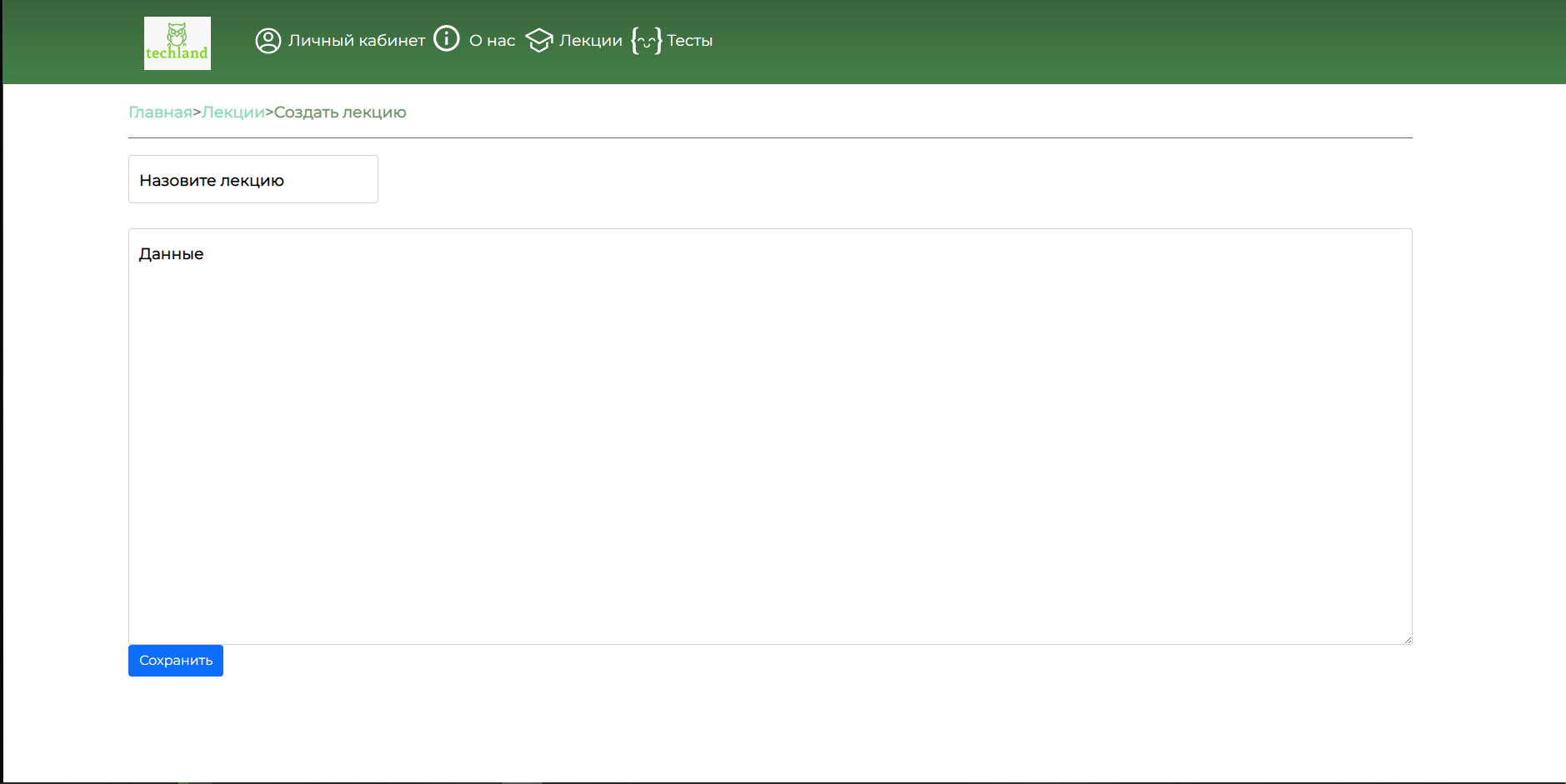


Рисунок 19 - Создание\редактирование лекции.

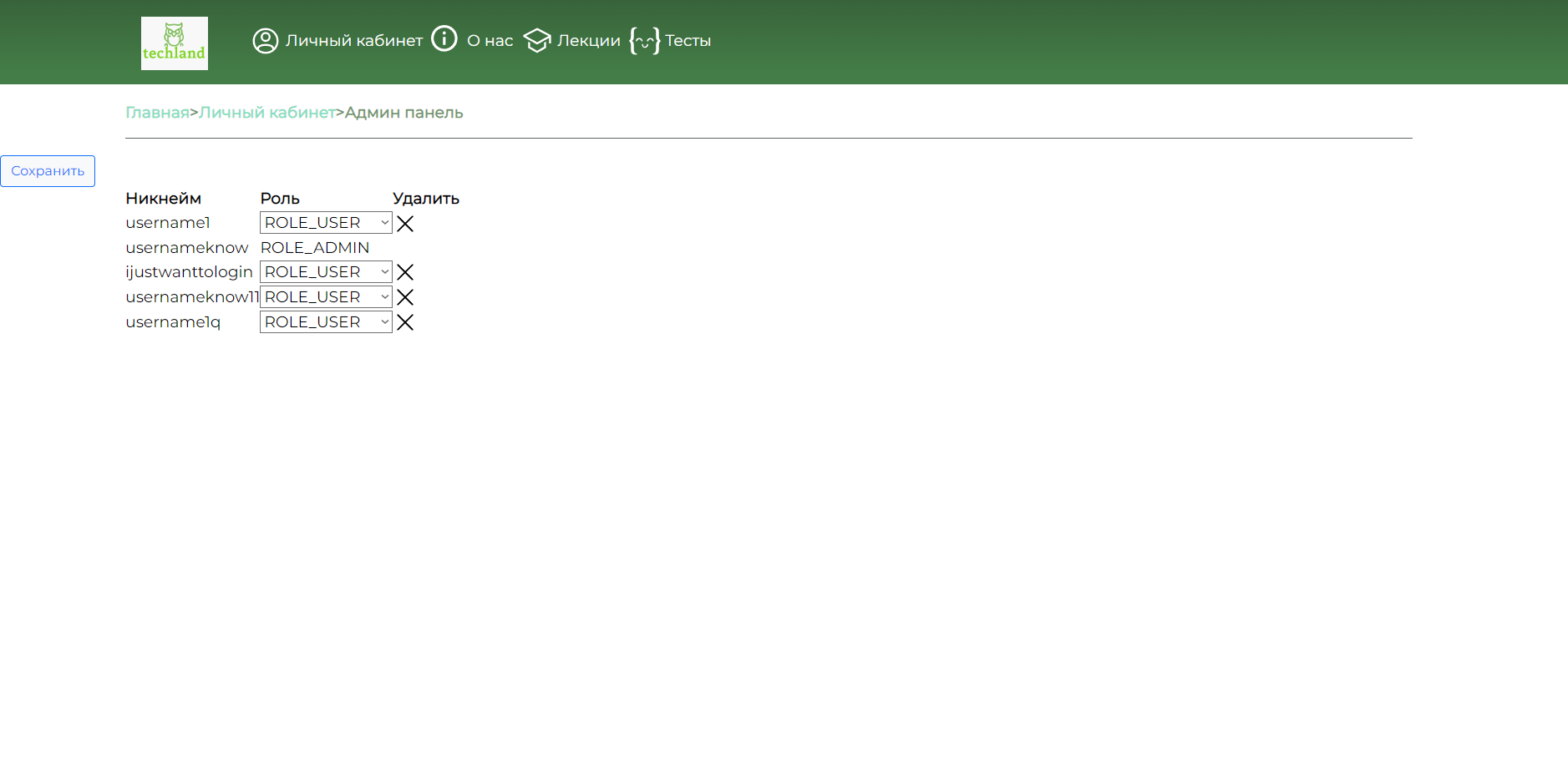


Рисунок 20 - Админ панель.

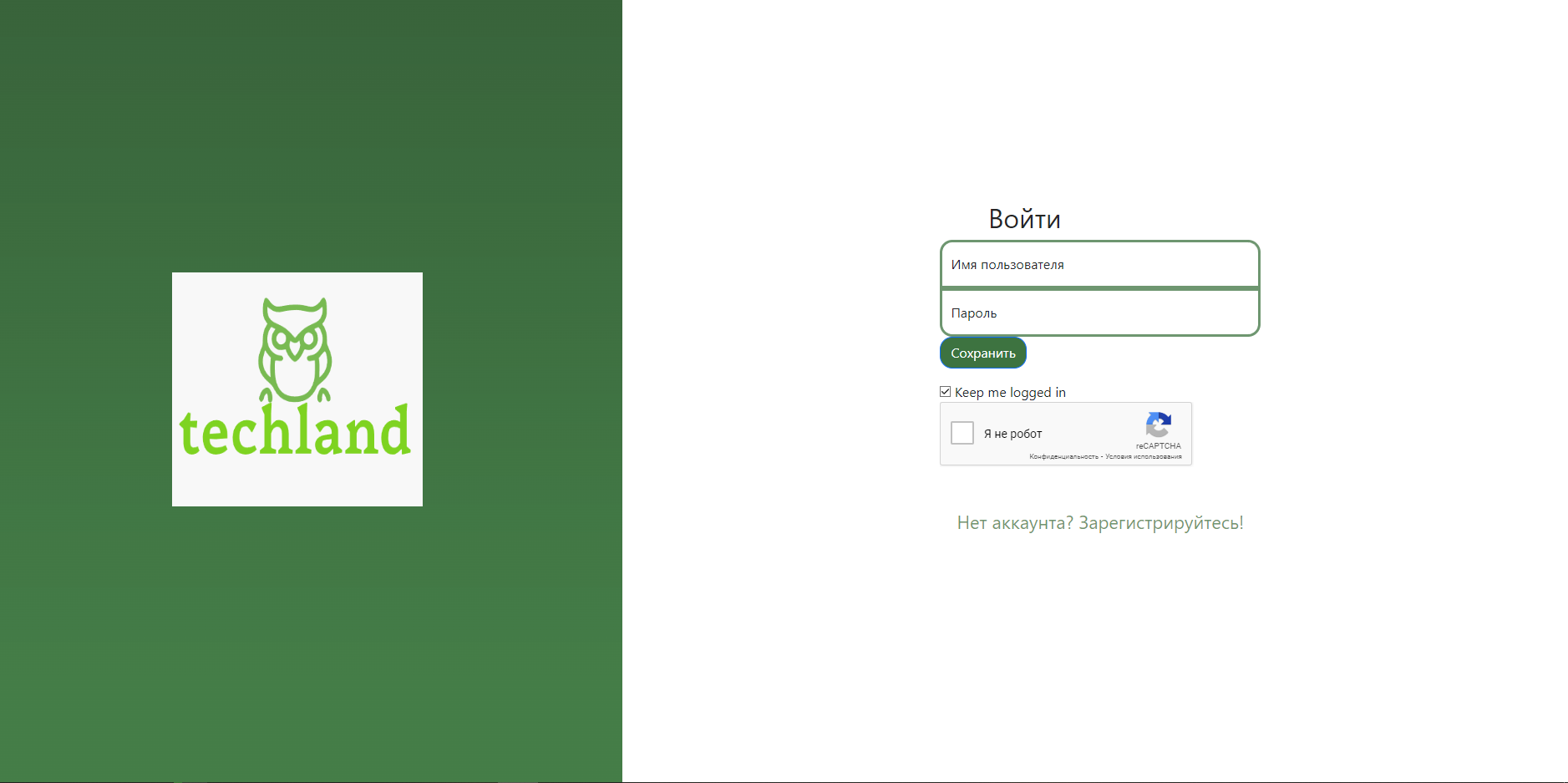


Рисунок 21 - Окно входа.

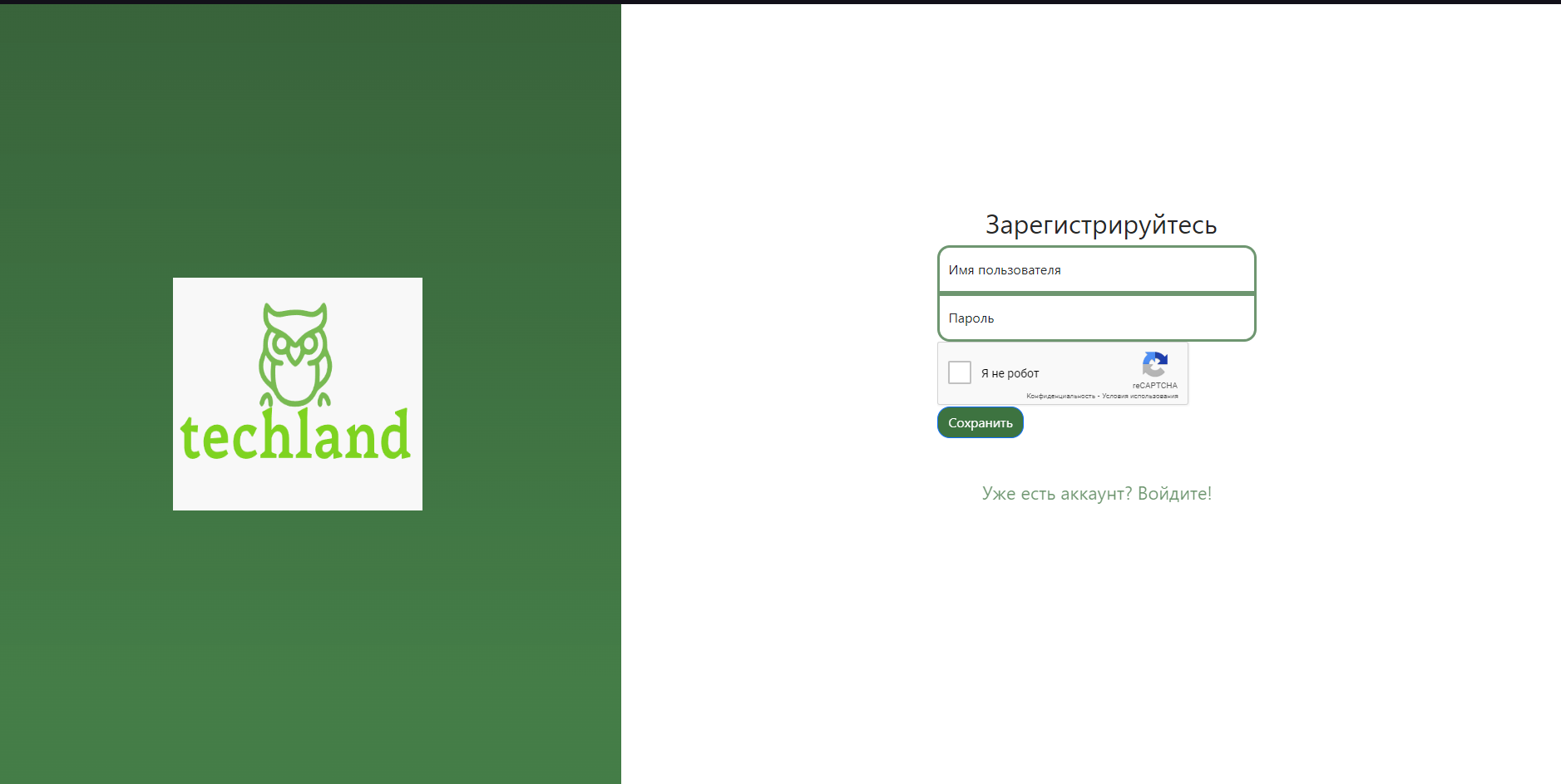


Рисунок 22 - Окно регистрации.

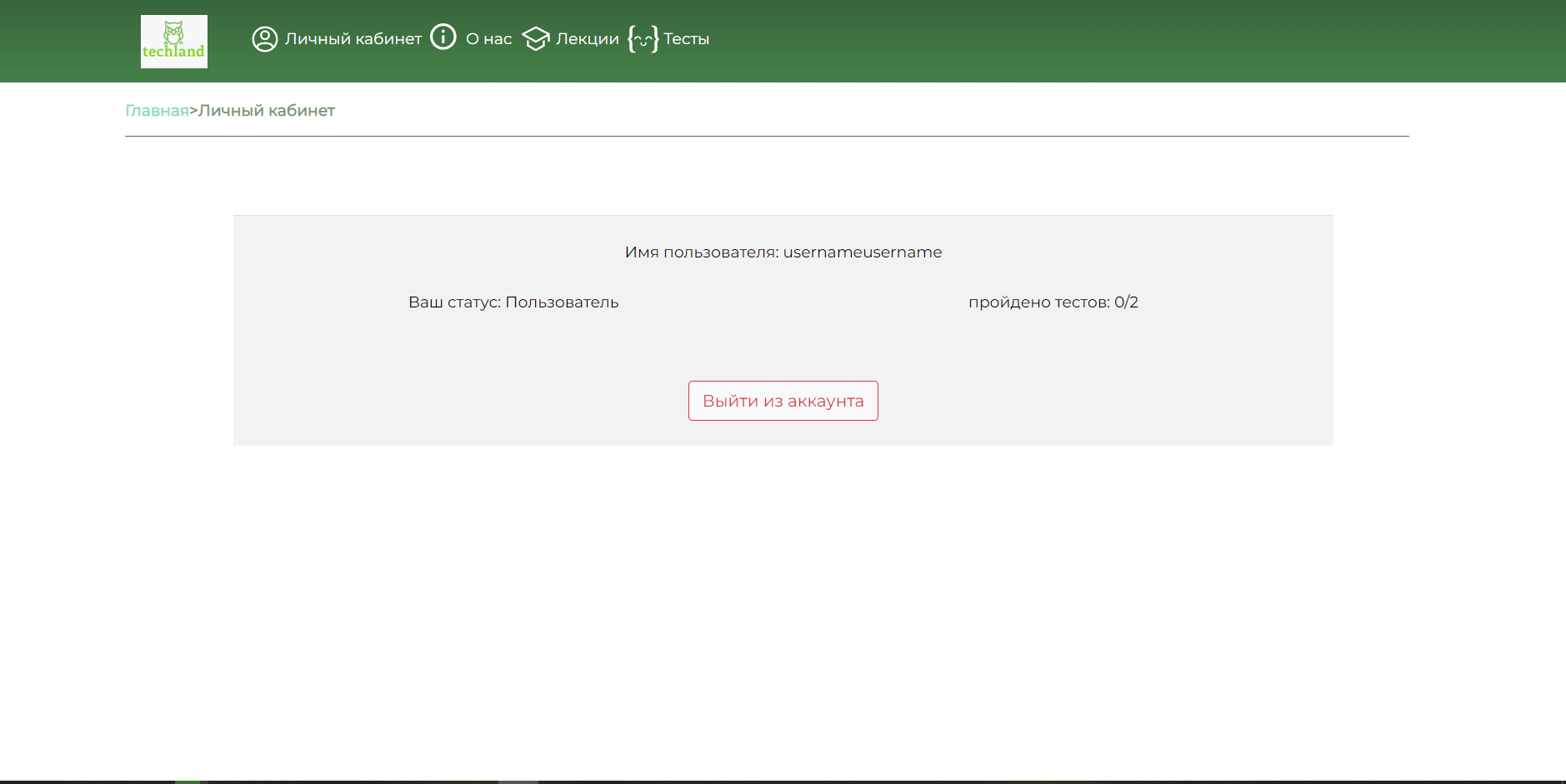


Рисунок 23 - Личный кабинет пользователя.

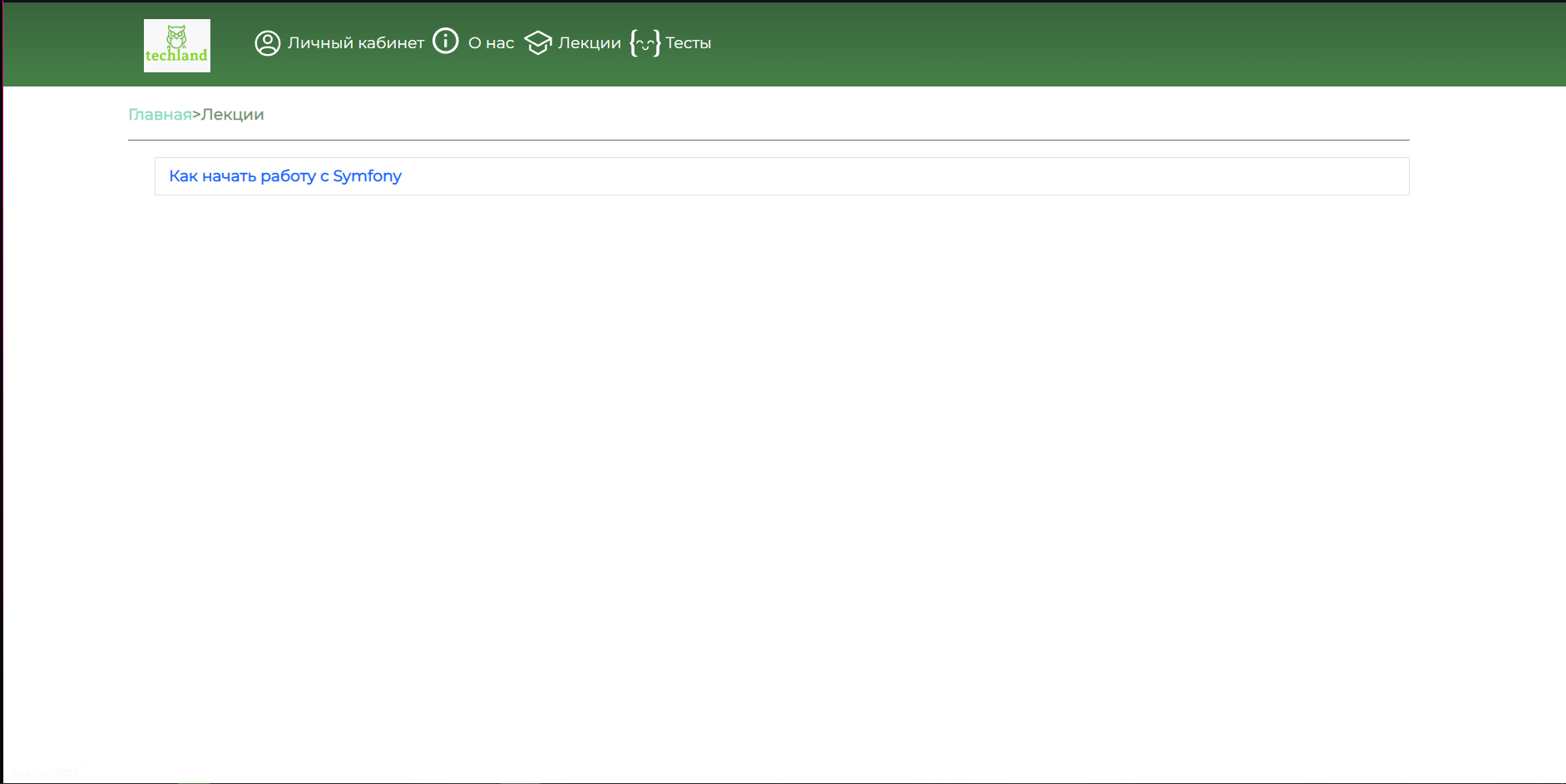


Рисунок 24 - Список лекций у пользователя.

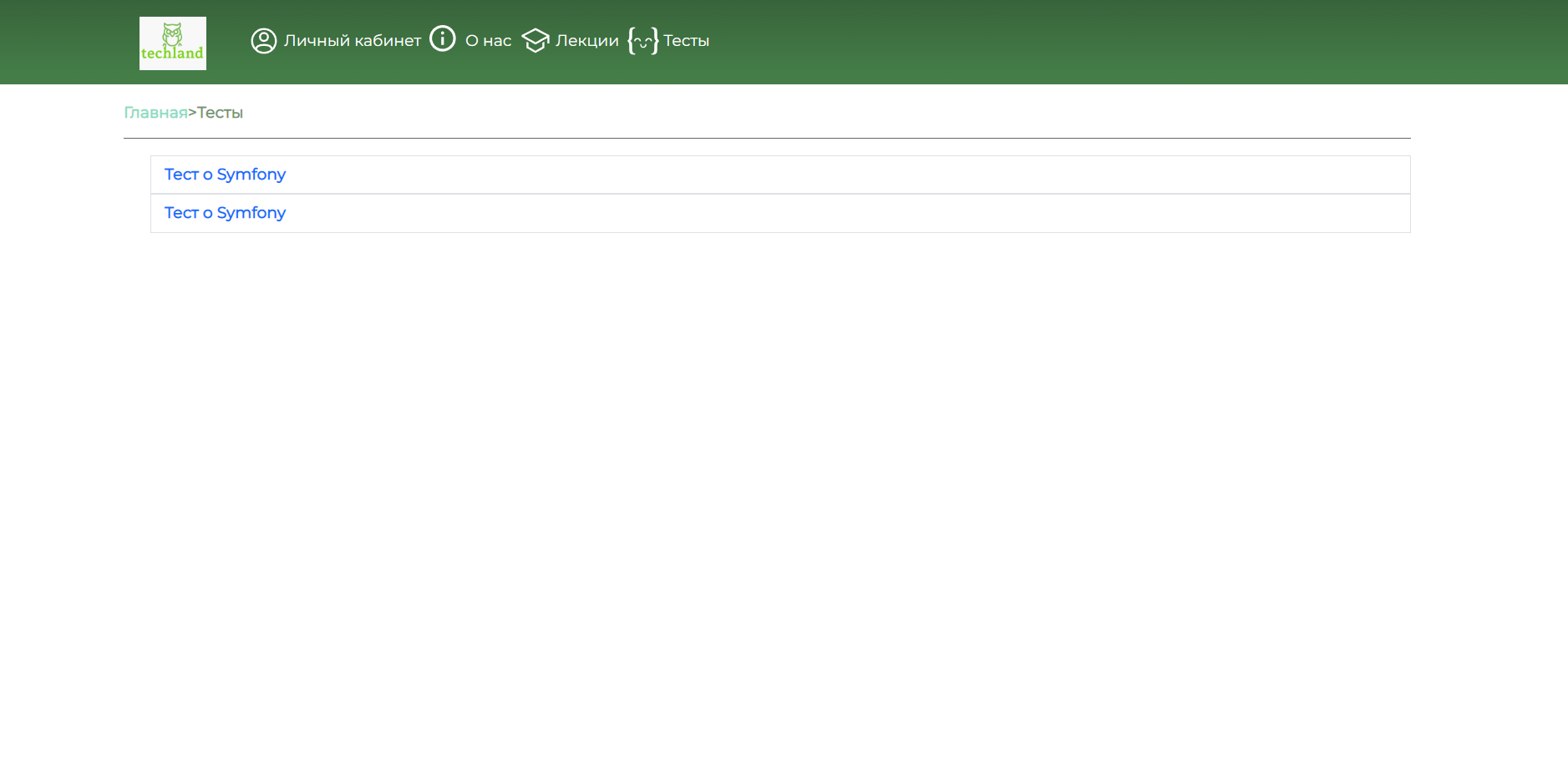


Рисунок 25 - Список тестов у пользователя.

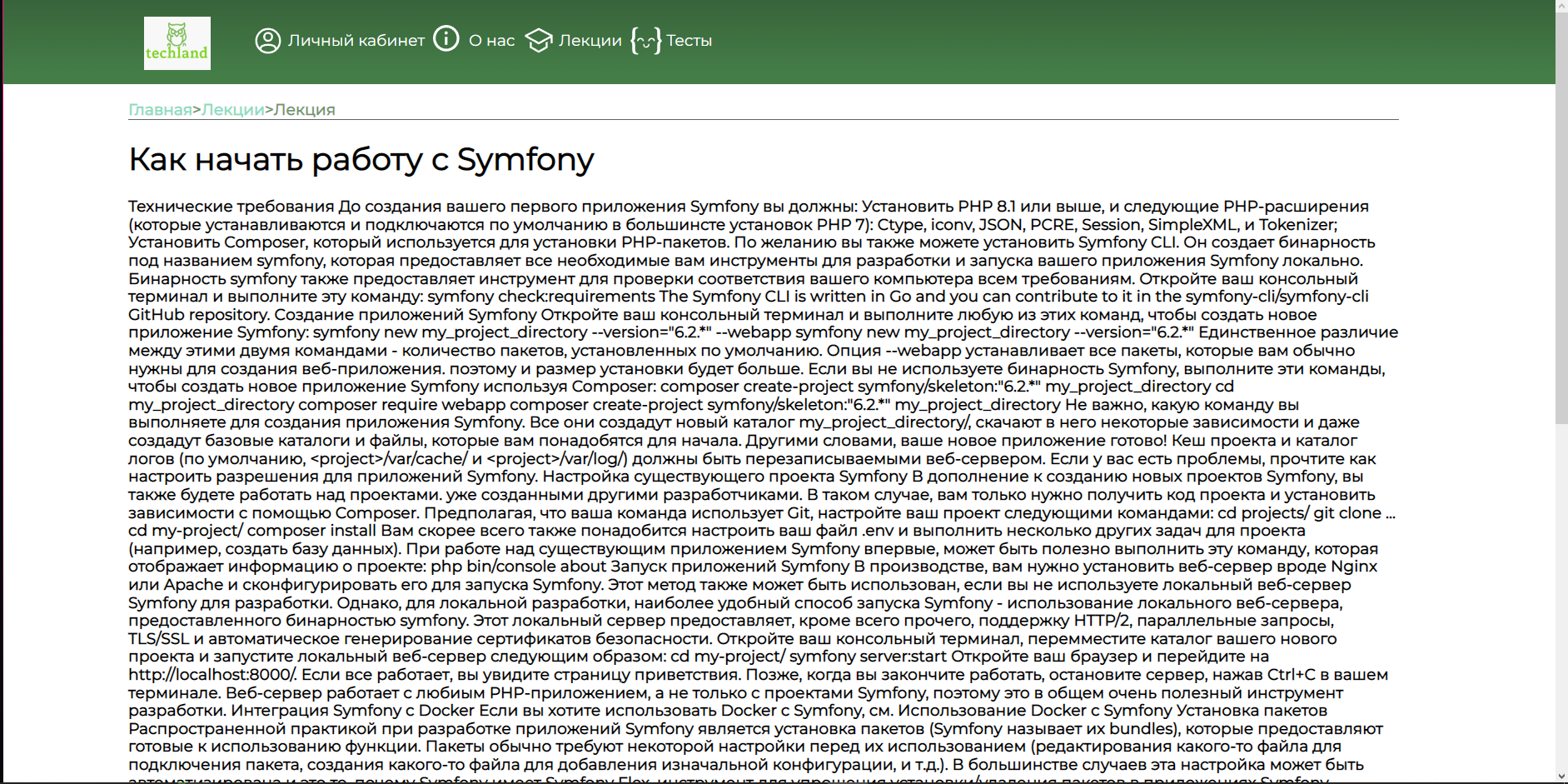


Рисунок 26 - Лекция.

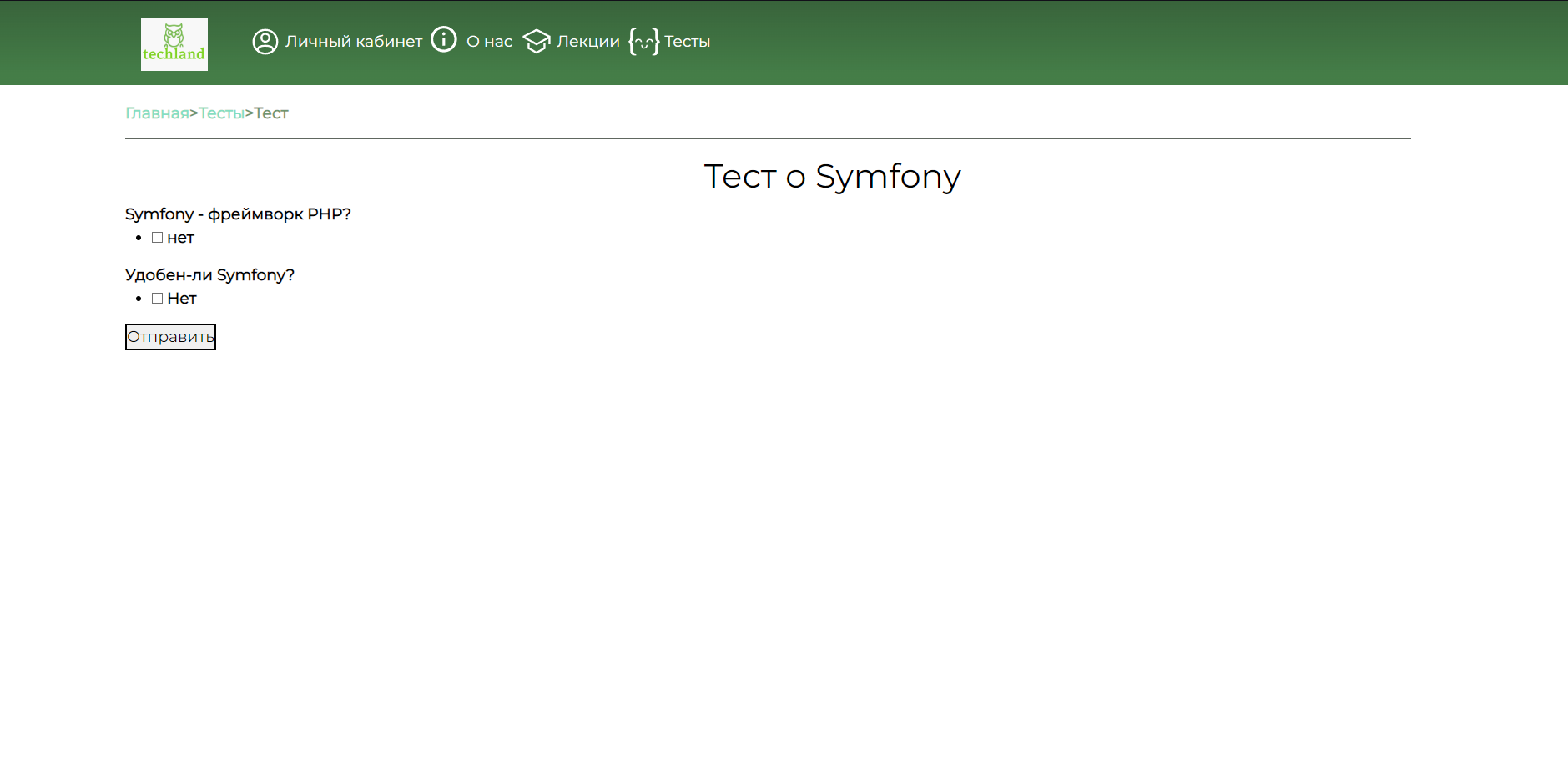


Рисунок 27 - Тест.

bin/console.php

#!/usr/bin/env php<?php use app\kernel;use symfony\bundle\frameworkbundle\console\application;if (!is\_file(dirname(*\_\_dir\_\_*) . '/vendor/autoload\_runtime.php')) {throw new logicexception('symfony runtime is missing. try running "composer require symfony/runtime".');}require\_once dirname(*\_\_dir\_\_*) . '/vendor/autoload\_runtime.php';return function (array $context) {$kernel = new kernel($context['app\_env'], (bool)$context['app\_debug']);return new application($kernel);};

bin/phpunit.php

#!/usr/bin/env php <?php if (!ini\_get('date.timezone')) { ini\_set('date.timezone', 'UTC'); } if (is\_file(dirname(*\_\_DIR\_\_*).'/vendor/phpunit/phpunit/phpunit')) { define('PHPUNIT\_COMPOSER\_INSTALL', dirname(*\_\_DIR\_\_*).'/vendor/autoload.php'); require *PHPUNIT\_COMPOSER\_INSTALL*; PHPUnit\TextUI\Command::*main*(); } else { if (!is\_file(dirname(*\_\_DIR\_\_*).'/vendor/symfony/phpunit-bridge/bin/simple-phpunit.php')) { echo "Unable to find the `simple-phpunit.php` script in `vendor/symfony/phpunit-bridge/bin/`.\n"; exit(1); } require dirname(*\_\_DIR\_\_*).'/vendor/symfony/phpunit-bridge/bin/simple-phpunit.php'; }

config/packages/cache.yaml

framework:  
 cache:  
 *# Unique name of your app: used to compute stable namespaces for cache keys.  
 #prefix\_seed: your\_vendor\_name/app\_name  
  
 # The "app" cache stores to the filesystem by default.  
 # The data in this cache should persist between deploys.  
 # Other options include:  
  
 # Redis  
 #app: cache.adapter.redis  
 #default\_redis\_provider: redis://localhost  
  
 # APCu (not recommended with heavy random-write workloads as memory fragmentation can cause perf issues)  
 #app: cache.adapter.apcu  
  
 # Namespaced pools use the above "app" backend by default  
 #pools:  
 #my.dedicated.cache: null*

config/packages/debug.yaml

when@dev:  
 debug:  
 *# Forwards VarDumper Data clones to a centralized server allowing to inspect dumps on CLI or in your browser.  
 # See the "server:dump" command to start a new server.* dump\_destination: "tcp://%env(VAR\_DUMPER\_SERVER)%"

config/packages/doctrine.yaml

doctrine:  
 dbal:  
 url: '%env(resolve:DATABASE\_URL)%'  
  
 *# IMPORTANT: You MUST configure your server version,  
 # either here or in the DATABASE\_URL env var (see .env file)  
 #server\_version: '15'* orm:  
 auto\_generate\_proxy\_classes: true  
 enable\_lazy\_ghost\_objects: true  
 naming\_strategy: doctrine.orm.naming\_strategy.underscore\_number\_aware  
 auto\_mapping: true  
 mappings:  
 App:  
 is\_bundle: false  
 dir: '%kernel.project\_dir%/src/Entity'  
 prefix: 'App\Entity'  
 alias: App  
  
when@test:  
 doctrine:  
 dbal:  
 *# "TEST\_TOKEN" is typically set by ParaTest* dbname\_suffix: '\_test%env(default::TEST\_TOKEN)%'  
  
when@prod:  
 doctrine:  
 orm:  
 auto\_generate\_proxy\_classes: false  
 proxy\_dir: '%kernel.build\_dir%/doctrine/orm/Proxies'  
 query\_cache\_driver:  
 type: pool  
 pool: doctrine.system\_cache\_pool  
 result\_cache\_driver:  
 type: pool  
 pool: doctrine.result\_cache\_pool  
  
 framework:  
 cache:  
 pools:  
 doctrine.result\_cache\_pool:  
 adapter: cache.app  
 doctrine.system\_cache\_pool:  
 adapter: cache.system

config/packages/doctrine\_migrations.yaml

doctrine\_migrations:  
 migrations\_paths:  
 *# namespace is arbitrary but should be different from App\Migrations  
 # as migrations classes should NOT be autoloaded* 'DoctrineMigrations': '%kernel.project\_dir%/migrations'  
 enable\_profiler: false

config/packages/framework.yaml

# see https://symfony.com/doc/current/reference/configuration/framework.html  
framework:  
 secret: '%env(APP\_SECRET)%'  
 http\_method\_override: false  
 handle\_all\_throwables: true  
 csrf\_protection: ~  
  
 # Enables session support. Note that the session will ONLY be started if you read or write from it.  
 # Remove or comment this section to explicitly disable session support.  
 session:  
 handler\_id: null  
 cookie\_secure: auto  
 cookie\_samesite: lax  
 storage\_factory\_id: session.storage.factory.native  
  
 #esi: true  
 #fragments: true  
 php\_errors:  
 log: true  
  
when@test:  
 framework:  
 test: true  
 session:  
 storage\_factory\_id: session.storage.factory.mock\_file

config/packages/google\_recaptcha.yaml

services:  
  
 *# Inject this service in your controllers/services to verify a submitted captcha.* ReCaptcha\ReCaptcha:  
 arguments:  
 $secret: '%env(GOOGLE\_RECAPTCHA\_SECRET)%'  
 $requestMethod: '@ReCaptcha\RequestMethod'  
  
 *# Curl is set here as default transport to communicate with Google servers.  
 # If you do not have php-curl extension, you can change for a socket or a plain POST request.  
 # Check out the repository for all other request methods:  
 # https://github.com/google/recaptcha/tree/master/src/ReCaptcha/RequestMethod* ReCaptcha\RequestMethod: '@ReCaptcha\RequestMethod\CurlPost'  
 ReCaptcha\RequestMethod\CurlPost: null  
 ReCaptcha\RequestMethod\Curl: null  
  
*# Uncomment this line if you want to inject the site key to all your Twig templates.  
# You can also inject the "google\_recaptcha\_site\_key" container parameter to your controllers.  
#twig:  
# globals:  
# google\_recaptcha\_site\_key: '%google\_recaptcha\_site\_key%'*

config/packages/messenger.yaml

services:  
  
 *# Inject this service in your controllers/services to verify a submitted captcha.* ReCaptcha\ReCaptcha:  
 arguments:  
 $secret: '%env(GOOGLE\_RECAPTCHA\_SECRET)%'  
 $requestMethod: '@ReCaptcha\RequestMethod'  
  
 *# Curl is set here as default transport to communicate with Google servers.  
 # If you do not have php-curl extension, you can change for a socket or a plain POST request.  
 # Check out the repository for all other request methods:  
 # https://github.com/google/recaptcha/tree/master/src/ReCaptcha/RequestMethod* ReCaptcha\RequestMethod: '@ReCaptcha\RequestMethod\CurlPost'  
 ReCaptcha\RequestMethod\CurlPost: null  
 ReCaptcha\RequestMethod\Curl: null  
  
*# Uncomment this line if you want to inject the site key to all your Twig templates.  
# You can also inject the "google\_recaptcha\_site\_key" container parameter to your controllers.  
#twig:  
# globals:  
# google\_recaptcha\_site\_key: '%google\_recaptcha\_site\_key%'*

config/packages/monolog.yaml

monolog:  
 channels:  
 - deprecation *# Deprecations are logged in the dedicated "deprecation" channel when it exists*when@dev:  
 monolog:  
 handlers:  
 main:  
 type: stream  
 path: "%kernel.logs\_dir%/%kernel.environment%.log"  
 level: debug  
 channels: ["!event"]  
 *# uncomment to get logging in your browser  
 # you may have to allow bigger header sizes in your Web server configuration  
 #firephp:  
 # type: firephp  
 # level: info  
 #chromephp:  
 # type: chromephp  
 # level: info* console:  
 type: console  
 process\_psr\_3\_messages: false  
 channels: ["!event", "!doctrine", "!console"]  
  
when@test:  
 monolog:  
 handlers:  
 main:  
 type: fingers\_crossed  
 action\_level: error  
 handler: nested  
 excluded\_http\_codes: [404, 405]  
 channels: ["!event"]  
 nested:  
 type: stream  
 path: "%kernel.logs\_dir%/%kernel.environment%.log"  
 level: debug  
  
when@prod:  
 monolog:  
 handlers:  
 main:  
 type: fingers\_crossed  
 action\_level: error  
 handler: nested  
 excluded\_http\_codes: [404, 405]  
 buffer\_size: 50 *# How many messages should be saved? Prevent memory leaks* nested:  
 type: stream  
 path: php://stderr  
 level: debug  
 formatter: monolog.formatter.json  
 console:  
 type: console  
 process\_psr\_3\_messages: false  
 channels: ["!event", "!doctrine"]  
 deprecation:  
 type: stream  
 channels: [deprecation]  
 path: php://stderr

config/packages/notifier.yaml

framework:  
 notifier:  
 *#chatter\_transports:  
 # slack: '%env(SLACK\_DSN)%'  
 # telegram: '%env(TELEGRAM\_DSN)%'  
 #texter\_transports:  
 # twilio: '%env(TWILIO\_DSN)%'  
 # nexmo: '%env(NEXMO\_DSN)%'* channel\_policy:  
 *# use chat/slack, chat/telegram, sms/twilio or sms/nexmo* urgent: ['email']  
 high: ['email']  
 medium: ['email']  
 low: ['email']  
 admin\_recipients:  
 - { email: admin@example.com }

config/packages/recaptcha.yaml

recaptcha:  
 google\_site\_key: '%env(GOOGLE\_RECAPTCHA\_SITE\_KEY)%'  
 google\_secret\_key: '%env(GOOGLE\_RECAPTCHA\_SECRET\_KEY)%'

config/packages/routing.yaml

framework:  
 router:  
 utf8: true  
  
 *# Configure how to generate URLs in non-HTTP contexts, such as CLI commands.  
 # See https://symfony.com/doc/current/routing.html#generating-urls-in-commands  
 #default\_uri: http://localhost*when@prod:  
 framework:  
 router:  
 strict\_requirements: null

config/packages/security.yaml

security:  
 *# https://symfony.com/doc/current/security.html#registering-the-user-hashing-passwords* password\_hashers:  
 Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface: 'auto'  
 *# https://symfony.com/doc/current/security.html#loading-the-user-the-user-provider* providers:  
 *# used to reload user from session & other features (e.g. switch\_user)* app\_user\_provider:  
 entity:  
 class: App\Entity\User  
 property: username  
 firewalls:  
 dev:  
 pattern: ^/(\_(profiler|wdt)|css|images|js)/  
 security: false  
 main:  
 lazy: true  
 provider: app\_user\_provider  
 remember-me:  
 secret: '%kernel.secret%' *# required* lifetime: 604800 *# 1 week in seconds* form\_login:  
 login\_path: login  
 check\_path: login  
 default\_target\_path: mainPage  
 username\_parameter: "registration\_form[username]"  
 password\_parameter: "registration\_form[password]"  
 logout:  
 path: logout  
 target: mainPage  
 *# activate different ways to authenticate  
 # https://symfony.com/doc/current/security.html#the-firewall  
  
 # https://symfony.com/doc/current/security/impersonating\_user.html  
 # switch\_user: true* role\_hierarchy:  
 ROLE\_EDITOR: ROLE\_USER  
 ROLE\_ADMIN: ROLE\_EDITOR  
 *# Easy way to control access for large sections of your site  
 # Note: Only the \*first\* access control that matches will be used* access\_control:  
 - { path: ^/(userCab|test/(list|\d+)), roles: ROLE\_USER }  
 - { path: ^/(edit|delete|create)/(lecture|test)(|/\d+$), roles: ROLE\_EDITOR }  
 - { path: ^/(admin|user/delete/\d+$), roles: ROLE\_ADMIN }  
  
when@test:  
 security:  
 password\_hashers:  
 *# By default, password hashers are resource intensive and take time. This is  
 # important to generate secure password hashes. In tests however, secure hashes  
 # are not important, waste resources and increase test times. The following  
 # reduces the work factor to the lowest possible values.* Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface:  
 algorithm: auto  
 cost: 4 *# Lowest possible value for bcrypt* time\_cost: 3 *# Lowest possible value for argon* memory\_cost: 10 *# Lowest possible value for argon*

config/packages/sensio\_framework\_extra.yaml

sensio\_framework\_extra:  
 router:  
 annotations: false

config/packages/translation.yaml

framework:  
 default\_locale: ru  
 translator:  
 default\_path: '%kernel.project\_dir%/translations'  
 fallbacks:  
 - en  
*# providers:  
# crowdin:  
# dsn: '%env(CROWDIN\_DSN)%'  
# loco:  
# dsn: '%env(LOCO\_DSN)%'  
# lokalise:  
# dsn: '%env(LOKALISE\_DSN)%'*

config/packages/twig.yaml

twig:  
 form\_themes: ['bootstrap\_5\_layout.html.twig']  
 default\_path: '%kernel.project\_dir%/templates'  
  
when@test:  
 twig:  
 strict\_variables: true

config/packages/validator.yaml

framework:  
 validation:  
 email\_validation\_mode: html5  
  
 *# Enables validator auto-mapping support.  
 # For instance, basic validation constraints will be inferred from Doctrine's metadata.  
 #auto\_mapping:  
 # App\Entity\: []*when@test:  
 framework:  
 validation:  
 not\_compromised\_password: false

config/packages/web\_profiler.yaml

when@dev:  
 web\_profiler:  
 toolbar: true  
 intercept\_redirects: false  
  
 framework:  
 profiler:  
 only\_exceptions: false  
 collect\_serializer\_data: true  
  
when@test:  
 web\_profiler:  
 toolbar: false  
 intercept\_redirects: false  
  
 framework:  
 profiler: { collect: false }

config/packages/white\_october\_breadcrumbs.yaml

white\_october\_breadcrumbs: ~

config/routes/attributes.yaml

controllers:  
 resource:  
 path: ../../src/Controller/  
 namespace: App\Controller  
 type: attribute

config/routes/framework.yaml

when@dev:  
 \_errors:  
 resource: '@FrameworkBundle/Resources/config/routing/errors.xml'  
 prefix: /\_error

config/routes/web\_profiler.yaml

when@dev:  
 web\_profiler\_wdt:  
 resource: '@WebProfilerBundle/Resources/config/routing/wdt.xml'  
 prefix: /\_wdt  
  
 web\_profiler\_profiler:  
 resource: '@WebProfilerBundle/Resources/config/routing/profiler.xml'  
 prefix: /\_profiler

config/validator/validation.yml

App\Entity\User:  
 properties:  
 username:  
 - NotBlank: ~  
 - Length:  
 min: 2  
 max: 4096  
 minMessage: 'Имя пользователя должно содержать не менее {{ limit }} символов'  
 maxMessage: 'Имя пользователя не должно превышать {{ limit }} символов'  
 password:  
 - NotBlank: ~

config/bundles.php

<?php return [Symfony\Bundle\FrameworkBundle\FrameworkBundle::class => ['all' => true], Doctrine\Bundle\DoctrineBundle\DoctrineBundle::class => ['all' => true], Doctrine\Bundle\MigrationsBundle\DoctrineMigrationsBundle::class => ['all' => true], Symfony\Bundle\DebugBundle\DebugBundle::class => ['dev' => true], Symfony\Bundle\TwigBundle\TwigBundle::class => ['all' => true], Symfony\Bundle\WebProfilerBundle\WebProfilerBundle::class => ['dev' => true, 'test' => true], Twig\Extra\TwigExtraBundle\TwigExtraBundle::class => ['all' => true], Symfony\Bundle\SecurityBundle\SecurityBundle::class => ['all' => true], Symfony\Bundle\MonologBundle\MonologBundle::class => ['all' => true], Symfony\Bundle\MakerBundle\MakerBundle::class => ['dev' => true], Sensio\Bundle\FrameworkExtraBundle\SensioFrameworkExtraBundle::class => ['all' => true], Captcha\Bundle\CaptchaBundle\CaptchaBundle::class => ['all' => true], VictorPrdh\RecaptchaBundle\RecaptchaBundle::class => ['all' => true], WhiteOctober\BreadcrumbsBundle\WhiteOctoberBreadcrumbsBundle::class => ['all' => true],];

config/preload.php

<?php if (file\_exists(dirname(*\_\_DIR\_\_*).'/var/cache/prod/App\_KernelProdContainer.preload.php')) {require dirname(*\_\_DIR\_\_*).'/var/cache/prod/App\_KernelProdContainer.preload.php';}

config/routes.yaml

mainPage:  
 path: /  
 controller: App\Controller\SiteController::index  
lectures\_list:  
 path: /lecture/list  
 controller: App\Controller\SiteController::lectures\_list  
tests\_list:  
 path: /test/list  
 controller: App\Controller\SiteController::tests\_list  
about:  
 path: /about  
 controller: App\Controller\SiteController::about  
cab:  
 path: /userCab  
 controller: App\Controller\SiteController::userCab  
login:  
 path: /login  
 controller: App\Controller\SiteController::login  
reg:  
 path: /reg  
 controller: App\Controller\SiteController::reg  
logout:  
 path: /logout  
 controller: App\Controller\SiteController::logout  
lecture:  
 path: /lecture/{id}  
 controller: App\Controller\SiteController::showLecture  
test:  
 path: /test/{id}  
 controller: App\Controller\SiteController::showTest  
admin:  
 path: /admin  
 controller: App\Controller\AdminController::admin  
createLecture:  
 path: /create/lecture  
 controller: App\Controller\AdminController::createLecture  
editLecture:  
 path: /edit/lecture/{id}  
 controller: App\Controller\AdminController::editLecture  
deleteLecture:  
 path: /delete/lecture/{id}  
 controller: App\Controller\AdminController::deleteLecture  
addTest:  
 path: /create/test  
 controller: App\Controller\AdminController::createTest  
deleteTest:  
 path: /delete/test/{id}  
 controller: App\Controller\AdminController::deleteTest  
deleteUser:  
 path: /user/delete/{id}  
 controller: App\Controller\AdminController::deleteUser

config/services.yaml

# This file is the entry point to configure your own services.  
# Files in the packages/ subdirectory configure your dependencies.  
  
# Put parameters here that don't need to change on each machine where the app is deployed  
# https://symfony.com/doc/current/best\_practices.html#use-parameters-for-application-configuration  
parameters:  
 google\_recaptcha\_site\_key: '%env(GOOGLE\_RECAPTCHA\_SITE\_KEY)%'  
  
services:  
 # default configuration for services in \*this\* file  
 \_defaults:  
 autowire: true # Automatically injects dependencies in your services.  
 autoconfigure: true # Automatically registers your services as commands, event subscribers, etc.  
  
 # makes classes in src/ available to be used as services  
 # this creates a service per class whose id is the fully-qualified class name  
 App\:  
 resource: '../src/'  
 exclude:  
 - '../src/DependencyInjection/'  
 - '../src/Entity/'  
 - '../src/Kernel.php'  
 App\Controller\SiteController:  
 arguments:  
 $authenticator: '@security.authenticator.form\_login.main'  
 # add more service definitions when explicit configuration is needed  
 # please note that last definitions always \*replace\* previous ones

migrations/Version20230309094144.php

<?php  
  
declare(strict\_types=1);  
  
namespace DoctrineMigrations;  
  
use Doctrine\DBAL\Schema\Schema;  
use Doctrine\Migrations\AbstractMigration;  
  
*/\*\*  
 \* Auto-generated Migration: Please modify to your needs!  
 \*/*final class Version20230309094144 extends AbstractMigration  
{  
 public function getDescription(): string  
 {  
 return '';  
 }  
  
 public function up(Schema $schema): void  
 {  
 // this up() migration is auto-generated, please modify it to your needs  
 $this->addSql('CREATE TABLE questions (id INT AUTO\_INCREMENT NOT NULL, tests\_id INT NOT NULL, question LONGTEXT NOT NULL, INDEX IDX\_8ADC54D5F5D80971 (tests\_id), PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4\_unicode\_ci` ENGINE = InnoDB');  
 $this->addSql('CREATE TABLE tests (id INT AUTO\_INCREMENT NOT NULL, name VARCHAR(255) NOT NULL, PRIMARY KEY(id)) DEFAULT CHARACTER SET utf8mb4 COLLATE `utf8mb4\_unicode\_ci` ENGINE = InnoDB');  
 $this->addSql('ALTER TABLE questions ADD CONSTRAINT FK\_8ADC54D5F5D80971 FOREIGN KEY (tests\_id) REFERENCES tests (id)');  
 $this->addSql('ALTER TABLE answers ADD questions\_id INT NOT NULL');  
 $this->addSql('ALTER TABLE answers ADD CONSTRAINT FK\_50D0C606BCB134CE FOREIGN KEY (questions\_id) REFERENCES questions (id)');  
 $this->addSql('CREATE INDEX IDX\_50D0C606BCB134CE ON answers (questions\_id)');  
 }  
  
 public function down(Schema $schema): void  
 {  
 // this down() migration is auto-generated, please modify it to your needs  
 $this->addSql('ALTER TABLE answers DROP FOREIGN KEY FK\_50D0C606BCB134CE');  
 $this->addSql('ALTER TABLE tests\_of\_user DROP FOREIGN KEY FK\_AD45C5E6F5D80971');  
 $this->addSql('ALTER TABLE questions DROP FOREIGN KEY FK\_8ADC54D5F5D80971');  
 $this->addSql('DROP TABLE questions');  
 $this->addSql('DROP TABLE tests');  
 $this->addSql('DROP INDEX IDX\_50D0C606BCB134CE ON answers');  
 $this->addSql('ALTER TABLE answers DROP questions\_id');  
 }  
}

public/css/public/publicInner.css

@import url("https://fonts.googleapis.com/css?family=Montserrat&display=swap");  
  
.montserrat {  
 font-family: 'Montserrat';  
 font-style: normal;  
 color: white **!important**;  
 font-size: clamp(14px, 1cqi, 24px);  
 font-weight: 700;  
}  
  
.darkText {  
 font-weight: 400;  
 color: #000000;  
}  
  
.wrapper {  
 width: 100%;  
 height: 100%;  
 conteiner-type: size;  
}  
  
.img {  
 width: 100%;  
 height: 100%;  
 position: relative;  
}  
  
\*, \*:after, \*:before {  
 padding: 0;  
 margin: 0;  
 box-sizing: border-box;  
}  
  
.button:active {  
 transition: all 0.1s ease-in-out;  
 box-shadow: 0 4px 4px rgba(0, 0, 0, 0.15);  
}  
  
body {  
 overflow: auto;  
 overflow-x: hidden;  
}  
  
a {  
 text-decoration: none **!important**;  
 box-sizing: content-box;  
}  
  
a:hover > \* {  
 color: #84D98A;  
 transition: all .3s ease-in-out;  
}

public/css/public/reset.css

/\*! normalize.css v8.0.1 | MIT License | github.com/necolas/normalize.css \*/  
  
/\* Document  
 ========================================================================== \*/  
  
/\*\*  
 \* 1. Correct the line height in all browsers.  
 \* 2. Prevent adjustments of font size after orientation changes in iOS.  
 \*/  
  
html {  
 line-height: 1.15; /\* 1 \*/  
 -webkit-text-size-adjust: 100%; /\* 2 \*/  
}  
  
/\* Sections  
 ========================================================================== \*/  
  
/\*\*  
 \* Remove the margin in all browsers.  
 \*/  
  
body {  
 margin: 0;  
}  
  
/\*\*  
 \* Render the `main` element consistently in IE.  
 \*/  
  
main {  
 display: block;  
}  
  
/\*\*  
 \* Correct the font size and margin on `h1` elements within `section` and  
 \* `article` contexts in Chrome, Firefox, and Safari.  
 \*/  
  
h1 {  
 font-size: 2em;  
 margin: 0.67em 0;  
}  
  
/\* Grouping content  
 ========================================================================== \*/  
  
/\*\*  
 \* 1. Add the correct box sizing in Firefox.  
 \* 2. Show the overflow in Edge and IE.  
 \*/  
  
hr {  
 box-sizing: content-box; /\* 1 \*/  
 height: 0; /\* 1 \*/  
 overflow: visible; /\* 2 \*/  
}  
  
/\*\*  
 \* 1. Correct the inheritance and scaling of font size in all browsers.  
 \* 2. Correct the odd `em` font sizing in all browsers.  
 \*/  
  
pre {  
 font-family: monospace, monospace; /\* 1 \*/  
 font-size: 1em; /\* 2 \*/  
}  
  
/\* Text-level semantics  
 ========================================================================== \*/  
  
/\*\*  
 \* Remove the gray background on active links in IE 10.  
 \*/  
  
a {  
 background-color: transparent;  
}  
  
/\*\*  
 \* 1. Remove the bottom border in Chrome 57-  
 \* 2. Add the correct text decoration in Chrome, Edge, IE, Opera, and Safari.  
 \*/  
  
abbr[title] {  
 border-bottom: none; /\* 1 \*/  
 text-decoration: underline; /\* 2 \*/  
 text-decoration: underline dotted; /\* 2 \*/  
}  
  
/\*\*  
 \* Add the correct font weight in Chrome, Edge, and Safari.  
 \*/  
  
b,  
strong {  
 font-weight: bolder;  
}  
  
/\*\*  
 \* 1. Correct the inheritance and scaling of font size in all browsers.  
 \* 2. Correct the odd `em` font sizing in all browsers.  
 \*/  
  
code,  
kbd,  
samp {  
 font-family: monospace, monospace; /\* 1 \*/  
 font-size: 1em; /\* 2 \*/  
}  
  
/\*\*  
 \* Add the correct font size in all browsers.  
 \*/  
  
small {  
 font-size: 80%;  
}  
  
/\*\*  
 \* Prevent `sub` and `sup` elements from affecting the line height in  
 \* all browsers.  
 \*/  
  
sub,  
sup {  
 font-size: 75%;  
 line-height: 0;  
 position: relative;  
 vertical-align: baseline;  
}  
  
sub {  
 bottom: -0.25em;  
}  
  
sup {  
 top: -0.5em;  
}  
  
/\* Embedded content  
 ========================================================================== \*/  
  
/\*\*  
 \* Remove the border on images inside links in IE 10.  
 \*/  
  
img {  
 border-style: none;  
}  
  
/\* Forms  
 ========================================================================== \*/  
  
/\*\*  
 \* 1. Change the font styles in all browsers.  
 \* 2. Remove the margin in Firefox and Safari.  
 \*/  
  
button,  
input,  
optgroup,  
select,  
textarea {  
 font-family: inherit; /\* 1 \*/  
 font-size: 100%; /\* 1 \*/  
 line-height: 1.15; /\* 1 \*/  
 margin: 0; /\* 2 \*/  
}  
  
/\*\*  
 \* Show the overflow in IE.  
 \* 1. Show the overflow in Edge.  
 \*/  
  
button,  
input { /\* 1 \*/  
 overflow: visible;  
}  
  
/\*\*  
 \* Remove the inheritance of text transform in Edge, Firefox, and IE.  
 \* 1. Remove the inheritance of text transform in Firefox.  
 \*/  
  
button,  
select { /\* 1 \*/  
 text-transform: none;  
}  
  
/\*\*  
 \* Correct the inability to style clickable types in iOS and Safari.  
 \*/  
  
button,  
[type="button"],  
[type="reset"],  
[type="submit"] {  
 -webkit-appearance: button;  
}  
  
/\*\*  
 \* Remove the inner border and padding in Firefox.  
 \*/  
  
button::-moz-focus-inner,  
[type="button"]::-moz-focus-inner,  
[type="reset"]::-moz-focus-inner,  
[type="submit"]::-moz-focus-inner {  
 border-style: none;  
 padding: 0;  
}  
  
/\*\*  
 \* Restore the focus styles unset by the previous rule.  
 \*/  
  
button:-moz-focusring,  
[type="button"]:-moz-focusring,  
[type="reset"]:-moz-focusring,  
[type="submit"]:-moz-focusring {  
 outline: 1px dotted ButtonText;  
}  
  
/\*\*  
 \* Correct the padding in Firefox.  
 \*/  
  
fieldset {  
 padding: 0.35em 0.75em 0.625em;  
}  
  
/\*\*  
 \* 1. Correct the text wrapping in Edge and IE.  
 \* 2. Correct the color inheritance from `fieldset` elements in IE.  
 \* 3. Remove the padding so developers are not caught out when they zero out  
 \* `fieldset` elements in all browsers.  
 \*/  
  
legend {  
 box-sizing: border-box; /\* 1 \*/  
 color: inherit; /\* 2 \*/  
 display: table; /\* 1 \*/  
 max-width: 100%; /\* 1 \*/  
 padding: 0; /\* 3 \*/  
 white-space: normal; /\* 1 \*/  
}  
  
/\*\*  
 \* Add the correct vertical alignment in Chrome, Firefox, and Opera.  
 \*/  
  
progress {  
 vertical-align: baseline;  
}  
  
/\*\*  
 \* Remove the default vertical scrollbar in IE 10+.  
 \*/  
  
textarea {  
 overflow: auto;  
}  
  
/\*\*  
 \* 1. Add the correct box sizing in IE 10.  
 \* 2. Remove the padding in IE 10.  
 \*/  
  
[type="checkbox"],  
[type="radio"] {  
 box-sizing: border-box; /\* 1 \*/  
 padding: 0; /\* 2 \*/  
}  
  
/\*\*  
 \* Correct the cursor style of increment and decrement buttons in Chrome.  
 \*/  
  
[type="number"]::-webkit-inner-spin-button,  
[type="number"]::-webkit-outer-spin-button {  
 height: auto;  
}  
  
/\*\*  
 \* 1. Correct the odd appearance in Chrome and Safari.  
 \* 2. Correct the outline style in Safari.  
 \*/  
  
[type="search"] {  
 -webkit-appearance: textfield; /\* 1 \*/  
 outline-offset: -2px; /\* 2 \*/  
}  
  
/\*\*  
 \* Remove the inner padding in Chrome and Safari on macOS.  
 \*/  
  
[type="search"]::-webkit-search-decoration {  
 -webkit-appearance: none;  
}  
  
/\*\*  
 \* 1. Correct the inability to style clickable types in iOS and Safari.  
 \* 2. Change font properties to `inherit` in Safari.  
 \*/  
  
::-webkit-file-upload-button {  
 -webkit-appearance: button; /\* 1 \*/  
 font: inherit; /\* 2 \*/  
}  
  
/\* Interactive  
 ========================================================================== \*/  
  
/\*  
 \* Add the correct display in Edge, IE 10+, and Firefox.  
 \*/  
  
details {  
 display: block;  
}  
  
/\*  
 \* Add the correct display in all browsers.  
 \*/  
  
summary {  
 display: list-item;  
}  
  
/\* Misc  
 ========================================================================== \*/  
  
/\*\*  
 \* Add the correct display in IE 10+.  
 \*/  
  
template {  
 display: none;  
}  
  
/\*\*  
 \* Add the correct display in IE 10.  
 \*/  
  
[hidden] {  
 display: none;  
}

public/css/\_header.css

@import "public.css";  
  
header {  
 height: 100cqw;  
 max-height: 101px;  
 width: 100cqw;  
 box-sizing: border-box;  
 background: linear-gradient(180deg, rgba(38, 38, 38, 0.2) 0%, rgba(132, 217, 138, 0.1) 87.45%), #3D7340;  
 position: fixed;  
 display: flex;  
 margin-bottom: 2%;  
 top: 0;  
 border: none;  
 z-index: 1000;  
 conteiner-type: size;  
}  
  
.logo {  
 justify-self: center;  
 position: relative;  
 margin-left: 9cqw;  
 top: 20%;  
}  
  
.logo-img {  
 max-width: 80px;  
 max-height: 64px;  
}  
  
.links{  
 position: relative;  
 top: 30px;  
 font-weight: 500;  
 display: flex;  
 margin-left: 50px;  
 justify-content: space-between;  
 gap: 10px;  
 justify-self: right;  
 max-height: 40px;  
}  
  
.bread {  
 position: relative;  
 margin-top: 121px;  
 margin-left: clamp(50px, 10cqw, 150px);  
 margin-right: clamp(50px, 10cqw, 200px);  
 margin-bottom: 20px;  
 min-width: 200px;  
}  
  
.bread a span{  
 color: rgba(91, 211, 173, 0.75);  
}  
  
.bread ol {  
 display: flex;  
}  
  
.bread a:hover > \* {  
 color: rgba(0, 79, 7, 0.75);  
 transition: all .3s ease-in-out;  
}  
  
.bread ol li \*:not(:is(a > span)) {  
 color: #3D7340BF;  
}  
  
.bottom-line {  
 border-bottom: 1px solid #606760;  
}  
  
.links-label {  
 display: flex;  
 align-items: center;  
 align-content: space-between;  
 gap: 5px;  
 color: #FFFFFF;  
}  
  
.links-label > svg, .links-label > svg > use {  
 width: 38px;  
 height: 38px;  
}  
  
a:hover #cab {  
 stroke: #84D98A;  
 transition: all .3s ease-in-out;  
}  
  
a:hover svg:not(#cab) \* {  
 fill: #84D98A;  
 stroke: none;  
 transition: all .3s ease-in-out;  
}  
  
ol {  
 list-style: none;  
}  
  
@media screen and (max-width: 1080px) {  
 header {  
 max-height: 202px;  
 flex-direction: column;  
 position: relative;  
 }  
  
 .logo {  
 width: 80px;  
 height: 64px;  
 left: 37%;  
 justify-self: center;  
 top: 10%;  
 }  
  
 .logo-img {  
 max-width: 80px;  
 max-height: 64px;  
 }  
  
 .links{  
 top: 50px;  
 gap: 10px;  
 justify-content: center;  
 flex-wrap: wrap;  
 margin-left: 10px;  
 }  
  
 .bread {  
 margin-top: 21px;  
 }  
  
}

public/css/about.css

.about {  
 margin: 0 10%;  
}  
  
.about-header {  
 margin-left: 45%;  
}  
  
.about-body {  
 line-height: 200%;  
}  
  
ul {  
 list-style-type: square;  
}

public/css/add-edit-lecture-main.css

.main {  
 margin-left: clamp(50px, 10cqw, 150px);  
 margin-right: clamp(50px, 10cqw, 200px);  
 color: #000000;  
}  
  
#lecture\_form\_name {  
 width: 300px;  
}  
#lecture\_form\_content {  
 margin: 30px 0 0 0 ;  
 height: 500px;  
}

public/css/admin.css

.main {  
 margin-left: clamp(50px, 10cqw, 150px);  
 margin-right: clamp(50px, 10cqw, 200px);  
 color: #000000;  
}  
.button-edit svg, .button-delete svg {  
 width: 30px;  
 height: 30px;  
 fill: #000000;  
 stroke: #000000;  
}  
  
:is(.button-delete, .button-edit):hover svg {  
 fill: #84D98A;  
 stroke: #84D98A;  
 transition: all .3s ease-in-out;  
}  
  
:is(.button-edit, .button-delete) {  
 border: none;  
 background: none;  
}

public/css/index-courses.css

.courses {  
 display: flex;  
 justify-content: space-evenly;  
 margin-top: 40px;  
 margin-left: clamp(50px, 10cqw, 200px);  
 margin-right: clamp(50px, 10cqw, 200px);  
}  
  
:is(.free, .with-price) {  
 display: flex;  
 flex-direction: column;  
 border: 1px solid #3D7340;  
 border-radius: 5px;  
 padding: 20px;  
}  
  
:is(.free, .with-price) div {  
 line-height: 150%;  
}  
  
.free h1 {  
 color: #3D7340;  
}  
  
.with-price h1 {  
 color: #92754E;  
}  
  
.price {  
 text-align: center;  
}  
  
:is(.free, .with-price) button {  
 border: none;  
 background-color: #3D7340;  
 color: white;  
 border-radius: 10px;  
 margin: 10px 30px;  
 padding: 10px 0;  
}  
  
ul {  
 list-style: none;  
}

public/css/index-info.css

.info {  
 margin-left: clamp(50px, 10cqw, 150px);  
 margin-right: clamp(50px, 10cqw, 200px);  
 font-size: clamp(20px, 1cqi, 36px);  
 line-height: 150%;  
}

public/css/lecture-main.css

.main {  
 margin-left: clamp(50px, 10cqw, 150px);  
 margin-right: clamp(50px, 10cqw, 200px);  
 color: #000000;  
}

public/css/lecture\_list-main.css

.main {  
 margin-left: clamp(50px, 10cqw, 150px);  
 margin-right: clamp(50px, 10cqw, 200px);  
 color: #000000;  
}  
  
.button-edit svg, .button-delete svg {  
 width: 30px;  
 height: 30px;  
 fill: #000000;  
 stroke: #000000;  
}  
  
:is(.button-delete, .button-edit):hover svg {  
 fill: #84D98A;  
 stroke: #84D98A;  
 transition: all .3s ease-in-out;  
}  
  
:is(.button-edit, .button-delete) {  
 border: none;  
 background: none;  
}  
  
.actions {  
 display: flex;  
}

public/css/login.css

@import "public.css";  
  
body {  
 background: #f2f2f2;  
 display: grid;  
 height: 100vh;  
 grid-template-columns: auto auto;  
}  
  
.logo {  
 width: 747px;  
 height: 100vh;  
 background: linear-gradient(180deg, rgba(38, 38, 38, 0.2) 0%, rgba(132, 217, 138, 0.1) 87.45%), #3D7340;  
}  
  
.logo-img {  
 position: relative;  
 max-width: 300px;  
 max-height: 280px;  
 margin: auto;  
 top: 35%;  
 left: 27.8%;  
}  
  
.login {  
 width: 384px;  
 height: 333px;  
 position: absolute;  
 margin: auto;  
 top: 25vh;  
 left: 60vw;  
}  
  
.title {  
 position: relative;  
 color: #262626;  
 width: 120px;  
 height: 55px;  
 font-size: 2em;  
 line-height: 55px;  
 left: 15%;  
}  
  
.form-floating input {  
 background: #ffffff;  
 border: 3px solid rgba(61, 115, 64, 0.75);  
 color: rgba(38, 40, 64, 0.5);  
 text-indent: 30px;  
 outline: none;  
}  
.form-floating:first-of-type input {  
 border-radius: 15px 15px 0 0;  
}  
  
.form-floating:not(:first-of-type) input {  
 border-radius: 0 0 15px 15px;  
}  
  
.form-floating input:focus {  
 border-color: #a68a56;  
}  
  
.login button[type=submit] {  
 color: #ffffff;  
 background: #3D7340;  
 border-radius: 15px;  
}  
  
.login button[type=submit]:hover {  
 background: #a68a56;  
}  
  
.login button[type=submit]:active {  
 transition: all 0.05s ease-in-out;  
 background: rgba(166, 138, 86, 0.5);  
}  
  
.bottom-title {  
 display: grid;  
 text-align: center;  
 margin-top: 50px;  
}  
  
.link {  
 text-decoration: none;  
 color: rgba(61, 115, 64, 0.75);  
 font-size: 22px;  
 line-height: 30px;  
}  
  
@media (max-width: 1366px) {  
 .logo {  
 width: 530px;  
 }  
  
 .login {  
 top: 20vh;  
 left: 57%;  
 }  
  
 .logo-img {  
 width: 231px;  
 height: 338px;  
 top: 25%;  
 left: 25%;  
 }  
  
 .title {  
 font-size: 25px;  
 line-height: 25px;  
 }  
  
 .field {  
 width: 280px;  
 height: 50px;  
 border: 2px solid rgba(61, 115, 64, 0.75);  
 font-size: 15px;  
 margin-top: 40px;  
 margin-left: 52px;  
 }  
  
 .field[type=submit] {  
 width: 180px;  
 height: 60px;  
 font-size: 17px;  
 border-radius: 15px;  
 margin-left: 100px;  
 }  
  
 .subtitle {  
 font-size: 15px;  
 line-height: 15px;  
 }  
  
 .link {  
 text-decoration: none;  
 color: rgba(61, 115, 64, 0.75);  
 font-size: 15px;  
 line-height: 30px;  
 }  
}

public/css/public.css

@import "public/reset.css";  
@import "public/publicInner.css";

public/css/test-main.css

.main {  
 margin-left: clamp(50px, 10cqw, 150px);  
 margin-right: clamp(50px, 10cqw, 200px);  
 color: #000000;  
}  
  
.test-header {  
 margin-left: 45%;  
}

public/css/test\_list-main.css

.main {  
 margin-left: clamp(50px, 10cqw, 150px);  
 margin-right: clamp(50px, 10cqw, 200px);  
 color: #000000;  
}  
  
.button-edit svg, .button-delete svg {  
 width: 30px;  
 height: 30px;  
 fill: #000000;  
 stroke: #000000;  
}  
  
:is(.button-delete, .button-edit):hover svg {  
 fill: #84D98A;  
 stroke: #84D98A;  
 transition: all .3s ease-in-out;  
}  
  
:is(.button-edit, .button-delete) {  
 border: none;  
 background: none;  
}  
  
.actions {  
 display: flex;  
}

public/css/userCab-main.css

.main {  
 margin: 10cqh 10cqw;  
 background-color: #F2F2F2;  
}  
  
.col {  
 display: grid;  
 container-type: size;  
}  
  
.row > \* {  
 padding-bottom: 30px;  
 padding-top: 30px;  
}

src/controller/AdminController.php

<?php namespace App\Controller;use App\Entity\Answers;use App\Entity\Lectures;use App\Entity\Questions;use App\Entity\Tests;use App\Entity\User;use App\Form\LectureFormType;use App\Form\TestFormType;use Doctrine\ORM\EntityManagerInterface;use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;use Symfony\Component\HttpFoundation\Request;use Symfony\Component\HttpFoundation\Response;use Symfony\Component\Routing\Annotation\Route;use WhiteOctober\BreadcrumbsBundle\Model\Breadcrumbs;class AdminController extends AbstractController{#[Route('/admin', name:'admin')] public function admin(EntityManagerInterface $manager, Request $request, Breadcrumbs $breadcrumbs): Response{$links = ['Главная' => "/", 'Личный кабинет' => "/userCab", 'Админ панель' => ""];$this->createBreadcrumb($links, $breadcrumbs);$array = ['this' => 'Создать лекцию', 'users' => $manager->getRepository(User::class)->findAll(), 'roles' => [0 => 'ROLE\_USER', 1 => 'ROLE\_EDITOR', 2 => 'ROLE\_ADMIN',],];return $this->render('admin.html.twig', $array);}public function createBreadcrumb($array, $breadcrumbs){foreach ($array as $name => $link) {$breadcrumbs->addItem($name, $link);}}#[Route('/create/lecture', name:'addLecture', priority: 1)] public function createLecture(EntityManagerInterface $manager, Request $request, Breadcrumbs $breadcrumbs): Response{$lectures = new Lectures();$form = $this->createForm(LectureFormType::class, $lectures);$links = ['Главная' => "/", 'Лекции' => "/lecture/list", 'Создать лекцию' => ""];$this->createBreadcrumb($links, $breadcrumbs);$form->handleRequest($request);if ($form->isSubmitted() && $form->isValid()) {$lectures = $form->getData();$manager->persist($lectures);$manager->flush();return $this->redirectToRoute('lectures\_list');}$array = ['form' => $form, 'this' => 'Создать лекцию',];return $this->render('add-edit-lecture.html.twig', $array);}#[Route('/edit/lecture/{id}', name:'editLecture')] public function editLecture(EntityManagerInterface $manager, Request $request, Breadcrumbs $breadcrumbs, int $id): Response{$lectures = $manager->getRepository(Lectures::class)->find($id);$form = $this->createForm(LectureFormType::class, $lectures);$links = ['Главная' => "/", 'Лекции' => "/lecture/list", 'Отредактировать лекцию' => ""];$this->createBreadcrumb($links, $breadcrumbs);$form->handleRequest($request);if ($form->isSubmitted() && $form->isValid()) {$manager->flush();return $this->redirectToRoute('lectures\_list');}$array = ['form' => $form, 'this' => 'Отредактировать лекцию',];return $this->render('add-edit-lecture.html.twig', $array);}#[Route('/delete/lecture/{id}', name:'deleteLecture')] public function deleteLecture(EntityManagerInterface $manager, int $id): Response{$manager->getRepository(Lectures::class)->remove($manager->getRepository(Lectures::class)->find($id), true);return $this->redirectToRoute('lectures\_list');}#[Route('/create/test', name:'addTest', priority: 30)] public function createTest(EntityManagerInterface $manager, Request $request, Breadcrumbs $breadcrumbs,): Response{$test = new Tests();$test->setName("");$form = $this->createForm(TestFormType::class, $test);$links = ['Главная' => "/", 'Тесты' => "/test/list", 'Создать тест' => ""];$this->createBreadcrumb($links, $breadcrumbs);$form->handleRequest($request);if ($form->isSubmitted() && $form->isValid()) {$test = $form->getData();$manager->persist($test);$manager->flush();return $this->redirectToRoute('tests\_list');}$array = ['this' => 'Создать тест', 'form' => $form, 'test' => $test];return $this->render('add-edit-test.html.twig', $array);}#[Route('/edit/test/{id}', name:'editTest')] public function editTest(EntityManagerInterface $manager, Request $request, Breadcrumbs $breadcrumbs, int $id): Response{$test = $manager->getRepository(Tests::class)->find($id);$form = $this->createForm(TestFormType::class, $test);$links = ['Главная' => "/", 'Тесты' => "/test/list", 'Отредактировать тест' => ""];$this->createBreadcrumb($links, $breadcrumbs);$form->handleRequest($request);if ($form->isSubmitted() && $form->isValid()) {$manager->flush();return $this->redirectToRoute('tests\_list');}$array = ['this' => 'Создать тест', 'form' => $form,];return $this->render('add-edit-test.html.twig', $array);}#[Route('/delete/test/{id}', name:'deleteTest')] public function deleteTest(EntityManagerInterface $manager, int $id): Response{$manager->getRepository(Tests::class)->remove($manager->getRepository(Tests::class)->find($id), true);return $this->redirectToRoute('tests\_list');}}

src/controller/SiteController.php

<?php namespace App\Controller;use App\Entity\Answers;use App\Entity\Lectures;use App\Entity\Questions;use App\Entity\Tests;use App\Entity\TestsOfUser;use App\Entity\User;use App\Form\RegistrationFormType;use App\Service\IdentifyService;use Doctrine\ORM\EntityManagerInterface;use Psr\Log\LoggerInterface;use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;use Symfony\Component\Form\Form;use Symfony\Component\HttpFoundation\Request;use Symfony\Component\HttpFoundation\Response;use Symfony\Component\PasswordHasher\Hasher\UserPasswordHasherInterface;use Symfony\Component\Routing\Annotation\Route;use Symfony\Component\Security\Http\Authentication\UserAuthenticatorInterface;use Symfony\Component\Security\Http\Authenticator\FormLoginAuthenticator;use Symfony\Component\Security\Http\Authenticator\Passport\Badge\RememberMeBadge;use WhiteOctober\BreadcrumbsBundle\Model\Breadcrumbs;class SiteController extends AbstractController{private EntityManagerInterface $manager;public function \_\_construct(EntityManagerInterface $manager, private FormLoginAuthenticator $authenticator){$this->manager = $manager;}#[Route('/', name: "mainPage")] public function index(Breadcrumbs $breadcrumbs): Response{$links = ['Главная' => ""];$this->createBreadcrumb($links, $breadcrumbs);$array = ['this' => 'Главная',];return $this->render('index.html.twig', $array);}public function createBreadcrumb($array, $breadcrumbs){foreach ($array as $name => $link) {$breadcrumbs->addItem($name, $link);}}#[Route('/lecture/list', name: "lectures\_list", priority: 1)] public function lectures\_list(Breadcrumbs $breadcrumbs): Response{$links = ['Главная' => "/", 'Лекции' => ""];$this->createBreadcrumb($links, $breadcrumbs);$array = ['this' => 'Лекции', 'lectures' => $this->manager->getRepository(Lectures::class)->findAll()];return $this->render('lectures.html.twig', $array);}#[Route('/lecture/{id}', name: "lecture")] public function showLecture(EntityManagerInterface $manager, Breadcrumbs $breadcrumbs, int $id): Response{$links = ['Главная' => "/", 'Лекции' => "/lecture/list", 'Лекция' => ""];$this->createBreadcrumb($links, $breadcrumbs);$array = ['this' => 'Лекция', 'lecture' => $manager->getRepository(Lectures::class)->find($id)];return $this->render('lecture.html.twig', $array);}#[Route('/test/list', name: "tests\_list", priority: 1)] public function tests\_list(EntityManagerInterface $manager, Breadcrumbs $breadcrumbs): Response{$size = $manager->getRepository(TestsOfUser::class)->count(['User' => $manager->getRepository(User::class)->findOneBy(['username' => $this->getUser()->getUserIdentifier()])->getId()]);$links = ['Главная' => "/", 'Тесты' => ""];$this->createBreadcrumb($links, $breadcrumbs);$array = ['this' => 'Тесты', 'tests' => $manager->getRepository(Tests::class)->findAll(), 'size' => $size, 'score' => $size > 0? $manager->getRepository(TestsOfUser::class)->findBy(['User' => $manager->getRepository(User::class)->findOneBy(['username' => $this->getUser()->getUserIdentifier()])->getId()]) : "",];return $this->render('tests.html.twig', $array);}#[Route('/test/{id}', name: "test")] public function showTest(EntityManagerInterface $manager, Request $request, Breadcrumbs $breadcrumbs, int $id, LoggerInterface $logger): Response{$links = ['Главная' => "/", 'Тесты' => "/test/list", 'Тест' => ""];$this->createBreadcrumb($links, $breadcrumbs);$test = $manager->getRepository(Tests::class)->find($id);$query = $request->get('res');$userAndTest = new TestsOfUser();if (isset($query)) {$res = $query;$score = 0;$allQuestions = $manager->getRepository(Questions::class)->findBy(['tests' => $id]);foreach ($res as $question => $answers) {foreach ($answers as $answer) {$true = $manager->getRepository(Answers::class)->find($answer)->isIsTrue();$score += $true? $answer == '1'? 1 : 0 : 0;}}$count = 0;foreach ($allQuestions as $question) {$allAnswers = $manager->getRepository(Answers::class)->findBy(['questions' => $question->getId()]);foreach ($allAnswers as $answer) {if ($answer->isIsTrue()) {$count++;}}}$userAndTest->setTests($test);$userAndTest->setUser($this->getUser());$userAndTest->setScore(($score / $count) \* 5);$manager->persist($userAndTest);$manager->flush();return $this->redirectToRoute('tests\_list');}$array = ['this' => 'Тест', 'test' => $test,];return $this->render('test.html.twig', $array);}#[Route('/about', name: "about")] public function about(Breadcrumbs $breadcrumbs): Response{$links = ['Главная' => "/", 'О нас' => ""];$this->createBreadcrumb($links, $breadcrumbs);$array = ['this' => 'О нас',];return $this->render('about.html.twig', $array);}#[Route('/userCab', name: "cab")] public function userCab(IdentifyService $service, Breadcrumbs $breadcrumbs): Response{$links = ['Главная' => "/", 'Личный кабинет' => ""];$this->createBreadcrumb($links, $breadcrumbs);$all = $this->manager->getRepository(Tests::class)->findAll();$count = 0;foreach ($all as $item) {$count++;}$array = ['username' => $this->getUser()->getUserIdentifier(), 'role' => $service->identifyRole(), 'tests' => $this->manager->getRepository(TestsOfUser::class)->count(['User' => $this->getUser()]) . '/' .$count, $this->manager->getRepository(Tests::class)->findAll(), 'this' => 'Личный кабинет',];return $this->render('userCab.html.twig', $array);}#[Route('/reg', name: "reg")] public function reg(Request $request, EntityManagerInterface $entityManager, UserPasswordHasherInterface $passwordHasher, UserAuthenticatorInterface $authenticator,): Response{$user = new User();$form = $this->createForm(RegistrationFormType::class, $user);$form->handleRequest($request);if ($form->isSubmitted() && $form->isValid()) {$user->setUsername($form->get('username')->getData());$user->setPassword($passwordHasher->hashPassword($user, $form->get('password')->getData()));$user->setRoles(['ROLE\_USER']);$entityManager->persist($user);$entityManager->flush();$rememberMe = new RememberMeBadge();$rememberMe->enable();$authenticator->authenticateUser($user, $this->authenticator, $request, [$rememberMe]);return $this->redirectToRoute('mainPage');}$array = ['form' => $form->createView(), 'title' => 'Зарегистрируйтесь', 'text' => 'Уже есть аккаунт? Войдите!', 'toWhere' => 'login'];return $this->render('auth.html.twig', $array);}#[Route(path: '/login', name: 'login', methods: ['GET', 'POST'])] public function login(Request $request,): Response{$user = new User();$form = $this->createForm(RegistrationFormType::class, $user);$array = ['form' => $form->createView(), 'title' => 'Войти', 'text' => 'Нет аккаунта? Зарегистрируйтесь!', 'toWhere' => 'reg', 'method' => 'POST'];return $this->render('auth.html.twig', $array);}public function checkCaptcha($logger,$form) {}#[Route('/logout', name: "logout", methods: ['GET'])] public function logout(): int{}}

src/entity/Answers.php

<?php namespace App\Entity;use App\Repository\AnswersRepository;use Doctrine\DBAL\Types\Types;use Doctrine\ORM\Mapping as ORM;#[*ORM*\Entity(repositoryClass: AnswersRepository::class)] class Answers{#[*ORM*\Id] #[*ORM*\GeneratedValue] #[*ORM*\Column] private ?int $id = null;#[*ORM*\Column(type: Types::*TEXT*)] private ?string $answer = null;#[*ORM*\Column] private ?bool $isTrue = null;#[*ORM*\ManyToOne(inversedBy: 'answers')] #[*ORM*\JoinColumn(nullable: false)] private ?Questions $questions = null;public function getId(): ?int{return $this->id;}public function getAnswer(): ?string{return $this->answer;}public function setAnswer(string $answer): self{$this->answer = $answer;return $this;}public function isIsTrue(): ?bool{return $this->isTrue;}public function setIsTrue(bool $isTrue): self{$this->isTrue = $isTrue;return $this;}public function getQuestions(): ?Questions{return $this->questions;}public function setQuestions(?Questions $questions): self{$this->questions = $questions;return $this;}}

src/entity/Lectures.php

<?php namespace App\Entity;use App\Repository\LecturesRepository;use Doctrine\DBAL\Types\Types;use Doctrine\ORM\Mapping as ORM;use Symfony\Bridge\Doctrine\Validator\Constraints\UniqueEntity;#[*ORM*\Entity(repositoryClass: LecturesRepository::class)] #[UniqueEntity(fields: ['name'], message: 'Такая лекция уже существует')] class Lectures{#[*ORM*\Id] #[*ORM*\GeneratedValue] #[*ORM*\Column] private ?int $id = null;#[*ORM*\Column(type: Types::*TEXT*, nullable: true)] private ?string $content = null;#[*ORM*\Column(length: 255)] private ?string $name = null;public function getId(): ?int{return $this->id;}public function getContent(): ?string{return $this->content;}public function setContent(?string $content): self{$this->content = $content;return $this;}public function getName(): ?string{return $this->name;}public function setName(string $name): self{$this->name = $name;return $this;}}

src/entity/Questions.php

<?php  
  
namespace App\Entity;  
  
use App\Repository\QuestionsRepository;  
use Doctrine\Common\Collections\ArrayCollection;  
use Doctrine\Common\Collections\Collection;  
use Doctrine\DBAL\Types\Types;  
use Doctrine\ORM\Mapping as ORM;  
  
#[*ORM*\Entity(repositoryClass: QuestionsRepository::class)]  
class Questions  
{  
 #[*ORM*\Id]  
 #[*ORM*\GeneratedValue]  
 #[*ORM*\Column]  
 private ?int $id = null;  
  
 #[*ORM*\Column(type: Types::*TEXT*)]  
 private ?string $question = null;  
  
 #[*ORM*\OneToMany(mappedBy: 'questions', targetEntity: Answers::class, orphanRemoval: true, cascade: ['persist'])]  
 private Collection $answers;  
  
 #[*ORM*\ManyToOne(inversedBy: 'questions')]  
 #[*ORM*\JoinColumn(nullable: false)]  
 private ?Tests $tests = null;  
  
 public function \_\_construct()  
 {  
 $this->answers = new ArrayCollection();  
 }  
  
 public function getId(): ?int  
 {  
 return $this->id;  
 }  
  
 public function getQuestion(): ?string  
 {  
 return $this->question;  
 }  
  
 public function setQuestion(string $question): self  
 {  
 $this->question = $question;  
  
 return $this;  
 }  
  
 */\*\*  
 \** ***@return*** *Collection<int, Answers>  
 \*/* public function getAnswers(): Collection  
 {  
 return $this->answers;  
 }  
  
 public function addAnswer(Answers $answer): self  
 {  
 if (!$this->answers->contains($answer)) {  
 $this->answers->add($answer);  
 $answer->setQuestions($this);  
 }  
  
 return $this;  
 }  
  
 public function removeAnswer(Answers $answer): self  
 {  
 if ($this->answers->removeElement($answer)) {  
 // set the owning side to null (unless already changed)  
 if ($answer->getQuestions() === $this) {  
 $answer->setQuestions(null);  
 }  
 }  
  
 return $this;  
 }  
  
 public function getTests(): ?Tests  
 {  
 return $this->tests;  
 }  
  
 public function setTests(?Tests $tests): self  
 {  
 $this->tests = $tests;  
  
 return $this;  
 }  
  
  
}

src/entity/Tests.php

<?php  
  
namespace App\Entity;  
  
use App\Repository\TestsRepository;  
use Doctrine\Common\Collections\ArrayCollection;  
use Doctrine\Common\Collections\Collection;  
use Doctrine\ORM\Mapping as ORM;  
  
#[ORM\Entity(repositoryClass: TestsRepository::class)]  
class Tests  
{  
 #[ORM\Id]  
 #[ORM\GeneratedValue]  
 #[ORM\Column]  
 private ?int $id = null;  
  
 #[ORM\Column(length: 255)]  
 private ?string $name = null;  
  
 #[ORM\OneToMany(mappedBy: 'tests', targetEntity: TestsOfUser::class, orphanRemoval: true, cascade: ['persist'])]  
 private Collection $testsOfUsers;  
  
 #[ORM\OneToMany(mappedBy: 'tests', targetEntity: Questions::class, orphanRemoval: true, cascade: ['persist'])]  
 private Collection $questions;  
  
 public function \_\_construct()  
 {  
 $this->testsOfUsers = new ArrayCollection();  
 $this->questions = new ArrayCollection();  
 }  
  
 public function getId(): ?int  
 {  
 return $this->id;  
 }  
  
 public function getName(): ?string  
 {  
 return $this->name;  
 }  
  
 public function setName(string $name): self  
 {  
 $this->name = $name;  
  
 return $this;  
 }  
 */\*\*  
 \** ***@return*** *Collection<int, TestsOfUser>  
 \*/* public function getTestsOfUsers(): Collection  
 {  
 return $this->testsOfUsers;  
 }  
  
 public function addTestsOfUser(TestsOfUser $testsOfUser): self  
 {  
 if (!$this->testsOfUsers->contains($testsOfUser)) {  
 $this->testsOfUsers->add($testsOfUser);  
 $testsOfUser->setTests($this);  
 }  
  
 return $this;  
 }  
  
 public function removeTestsOfUser(TestsOfUser $testsOfUser): self  
 {  
 if ($this->testsOfUsers->removeElement($testsOfUser)) {  
 // set the owning side to null (unless already changed)  
 if ($testsOfUser->getTests() === $this) {  
 $testsOfUser->setTests(null);  
 }  
 }  
  
 return $this;  
 }  
  
 */\*\*  
 \** ***@return*** *Collection<int, Questions>  
 \*/* public function getQuestions(): Collection  
 {  
 return $this->questions;  
 }  
  
 public function addQuestion(Questions $question): self  
 {  
 if (!$this->questions->contains($question)) {  
 $this->questions->add($question);  
 $question->setTests($this);  
 }  
  
 return $this;  
 }  
  
 public function removeQuestion(Questions $question): self  
 {  
 if ($this->questions->removeElement($question)) {  
 // set the owning side to null (unless already changed)  
 if ($question->getTests() === $this) {  
 $question->setTests(null);  
 }  
 }  
  
 return $this;  
 }  
}

src/entity/TestsOfUser.php

<?php  
  
namespace App\Entity;  
  
use App\Repository\TestsOfUserRepository;  
use Doctrine\ORM\Mapping as ORM;  
  
#[*ORM*\Entity(repositoryClass: TestsOfUserRepository::class)]  
class TestsOfUser  
{  
 #[*ORM*\Id]  
 #[*ORM*\GeneratedValue]  
 #[*ORM*\Column]  
 private ?int $id = null;  
  
 #[*ORM*\ManyToOne(inversedBy: 'testsOfUsers')]  
 #[*ORM*\JoinColumn(nullable: false)]  
 private ?User $User = null;  
  
 #[*ORM*\ManyToOne(inversedBy: 'testsOfUsers')]  
 #[*ORM*\JoinColumn(nullable: false)]  
 private ?Tests $tests = null;  
  
 #[*ORM*\Column]  
 private ?int $score = null;  
  
 public function getId(): ?int  
 {  
 return $this->id;  
 }  
  
 public function getUser(): ?User  
 {  
 return $this->User;  
 }  
  
 public function setUser(?User $User): self  
 {  
 $this->User = $User;  
  
 return $this;  
 }  
  
 public function getTests(): ?Tests  
 {  
 return $this->tests;  
 }  
  
 public function setTests(?Tests $tests): self  
 {  
 $this->tests = $tests;  
  
 return $this;  
 }  
  
 public function getScore(): ?int  
 {  
 return $this->score;  
 }  
  
 public function setScore(int $score): self  
 {  
 $this->score = $score;  
  
 return $this;  
 }  
}

src/entity/User.php

<?php  
  
namespace App\Entity;  
  
use App\Repository\UserRepository;  
use Doctrine\ORM\Mapping as ORM;  
use Doctrine\Common\Collections\ArrayCollection;  
use Doctrine\Common\Collections\Collection;  
use Symfony\Bridge\Doctrine\Validator\Constraints\UniqueEntity;  
use Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface;  
use Symfony\Component\Security\Core\User\UserInterface;  
use Symfony\Component\Validator\Constraints as Assert;  
  
#[ORM\Entity(repositoryClass: UserRepository::class)]  
#[UniqueEntity(fields: ['username'], message: 'Аккаунт с таким именем уже существует')]  
class User implements UserInterface, PasswordAuthenticatedUserInterface  
{  
 #[ORM\Id]  
 #[ORM\GeneratedValue]  
 #[ORM\Column]  
 private ?int $id = null;  
  
 #[ORM\Column(length: 180, unique: true)]  
 private ?string $username = null;  
  
 #[ORM\Column]  
 private array $roles = [];  
  
 */\*\*  
 \** ***@var*** *string The hashed password  
 \*/* #[ORM\Column]  
 private ?string $password = null;  
  
 #[ORM\OneToMany(mappedBy: 'User', targetEntity: TestsOfUser::class, orphanRemoval: true, cascade: ['persist'])]  
 private Collection $testsOfUsers;  
  
 public function \_\_construct()  
 {  
 $this->testsOfUsers = new ArrayCollection();  
 }  
  
 public function getId(): ?int  
 {  
 return $this->id;  
 }  
  
 public function getUsername(): ?string  
 {  
 return $this->username;  
 }  
  
 public function setUsername(string $username): self  
 {  
 $this->username = $username;  
  
 return $this;  
 }  
  
 */\*\*  
 \* A visual identifier that represents this user.  
 \*  
 \** ***@see*** *UserInterface  
 \*/* public function getUserIdentifier(): string  
 {  
 return (string) $this->username;  
 }  
  
 public function getRoles(): array  
 {  
 return $this->roles;  
 }  
  
 public function setRoles(array $roles): self  
 {  
 $this->roles = $roles;  
  
 return $this;  
 }  
  
 public function getPassword(): ?string  
 {  
 return $this->password;  
 }  
  
 public function setPassword(string $password): self  
 {  
 $this->password = $password;  
  
 return $this;  
 }  
  
 */\*\*  
 \** ***@see*** *UserInterface  
 \*/* public function eraseCredentials()  
 {  
 // If you store any temporary, sensitive data on the user, clear it here  
 // $this->plainPassword = null;  
 }  
 */\*\*  
 \** ***@return*** *Collection<int, TestsOfUser>  
 \*/* public function getTestsOfUsers(): Collection  
 {  
 return $this->testsOfUsers;  
 }  
  
 public function addTestsOfUser(TestsOfUser $testsOfUser): self  
 {  
 if (!$this->testsOfUsers->contains($testsOfUser)) {  
 $this->testsOfUsers->add($testsOfUser);  
 $testsOfUser->setUser($this);  
 }  
  
 return $this;  
 }  
  
 public function removeTestsOfUser(TestsOfUser $testsOfUser): self  
 {  
 if ($this->testsOfUsers->removeElement($testsOfUser)) {  
 // set the owning side to null (unless already changed)  
 if ($testsOfUser->getUser() === $this) {  
 $testsOfUser->setUser(null);  
 }  
 }  
  
 return $this;  
 }  
}

src/form/AnswersFormType.php

<?php  
  
namespace App\Form;  
  
use App\Entity\Answers;  
use Symfony\Component\Form\AbstractType;  
use Symfony\Component\Form\Extension\Core\Type\CheckboxType;  
use Symfony\Component\Form\Extension\Core\Type\TextType;  
use Symfony\Component\Form\FormBuilderInterface;  
use Symfony\Component\OptionsResolver\OptionsResolver;  
  
class AnswersFormType extends AbstractType  
{  
 public function buildForm(FormBuilderInterface $builder, array $options): void  
 {  
 $builder  
 ->add('answer', TextType::class, [  
 'label' => 'Ответ',  
 'attr' => [  
 'placeholder' => 'Ответ'  
 ],  
 'row\_attr' => [  
 'class' => 'form-floating',  
 ],  
 ])  
 ->add('isTrue', CheckboxType::class, [  
 'label' => 'Правильный',  
 'empty\_data' => false  
 ]);  
 }  
  
 public function configureOptions(OptionsResolver $resolver): void  
 {  
 $resolver->setDefaults([  
 'data\_class' => Answers::class,  
 ]);  
 }  
}

src/form/LectureFormType.php

<?php  
  
namespace App\Form;  
  
use App\Entity\Lectures;  
use Symfony\Component\Form\AbstractType;  
use Symfony\Component\Form\Extension\Core\Type\SubmitType;  
use Symfony\Component\Form\Extension\Core\Type\TextareaType;  
use Symfony\Component\Form\Extension\Core\Type\TextType;  
use Symfony\Component\Form\FormBuilderInterface;  
use Symfony\Component\OptionsResolver\OptionsResolver;  
  
class LectureFormType extends AbstractType  
{  
 public function buildForm(FormBuilderInterface $builder, array $options): void  
 {  
 $builder  
 ->add('name', TextType::class, [  
 'label' => 'Назовите лекцию',  
 'attr' => [  
 'placeholder' => 'Название лекции',  
 ],  
 'row\_attr' => [  
 'class' => 'form-floating',  
 ],  
 ])  
 ->add('content', TextareaType::class, [  
 'label' => 'Данные',  
 'attr' => [  
 'placeholder' => 'Данные'  
 ],  
 'row\_attr' => [  
 'class' => 'form-floating',  
 ],  
 ])  
 ->add('save', SubmitType::class, [  
 'label' => 'Сохранить'  
 ]);  
 }  
  
 public function configureOptions(OptionsResolver $resolver): void  
 {  
 $resolver->setDefaults([  
 'data\_class' => Lectures::class,  
 'csrf\_protection' => true,  
 'csrf\_field\_name' => '\_token',  
 'csrf\_token\_id' => 'this-Time\_LectUre',  
 ]);  
 }  
}

src/form/QuestionsFormType.php

<?php  
  
namespace App\Form;  
  
use App\Entity\Questions;  
use Symfony\Component\Form\AbstractType;  
use Symfony\Component\Form\Extension\Core\Type\ButtonType;  
use Symfony\Component\Form\Extension\Core\Type\CollectionType;  
use Symfony\Component\Form\Extension\Core\Type\TextType;  
use Symfony\Component\Form\FormBuilderInterface;  
use Symfony\Component\OptionsResolver\OptionsResolver;  
  
class QuestionsFormType extends AbstractType  
{  
 public function buildForm(FormBuilderInterface $builder, array $options): void  
 {  
 $builder  
 ->add('question', TextType::class, [  
 'label' => 'Вопрос',  
 'attr' => [  
 'placeholder' => 'Вопрос'  
 ],  
 'row\_attr' => [  
 'class' => 'form-floating',  
 ],  
 ])  
 ->add('answers', CollectionType::class, [  
 'entry\_type' => AnswersFormType::class,  
 'allow\_add' => true,  
 'allow\_delete' => true,  
 'by\_reference' => false,  
 'attr' => [  
 'class' => 'answers'  
 ]  
 ])  
 ->add('addNewAnswer', ButtonType::class, [  
 'attr' => [  
 'class' => 'add\_item\_link\_answer btn-secondary',  
 'mapped' => 'false',  
 ]  
 ]);  
 }  
  
 public function configureOptions(OptionsResolver $resolver): void  
 {  
 $resolver->setDefaults([  
 'data\_class' => Questions::class,  
 'csrf\_protection' => true,  
 'csrf\_field\_name' => '\_token',  
 'csrf\_token\_id' => 'THEREQuestiON',  
 ]);  
 }  
}

src/form/RegistrationFormType.php

<?php  
  
namespace App\Form;  
  
use App\Entity\User;  
use Symfony\Component\Form\AbstractType;  
use Symfony\Component\Form\Extension\Core\Type\PasswordType;  
use Symfony\Component\Form\Extension\Core\Type\SubmitType;  
use Symfony\Component\Form\Extension\Core\Type\TextType;  
use Symfony\Component\Form\FormBuilderInterface;  
use Symfony\Component\OptionsResolver\OptionsResolver;  
use VictorPrdh\RecaptchaBundle\Form\ReCaptchaType;  
  
class RegistrationFormType extends AbstractType  
{  
 public function buildForm(FormBuilderInterface $builder, array $options): void  
 {  
 $builder  
 ->add('username', TextType::class, [  
 'label' => 'Имя пользователя',  
 'attr' => [  
 'placeholder' => 'Имя пользователя'  
 ],  
 'row\_attr' => [  
 'class' => 'form-floating',  
 ],  
 ])  
 ->add('password', PasswordType::class, [  
 'label' => 'Пароль',  
 'attr' => ['autocomplete' => 'new-password', 'placeholder' => 'Пароль',],  
 'row\_attr' => [  
 'class' => 'form-floating',  
 ],  
 ])  
 ->add('captcha', ReCaptchaType::class, [])  
 ->add('save', SubmitType::class, [  
 'label' => 'Сохранить'  
 ]);  
 }  
  
 public function configureOptions(OptionsResolver $resolver): void  
 {  
 $resolver->setDefaults([  
 'data\_class' => User::class,  
 'csrf\_protection' => true,  
 'csrf\_field\_name' => '\_token',  
 'csrf\_token\_id' => 'just\_LogIn',  
 ]);  
 }  
}

src/form/TestFormType.php

<?php  
  
namespace App\Form;  
  
use App\Entity\Tests;  
use Symfony\Component\Form\AbstractType;  
use Symfony\Component\Form\Extension\Core\Type\CollectionType;  
use Symfony\Component\Form\Extension\Core\Type\SubmitType;  
use Symfony\Component\Form\Extension\Core\Type\TextType;  
use Symfony\Component\Form\FormBuilderInterface;  
use Symfony\Component\OptionsResolver\OptionsResolver;  
  
class TestFormType extends AbstractType  
{  
 public function buildForm(FormBuilderInterface $builder, array $options): void  
 {  
 $builder  
 ->add('name', TextType::class, [  
 'label' => 'Название теста',  
 'attr' => [  
 'placeholder' => 'Название теста'  
 ],  
 'row\_attr' => [  
 'class' => 'form-floating',  
 ],  
 ])  
 ->add('questions', CollectionType::class, [  
 'entry\_type' => QuestionsFormType::class,  
 'entry\_options' => ['label' => false],  
 'allow\_add' => true,  
 'allow\_delete' => true,  
 'by\_reference' => false,  
 'attr' => [  
 'class' => 'questions',  
 'onload' => 'addFormToCollection()'  
 ]  
 ])  
 ->add('save', SubmitType::class, [  
 'label' => 'Сохранить'  
 ]);  
 }  
  
 public function configureOptions(OptionsResolver $resolver): void  
 {  
 $resolver->setDefaults([  
 'data\_class' => Tests::class,  
 'csrf\_protection' => false,  
 'csrf\_field\_name' => '\_token',  
 'csrf\_token\_id' => 'AND-there-test',  
 ]);  
 }  
}

src/repository/AnswersRepository.php

<?php  
  
namespace App\Repository;  
  
use App\Entity\Answers;  
use Doctrine\Bundle\DoctrineBundle\Repository\ServiceEntityRepository;  
use Doctrine\Persistence\ManagerRegistry;  
  
*/\*\*  
 \** ***@extends*** *ServiceEntityRepository<Answers>  
 \*  
 \** ***@method*** *Answers|null find($id, $lockMode = null, $lockVersion = null)  
 \** ***@method*** *Answers|null findOneBy(array $criteria, array $orderBy = null)  
 \** ***@method*** *Answers[] findAll()  
 \** ***@method*** *Answers[] findBy(array $criteria, array $orderBy = null, $limit = null, $offset = null)  
 \*/*class AnswersRepository extends ServiceEntityRepository  
{  
 public function \_\_construct(ManagerRegistry $registry)  
 {  
 parent::*\_\_construct*($registry, Answers::class);  
 }  
  
 public function save(Answers $entity, bool $flush = false): void  
 {  
 $this->getEntityManager()->persist($entity);  
  
 if ($flush) {  
 $this->getEntityManager()->flush();  
 }  
 }  
  
 public function remove(Answers $entity, bool $flush = false): void  
 {  
 $this->getEntityManager()->remove($entity);  
  
 if ($flush) {  
 $this->getEntityManager()->flush();  
 }  
 }  
  
// /\*\*  
// \* @return Answers[] Returns an array of Answers objects  
// \*/  
// public function findByExampleField($value): array  
// {  
// return $this->createQueryBuilder('a')  
// ->andWhere('a.exampleField = :val')  
// ->setParameter('val', $value)  
// ->orderBy('a.id', 'ASC')  
// ->setMaxResults(10)  
// ->getQuery()  
// ->getResult()  
// ;  
// }  
  
// public function findOneBySomeField($value): ?Answers  
// {  
// return $this->createQueryBuilder('a')  
// ->andWhere('a.exampleField = :val')  
// ->setParameter('val', $value)  
// ->getQuery()  
// ->getOneOrNullResult()  
// ;  
// }  
}

src/repository/LecturesRepository.php

<?php  
  
namespace App\Repository;  
  
use App\Entity\Lectures;  
use Doctrine\Bundle\DoctrineBundle\Repository\ServiceEntityRepository;  
use Doctrine\Persistence\ManagerRegistry;  
  
*/\*\*  
 \** ***@extends*** *ServiceEntityRepository<Lectures>  
 \*  
 \** ***@method*** *Lectures|null find($id, $lockMode = null, $lockVersion = null)  
 \** ***@method*** *Lectures|null findOneBy(array $criteria, array $orderBy = null)  
 \** ***@method*** *Lectures[] findAll()  
 \** ***@method*** *Lectures[] findBy(array $criteria, array $orderBy = null, $limit = null, $offset = null)  
 \*/*class LecturesRepository extends ServiceEntityRepository  
{  
 public function \_\_construct(ManagerRegistry $registry)  
 {  
 parent::*\_\_construct*($registry, Lectures::class);  
 }  
  
 public function save(Lectures $entity, bool $flush = false): void  
 {  
 $this->getEntityManager()->persist($entity);  
  
 if ($flush) {  
 $this->getEntityManager()->flush();  
 }  
 }  
  
 public function remove(Lectures $entity, bool $flush = false): void  
 {  
 $this->getEntityManager()->remove($entity);  
  
 if ($flush) {  
 $this->getEntityManager()->flush();  
 }  
 }  
  
// /\*\*  
// \* @return Lectures[] Returns an array of Lectures objects  
// \*/  
// public function findByExampleField($value): array  
// {  
// return $this->createQueryBuilder('l')  
// ->andWhere('l.exampleField = :val')  
// ->setParameter('val', $value)  
// ->orderBy('l.id', 'ASC')  
// ->setMaxResults(10)  
// ->getQuery()  
// ->getResult()  
// ;  
// }  
  
// public function findOneBySomeField($value): ?Lectures  
// {  
// return $this->createQueryBuilder('l')  
// ->andWhere('l.exampleField = :val')  
// ->setParameter('val', $value)  
// ->getQuery()  
// ->getOneOrNullResult()  
// ;  
// }  
}

src/repository/QuestionsRepository.php

<?php  
  
namespace App\Repository;  
  
use App\Entity\Questions;  
use Doctrine\Bundle\DoctrineBundle\Repository\ServiceEntityRepository;  
use Doctrine\Persistence\ManagerRegistry;  
  
*/\*\*  
 \** ***@extends*** *ServiceEntityRepository<Questions>  
 \*  
 \** ***@method*** *Questions|null find($id, $lockMode = null, $lockVersion = null)  
 \** ***@method*** *Questions|null findOneBy(array $criteria, array $orderBy = null)  
 \** ***@method*** *Questions[] findAll()  
 \** ***@method*** *Questions[] findBy(array $criteria, array $orderBy = null, $limit = null, $offset = null)  
 \*/*class QuestionsRepository extends ServiceEntityRepository  
{  
 public function \_\_construct(ManagerRegistry $registry)  
 {  
 parent::*\_\_construct*($registry, Questions::class);  
 }  
  
 public function save(Questions $entity, bool $flush = false): void  
 {  
 $this->getEntityManager()->persist($entity);  
  
 if ($flush) {  
 $this->getEntityManager()->flush();  
 }  
 }  
  
 public function remove(Questions $entity, bool $flush = false): void  
 {  
 $this->getEntityManager()->remove($entity);  
  
 if ($flush) {  
 $this->getEntityManager()->flush();  
 }  
 }  
  
// /\*\*  
// \* @return Questions[] Returns an array of Questions objects  
// \*/  
// public function findByExampleField($value): array  
// {  
// return $this->createQueryBuilder('q')  
// ->andWhere('q.exampleField = :val')  
// ->setParameter('val', $value)  
// ->orderBy('q.id', 'ASC')  
// ->setMaxResults(10)  
// ->getQuery()  
// ->getResult()  
// ;  
// }  
  
// public function findOneBySomeField($value): ?Questions  
// {  
// return $this->createQueryBuilder('q')  
// ->andWhere('q.exampleField = :val')  
// ->setParameter('val', $value)  
// ->getQuery()  
// ->getOneOrNullResult()  
// ;  
// }  
}

src/repository/TestsOfUserRepository.php

<?php  
  
namespace App\Repository;  
  
use App\Entity\TestsOfUser;  
use Doctrine\Bundle\DoctrineBundle\Repository\ServiceEntityRepository;  
use Doctrine\Persistence\ManagerRegistry;  
  
*/\*\*  
 \** ***@extends*** *ServiceEntityRepository<TestsOfUser>  
 \*  
 \** ***@method*** *TestsOfUser|null find($id, $lockMode = null, $lockVersion = null)  
 \** ***@method*** *TestsOfUser|null findOneBy(array $criteria, array $orderBy = null)  
 \** ***@method*** *TestsOfUser[] findAll()  
 \** ***@method*** *TestsOfUser[] findBy(array $criteria, array $orderBy = null, $limit = null, $offset = null)  
 \*/*class TestsOfUserRepository extends ServiceEntityRepository  
{  
 public function \_\_construct(ManagerRegistry $registry)  
 {  
 parent::*\_\_construct*($registry, TestsOfUser::class);  
 }  
  
 public function save(TestsOfUser $entity, bool $flush = false): void  
 {  
 $this->getEntityManager()->persist($entity);  
  
 if ($flush) {  
 $this->getEntityManager()->flush();  
 }  
 }  
  
 public function remove(TestsOfUser $entity, bool $flush = false): void  
 {  
 $this->getEntityManager()->remove($entity);  
  
 if ($flush) {  
 $this->getEntityManager()->flush();  
 }  
 }  
  
// /\*\*  
// \* @return TestsOfUser[] Returns an array of TestsOfUser objects  
// \*/  
// public function findByExampleField($value): array  
// {  
// return $this->createQueryBuilder('t')  
// ->andWhere('t.exampleField = :val')  
// ->setParameter('val', $value)  
// ->orderBy('t.id', 'ASC')  
// ->setMaxResults(10)  
// ->getQuery()  
// ->getResult()  
// ;  
// }  
  
// public function findOneBySomeField($value): ?TestsOfUser  
// {  
// return $this->createQueryBuilder('t')  
// ->andWhere('t.exampleField = :val')  
// ->setParameter('val', $value)  
// ->getQuery()  
// ->getOneOrNullResult()  
// ;  
// }  
}

src/repository/TestsRepository.php

<?php  
  
namespace App\Repository;  
  
use App\Entity\Tests;  
use Doctrine\Bundle\DoctrineBundle\Repository\ServiceEntityRepository;  
use Doctrine\Persistence\ManagerRegistry;  
  
*/\*\*  
 \** ***@extends*** *ServiceEntityRepository<Tests>  
 \*  
 \* @method Tests|null find($id, $lockMode = null, $lockVersion = null)  
 \* @method Tests|null findOneBy(array $criteria, array $orderBy = null)  
 \* @method Tests[] findAll()  
 \* @method Tests[] findBy(array $criteria, array $orderBy = null, $limit = null, $offset = null)  
 \*/*class TestsRepository extends ServiceEntityRepository  
{  
 public function \_\_construct(ManagerRegistry $registry)  
 {  
 parent::\_\_construct($registry, Tests::class);  
 }  
  
 public function save(Tests $entity, bool $flush = false): void  
 {  
 $this->getEntityManager()->persist($entity);  
  
 if ($flush) {  
 $this->getEntityManager()->flush();  
 }  
 }  
  
 public function remove(Tests $entity, bool $flush = false): void  
 {  
 $this->getEntityManager()->remove($entity);  
  
 if ($flush) {  
 $this->getEntityManager()->flush();  
 }  
 }  
  
// /\*\*  
// \* @return Tests[] Returns an array of Tests objects  
// \*/  
// public function findByExampleField($value): array  
// {  
// return $this->createQueryBuilder('t')  
// ->andWhere('t.exampleField = :val')  
// ->setParameter('val', $value)  
// ->orderBy('t.id', 'ASC')  
// ->setMaxResults(10)  
// ->getQuery()  
// ->getResult()  
// ;  
// }  
  
// public function findOneBySomeField($value): ?Tests  
// {  
// return $this->createQueryBuilder('t')  
// ->andWhere('t.exampleField = :val')  
// ->setParameter('val', $value)  
// ->getQuery()  
// ->getOneOrNullResult()  
// ;  
// }  
}

src/repository/UserRepository.php

<?php  
  
namespace App\Repository;  
  
use App\Entity\User;  
use Doctrine\Bundle\DoctrineBundle\Repository\ServiceEntityRepository;  
use Doctrine\Persistence\ManagerRegistry;  
use Symfony\Component\Security\Core\Exception\UnsupportedUserException;  
use Symfony\Component\Security\Core\User\PasswordAuthenticatedUserInterface;  
use Symfony\Component\Security\Core\User\PasswordUpgraderInterface;  
  
*/\*\*  
 \** ***@extends*** *ServiceEntityRepository<User>  
 \*  
 \* @method User|null find($id, $lockMode = null, $lockVersion = null)  
 \* @method User|null findOneBy(array $criteria, array $orderBy = null)  
 \* @method User[] findAll()  
 \* @method User[] findBy(array $criteria, array $orderBy = null, $limit = null, $offset = null)  
 \*/*class UserRepository extends ServiceEntityRepository implements PasswordUpgraderInterface  
{  
 public function \_\_construct(ManagerRegistry $registry)  
 {  
 parent::\_\_construct($registry, User::class);  
 }  
  
 public function save(User $entity, bool $flush = false): void  
 {  
 $this->getEntityManager()->persist($entity);  
  
 if ($flush) {  
 $this->getEntityManager()->flush();  
 }  
 }  
  
 public function remove(User $entity, bool $flush = false): void  
 {  
 $this->getEntityManager()->remove($entity);  
  
 if ($flush) {  
 $this->getEntityManager()->flush();  
 }  
 }  
  
 */\*\*  
 \* Used to upgrade (rehash) the user's password automatically over time.  
 \*/* public function upgradePassword(PasswordAuthenticatedUserInterface $user, string $newHashedPassword): void  
 {  
 if (!$user instanceof User) {  
 throw new UnsupportedUserException(sprintf('Instances of "%s" are not supported.', \get\_class($user)));  
 }  
  
 $user->setPassword($newHashedPassword);  
  
 $this->save($user, true);  
 }  
  
// /\*\*  
// \* @return User[] Returns an array of User objects  
// \*/  
// public function findByExampleField($value): array  
// {  
// return $this->createQueryBuilder('u')  
// ->andWhere('u.exampleField = :val')  
// ->setParameter('val', $value)  
// ->orderBy('u.id', 'ASC')  
// ->setMaxResults(10)  
// ->getQuery()  
// ->getResult()  
// ;  
// }  
  
// public function findOneBySomeField($value): ?User  
// {  
// return $this->createQueryBuilder('u')  
// ->andWhere('u.exampleField = :val')  
// ->setParameter('val', $value)  
// ->getQuery()  
// ->getOneOrNullResult()  
// ;  
// }  
}

src/service/IdentifyService.php

<?php  
  
namespace App\Service;  
  
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;  
  
class IdentifyService extends AbstractController  
{  
 public function identifyRole() {  
 switch ($this->getUser()->getRoles()[0]) {  
 case "ROLE\_USER": {  
 return 'Пользователь';  
 }  
 case "ROLE\_PREMIUM": {  
 return 'Поддержал рублём';  
 }  
 case "ROLE\_EDITOR": {  
 return 'Редактор';  
 }  
 case "ROLE\_ADMIN": {  
 return 'Админ';  
 }  
 }  
 }  
}

src/Kernel.php

<?php  
  
namespace App;  
  
use Symfony\Bundle\FrameworkBundle\Kernel\MicroKernelTrait;  
use Symfony\Component\HttpKernel\Kernel as BaseKernel;  
  
class Kernel extends BaseKernel  
{  
 use MicroKernelTrait;  
}

templates/\_header.html.twig

{% block header %}  
 <header>  
 <a class="logo" href="{{ path('mainPage') }}">  
 <img class="logo-img img" src="{{ asset('img/logo.png') }}" alt="">  
 </a>  
 <div class="links a">  
 {% if is\_granted('IS\_AUTHENTICATED') %}  
 <a href="{{ path('cab') }}">  
 <label class="links-label" for="cab">  
 <svg viewBox="0 0 24 24" id="cab" fill="none" xmlns="http://www.w3.org/2000/svg" stroke="#ffffff">  
 <path d="M18 18.7023C18 15.6706 14.5 15 12 15C9.5 15 6 15.6706 6 18.7023M21 12C21 16.9706 16.9706 21 12 21C7.02944 21 3 16.9706 3 12C3 7.02944 7.02944 3 12 3C16.9706 3 21 7.02944 21 12ZM15 9C15 10.6569 13.6569 12 12 12C10.3431 12 9 10.6569 9 9C9 7.34315 10.3431 6 12 6C13.6569 6 15 7.34315 15 9Z"  
 stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"></path>  
 </svg>  
 Личный кабинет  
 </label>  
 </a>  
 {% endif %}  
 <a href="{{ path('about') }}">  
 <label class="links-label" for="about">  
 <svg viewBox="0 0 512 512" id="about" xmlns="http://www.w3.org/2000/svg" fill="#ffffff">  
 <path d="M213.333333,3.55271368e-14 C95.51296,3.55271368e-14 3.55271368e-14,95.51168 3.55271368e-14,213.333333 C3.55271368e-14,331.153707 95.51296,426.666667 213.333333,426.666667 C331.154987,426.666667 426.666667,331.153707 426.666667,213.333333 C426.666667,95.51168 331.154987,3.55271368e-14 213.333333,3.55271368e-14 Z M213.333333,384 C119.227947,384 42.6666667,307.43872 42.6666667,213.333333 C42.6666667,119.227947 119.227947,42.6666667 213.333333,42.6666667 C307.44,42.6666667 384,119.227947 384,213.333333 C384,307.43872 307.44,384 213.333333,384 Z M240.04672,128 C240.04672,143.46752 228.785067,154.666667 213.55008,154.666667 C197.698773,154.666667 186.713387,143.46752 186.713387,127.704107 C186.713387,112.5536 197.99616,101.333333 213.55008,101.333333 C228.785067,101.333333 240.04672,112.5536 240.04672,128 Z M192.04672,192 L234.713387,192 L234.713387,320 L192.04672,320 L192.04672,192 Z"  
 id="Shape"></path>  
 </svg>  
 О нас  
 </label>  
 </a>  
 <a href="{{ path('lectures\_list') }}">  
 <label class="links-label" for="lecture">  
 <svg viewBox="0 0 512 512" id="lecture" xmlns="http://www.w3.org/2000/svg" fill="#ffffff">  
 <polygon fill="#ffffff"  
 points="368 350.643 256 413.643 144 350.643 144 284.081 112 266.303 112 369.357 256 450.357 400 369.357 400 266.303 368 284.081 368 350.643"></polygon>  
 <path fill="#ffffff"  
 d="M256,45.977,32,162.125v27.734L256,314.3,448,207.637V296h32V162.125ZM416,188.808l-32,17.777L256,277.7,128,206.585,96,188.808,73.821,176.486,256,82.023l182.179,94.463Z"></path>  
 </svg>  
 Лекции  
 </label>  
 </a>  
 <a href="{{ path('tests\_list') }}">  
 <label class="links-label" for="test">  
 <svg fill="#ffffff" id="test" viewBox="0 0 24 24" role="img" xmlns="http://www.w3.org/2000/svg">  
 <path d="M4.35 1.66c-.959 0-1.686.298-2.181.893-.496.595-.744 1.464-.744 2.605v3.474c0 .805-.086 1.424-.259 1.858-.173.434-.493.84-.96 1.218-.138.113-.206.206-.206.278 0 .072.068.165.205.277.476.386.798.794.967 1.225.17.43.253 1.047.253 1.851v3.462c0 1.15.25 2.023.75 2.618.5.595 1.224.892 2.174.892h.882c.379 0 .618-.018.72-.054.1-.036.15-.127.15-.271 0-.201-.133-.334-.399-.398l-.483-.109c-1.063-.241-1.595-1.29-1.595-3.148v-3.293c0-1.247-.302-2.127-.906-2.642l-.23-.193c-.112-.096-.168-.169-.168-.217 0-.056.056-.129.169-.217l.23-.193c.603-.515.905-1.395.905-2.642V5.641c0-1.11.135-1.88.405-2.31.27-.43.832-.762 1.685-.995.258-.073.387-.19.387-.35 0-.217-.29-.326-.87-.326zm14.419.029c-.58 0-.87.108-.87.325 0 .161.128.278.386.35.854.233 1.416.565 1.686.995.27.43.405 1.2.405 2.31v3.294c0 1.246.302 2.126.906 2.641l.229.193c.113.089.17.161.17.217 0 .049-.057.121-.17.217l-.23.193c-.603.515-.905 1.396-.905 2.642v3.293c0 1.858-.532 2.907-1.595 3.149l-.484.108c-.266.064-.398.197-.398.398 0 .145.05.235.15.272.102.036.341.054.72.054h.882c.95 0 1.675-.298 2.174-.893.5-.595.75-1.467.75-2.617v-3.462c0-.805.084-1.422.253-1.852.17-.43.491-.838.967-1.224.137-.113.205-.205.205-.278 0-.072-.068-.165-.205-.277-.468-.378-.788-.784-.961-1.218-.173-.435-.26-1.054-.26-1.858V5.187c0-1.142-.247-2.01-.743-2.606-.495-.595-1.222-.892-2.18-.892zM7.683 9.735c-1.456 0-2.64 1.111-2.64 2.478h1.02c0-.838.727-1.52 1.62-1.52.892 0 1.619.682 1.619 1.52h1.02c0-1.367-1.183-2.478-2.64-2.478zm8.406 0c-1.456 0-2.639 1.111-2.639 2.478h1.02c0-.838.727-1.52 1.62-1.52.892 0 1.62.682 1.62 1.52h1.02c0-1.367-1.185-2.478-2.64-2.478zM9.71 14.36v.561c0 1.277 1.062 2.316 2.366 2.316 1.305 0 2.367-1.039 2.367-2.316v-.56h-.934v.56c0 .877-.76 1.426-1.433 1.426s-1.48-.273-1.48-1.426v-.56z"></path>  
 </svg>  
 Тесты  
 </label>  
 </a>  
 {% if not is\_granted('IS\_AUTHENTICATED') %}  
 <a href="{{ path('login') }}">  
 <label class="links-label" for="login">  
 <svg viewBox="0 0 24 24" id="login" xmlns="http://www.w3.org/2000/svg">  
 <path fill-rule="evenodd" clip-rule="evenodd"  
 d="M8 6C8 3.79086 9.79086 2 12 2H17.5C19.9853 2 22 4.01472 22 6.5V17.5C22 19.9853 19.9853 22 17.5 22H12C9.79086 22 8 20.2091 8 18V17C8 16.4477 8.44772 16 9 16C9.55228 16 10 16.4477 10 17V18C10 19.1046 10.8954 20 12 20H17.5C18.8807 20 20 18.8807 20 17.5V6.5C20 5.11929 18.8807 4 17.5 4H12C10.8954 4 10 4.89543 10 6V7C10 7.55228 9.55228 8 9 8C8.44772 8 8 7.55228 8 7V6ZM12.2929 8.29289C12.6834 7.90237 13.3166 7.90237 13.7071 8.29289L16.7071 11.2929C17.0976 11.6834 17.0976 12.3166 16.7071 12.7071L13.7071 15.7071C13.3166 16.0976 12.6834 16.0976 12.2929 15.7071C11.9024 15.3166 11.9024 14.6834 12.2929 14.2929L13.5858 13L5 13C4.44772 13 4 12.5523 4 12C4 11.4477 4.44772 11 5 11L13.5858 11L12.2929 9.70711C11.9024 9.31658 11.9024 8.68342 12.2929 8.29289Z"  
 fill="#ffffff"></path>  
 </svg>  
 Войти  
 </label>  
 </a>  
 {% endif %}  
 </div>  
 </header>  
 <div class="bread bottom-line montserrat">  
 {{ wo\_render\_breadcrumbs({separator: '>'}) }}  
 </div>  
{% endblock %}

templates/about.html.twig

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>{{ this }}</title>  
 {% block stylesheet %}  
 <link href="{{ asset('css/about.css') }}" rel="stylesheet"/>  
 <link href="{{ asset('css/\_header.css') }}" rel="stylesheet"/>  
 {% endblock %}  
</head>  
<body>  
<div class="wrapper montserrat">  
 {{ block('header', '\_header.html.twig') }}  
 <div class="container about text-left darkText">  
 <h1 class="about-header">  
 TechLand  
 </h1>  
 <span class="about-body">  
 <span>  
 Компания TechLand была основана в 2023 году и с того времени ведёт свою образовательная деятельность.<br><br>  
 Мы преследуем следующие цели:  
 </span>  
 <ul>  
 <li>  
 Распространить как можно большему количеству людей знания о языке программирования PHP  
 </li>  
 <li>  
 Повысить образованность программистов  
 </li>  
 </ul>  
 <br>  
 Наша компания планирует развиваться в следующих направлениях:  
 <ul>  
 <li>  
 Расшириться в большее количество языков и затронуть как можно больший объём информации  
 </li>  
 <li>  
 Предоставить массам возможность понять программирование  
 </li>  
 </ul>  
 </span>  
 <br>  
 <div class="bottom-line"></div>  
 <h1 class="about-header">  
 Контакты  
 </h1>  
 <span class="about-body">  
 Номер телефона: +79999999999<br>  
 Адрес электронной почты: techland.ru@work.com<br>  
 </span>  
 </div>  
</div>  
</body>  
</html>

templates/add-edit-lecture.html.twig

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>{{ this }}</title>  
 {% form\_theme form 'bootstrap\_5\_layout.html.twig' %}  
 {% block stylesheet %}  
 <link href="{{ asset('css/\_header.css') }}" rel="stylesheet"/>  
 <link href="{{ asset('css/add-edit-lecture-main.css') }}" rel="stylesheet"/>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">  
 {% endblock %}  
 {% block javascript %}  
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>  
 {% endblock %}  
</head>  
<body>  
<div class="wrapper montserrat">  
 {{ block('header', '\_header.html.twig') }}  
 <div class="main">  
 {{ form\_start(form) }}  
 {{ form\_row(form.name) }}  
 {{ form\_end(form) }}  
 </div>  
</div>  
</body>  
</html>

templates/add-edit-test.html.twig

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>{{ this }}</title>  
 {% block stylesheet %}  
 <link href="{{ asset('css/\_header.css') }}" rel="stylesheet"/>  
 <link href="{{ asset('css/test-main.css') }}" rel="stylesheet"/>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"  
 integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"  
 crossorigin="anonymous">  
 {% endblock %}  
 {% block javascript %}  
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"  
 integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"  
 crossorigin="anonymous"></script>  
 {% endblock %}  
</head>  
<body>  
<div class="wrapper montserrat">  
 {{ block('header', '\_header.html.twig') }}  
 <div class="main">  
 {{ form\_start(form) }}  
 {{ form\_row(form.name) }}  
 <ul class="questions"  
 data-index="{{ form.questions|length > 0 ? form.questions|last.vars.name + 1 : 0 }}"  
 data-prototype="{{ form\_widget(form.questions.vars.prototype)|e('html\_attr') }}">  
 </ul>  
 <button type="button" class="add\_item\_link btn" data-collection-holder-class="questions">Добавить вопрос</button>  
 {{ form\_row(form.save) }}  
 {{ form\_end(form, {render\_rest: false}) }}  
 <script>  
 const addQuestion = document.querySelector('.add\_item\_link')  
 addQuestion.dataset.collectionHolderClass='questions'  
  
 const newLinkLi = document.createElement('li')  
 document.getElementsByName('test\_form').item(0).append(addQuestion)  
  
 const collection = document.querySelector('.questions')  
 collection.appendChild(addQuestion)  
  
 const addQuestionToCollection = () => {  
 const collectionHolder = document.querySelector('.questions');  
  
 const item = document.createElement('li');  
  
 item.innerHTML = collectionHolder  
 .dataset  
 .prototype  
 .replace(  
 /\_\_name\_\_/g,  
 collectionHolder.dataset.index  
 );  
 collectionHolder.appendChild(item);  
 const addAnswer = item.querySelector('.add\_item\_link\_answer')  
 addAnswer.addEventListener("click", addFormToCollection)  
 addAnswer.dataset.collectionHolderClass = 'answers'  
 addAnswer.innerText = 'Добавить ответ'  
 item.appendChild(addAnswer)  
 item.querySelector('legend.required').innerText = 'Ответы'  
  
 collectionHolder.dataset.index++;  
 }  
  
 const addFormToCollection = (e) => {  
 const collectionHolder = e.target.closest('li').querySelector('.answers')  
  
 const item = document.createElement('li');  
  
 item.innerHTML = collectionHolder  
 .dataset  
 .prototype  
 .replace(  
 /\_\_name\_\_/g,  
 collectionHolder.dataset.index  
 );  
 collectionHolder.appendChild(item);  
 item.setAttribute('id', 'answer' + collectionHolder.dataset.index)  
 item.querySelector('input.form-check-input').removeAttribute('required')  
 item.querySelector('legend.required').remove()  
  
 collectionHolder.dataset.index++;  
 }  
  
 addQuestion.addEventListener("click", addQuestionToCollection)  
 </script>  
 </div>  
</div>  
</body>  
</html>

templates/admin.html.twig

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>{{ this }}</title>  
 {% block stylesheet %}  
 <link href="{{ asset('css/\_header.css') }}" rel="stylesheet"/>  
 <link href="{{ asset('css/admin.css') }}" rel="stylesheet"/>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"  
 integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"  
 crossorigin="anonymous">  
 {% endblock %}  
 {% block javascript %}  
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"  
 integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"  
 crossorigin="anonymous"></script>  
 {% endblock %}  
</head>  
<body>  
<div class="wrapper montserrat">  
 {{ block('header', '\_header.html.twig') }}  
 <form>  
 <button class="btn btn-light btn-outline-primary" type="submit">Сохранить</button>  
  
 <table class="darkText main">  
 <tr>  
 <th>Никнейм</th>  
 <th>Роль</th>  
 <th>Удалить</th>  
 </tr>  
 {% for key, user in users %}  
 <tr>  
 <td>  
 {{ user.username }}  
 </td>  
 <td>  
 {% if user.username != app.user.username %}  
 <select name="users[{{ user.username }}]">  
 <option value="{{ user.roles[0] }}">  
 {{ user.roles[0] }}  
 </option>  
 {% for key, role in roles %}  
 {% if user.roles[0] != role %}  
 <option value="{{ role }}">  
 {{ role }}  
 </option>  
 {% endif %}  
 {% endfor %}  
 </select>  
 {% else %}  
 <span>  
 {{ user.roles[0] }}  
 </span>  
 {% endif %}  
 </td>  
 <td>  
 {% if user.username != app.user.username %}  
 <form action="{{ path('deleteUser', {id: user.id}) }}">  
 <button class="button-delete" type="submit">  
 <svg viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/svg">  
 <g id="SVGRepo\_bgCarrier" stroke-width="0"></g>  
 <g id="SVGRepo\_tracerCarrier" stroke-linecap="round" stroke-linejoin="round"></g>  
 <g id="SVGRepo\_iconCarrier">  
 <path d="M19 5L5 19M5.00001 5L19 19" stroke-width="1.5"  
 stroke-linecap="round" stroke-linejoin="round"></path>  
 </g>  
 </svg>  
 </button>  
 </form>  
 {% endif %}  
 </td>  
 </tr>  
 {% endfor %}  
 </table>  
 </form>  
</div>  
</body>  
</html>

templates/auth.html.twig

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <meta http-equiv="X-UA-Compatible" content="IE=edge">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <link rel="stylesheet" href="{{ asset('css/login.css') }}">  
 {% form\_theme form 'bootstrap\_5\_layout.html.twig' %}  
 {% block stylesheet %}  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">  
 {% endblock %}  
 {% block javascript %}  
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js" integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM" crossorigin="anonymous"></script>  
 {% endblock %}  
 <title>{{ title }}</title></head>  
<body>  
<div class="logo">  
 <a href="{{ path('mainPage') }}">  
 <img src="{{ asset('img/logo.png') }}" class="logo-img img" alt="">  
 </a>  
</div>  
<div class="login">  
 <span class="title">{{ title }}</span>  
 {{ form\_start(form) }}  
 {{ form\_errors(form) }}  
 {{ form\_row(form.username) }}  
 {{ form\_row(form.password) }}  
 {% if toWhere != 'reg' %}  
 {{ form\_row(form.captcha) }}  
 {% endif %}  
 {{ form\_row(form.save) }}  
 {% if toWhere == 'reg' %}  
 <label>  
 <input type="checkbox" name="\_remember\_me" checked/>  
 Keep me logged in  
 </label>  
 {% endif %}  
 <input type="hidden" name="\_target\_path" value="{{ path('mainPage') }}">  
 {{ form\_end(form) }}  
 <div class="bottom-title text">  
 <a href="{{ path( toWhere ) }}" class="link">  
 {{ text }}  
 </a>  
 </div>  
</div>  
</body>  
</html>

templates/index.html.twig

<!DOCTYPE html>  
<html lang="">  
<head>  
 <meta charset='UTF-8'>  
 <title>{{ this }}</title>  
 <link href="{{ asset('css/\_header.css') }}" rel="stylesheet"/>  
 <link href="{{ asset('css/index-info.css') }}" rel="stylesheet"/>  
 <link href="{{ asset('css/index-courses.css') }}" rel="stylesheet"/>  
</head>  
<body>  
<div class="wrapper montserrat">  
 {{ block('header', '\_header.html.twig') }}  
 <div class="info darkText">  
 <span>  
 Наша компания TechLand занимается предоставлением обучающего материала и обучению.<br><br>  
 На нашем сайте вы всегда можете найти информацию по языку программирования PHP или же проверить свои знания на имеющихся на сайте тестах  
 </span>  
 </div>  
 <div class="courses darkText">  
 <div class="free">  
 <h1>  
 Полный доступ  
 </h1>  
 <div>  
 <span>  
 Вы получаете доступ ко всему, что есть на этом сайте:  
 </span>  
 <ul>  
 <li>  
 Лекциям;  
 </li>  
 <li>  
 Тестам.  
 </li>  
 </ul>  
 </div>  
 <span class="price">  
 <br>Цена: 0.00 руб  
 </span>  
 <button>  
 Получить  
 </button>  
 </div>  
 <div class="with-price">  
 <h1>  
 Поддержите нас рублём  
 </h1>  
 <div>  
 <span>  
 Вы сделаете нам приятно, ведь у нас на сайте нет рекламы!  
 </span>  
 <ul>  
 <li>  
 Мы ничего не можем вам предложить.  
 </li>  
 <li>  
 Но будем очень благодарны вам!  
 </li>  
 </ul>  
 </div>  
 <span class="price">  
 <br>Цена: 500.00 руб  
 </span>  
 <button>  
 Поддержать  
 </button>  
 </div>  
 </div>  
</div>  
</body>  
</html>

templates/lecture.html.twig

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>{{ this }}</title>  
 {% block stylesheet %}  
 <link href="{{ asset('css/\_header.css') }}" rel="stylesheet"/>  
 <link href="{{ asset('css/lecture-main.css') }}" rel="stylesheet"/>  
 {% endblock %}  
</head>  
<body>  
<div class="wrapper montserrat">  
 {{ block('header', '\_header.html.twig') }}  
 <div class="main">  
 <h1>  
 {{ lecture.name }}  
 </h1>  
 <span>  
 {{ lecture.content }}  
 </span>  
 </div>  
</div>  
</body>  
</html>

templates/lectures.html.twig

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>{{ this }}</title>  
 {% block stylesheet %}  
 <link href="{{ asset('css/\_header.css') }}" rel="stylesheet"/>  
 <link href="{{ asset('css/lecture\_list-main.css') }}" rel="stylesheet"/>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"  
 integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"  
 crossorigin="anonymous">  
 {% endblock %}  
 {% block javascript %}  
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"  
 integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"  
 crossorigin="anonymous"></script>  
 {% endblock %}  
</head>  
<body>  
<div class="wrapper montserrat">  
 {{ block('header', '\_header.html.twig') }}  
 <div class="main">  
 <ul>  
 {% if is\_granted('ROLE\_EDITOR') %}  
 <li class="list-group-item">  
 <a href="{{ path('addLecture') }}">Сделать лекцию</a>  
 </li>  
 {% endif %}  
 {% for key, lecture in lectures %}  
 <li class="lecture border list-group-item">  
 <a href="{{ path('lecture', {id: lecture.id}) }}">  
 {{ lecture.name }}  
 </a>  
 <div class="actions float-end ">  
 {% if is\_granted('ROLE\_EDITOR') %}  
 <form action="{{ path('editLecture', {id: lecture.id}) }}">  
 <button class="button-edit" type="submit">  
 <svg viewBox="0 0 24 24" fill="none" xmlns="http://www.w3.org/2000/svg">  
 <g id="SVGRepo\_bgCarrier" stroke-width="0"></g>  
 <g id="SVGRepo\_tracerCarrier" stroke-linecap="round" stroke-linejoin="round"></g>  
 <g id="SVGRepo\_iconCarrier">  
 <path d="M18 10L21 7L17 3L14 6M18 10L8 20H4V16L14 6M18 10L14 6" stroke="#ffffff"  
 stroke-width="1.5" stroke-linecap="round" stroke-linejoin="round"></path>  
 </g>  
 </svg>  
 </button>  
 </form>  
 <form action="{{ path('deleteLecture', {id: lecture.id}) }}">  
 <button class="button-delete" type="submit">  
 <svg viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/svg">  
 <g id="SVGRepo\_bgCarrier" stroke-width="0"></g>  
 <g id="SVGRepo\_tracerCarrier" stroke-linecap="round" stroke-linejoin="round"></g>  
 <g id="SVGRepo\_iconCarrier">  
 <path d="M19 5L5 19M5.00001 5L19 19" stroke-width="1.5"  
 stroke-linecap="round" stroke-linejoin="round"></path>  
 </g>  
 </svg>  
 </button>  
 </form>  
 {% endif %}  
 </div>  
 </li>  
 {% endfor %}  
 </ul>  
 </div>  
</div>  
</body>  
</html>

templates/test.html.twig

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>{{ this }}</title>  
 {% block stylesheet %}  
 <link href="{{ asset('css/\_header.css') }}" rel="stylesheet"/>  
 <link href="{{ asset('css/test-main.css') }}" rel="stylesheet"/>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"  
 integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"  
 crossorigin="anonymous">  
 {% endblock %}  
 {% block javascript %}  
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"  
 integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"  
 crossorigin="anonymous"></script>  
 {% endblock %}  
</head>  
<body>  
<div class="wrapper montserrat">  
 {{ block('header', '\_header.html.twig') }}  
 <div class="main">  
 <h1 class="test-header">  
 {{ test.name }}  
 </h1>  
 <form>  
 {% for key, question in test.questions %}  
 <div class="question">  
 {{ question.question }}  
 </div>  
 <ul class="answers">  
 {% for key, answer in question.answers %}  
 <li>  
 <input type="checkbox" id="{{ answer.id }}" name="res[{{ question.id }}][{{ answer.id }}]" value="0">  
 <label for="{{ answer.id }}">  
 {{ answer.answer }}  
 </label>  
 </li>  
 {% endfor %}  
 </ul>  
 {% endfor %}  
 <button type="submit"> Отправить</button>  
 </form>  
 </div>  
  
 <script>  
 const changeValue = (e) => {  
 e.target.setAttribute('value', e.target.getAttribute('value') === '0'? 1 : 0)  
 }  
 document.querySelectorAll('input[type="checkbox"]').forEach(  
 input => input.addEventListener('change', changeValue)  
 )  
 </script>  
</div>  
</body>  
</html>

templates/tests.html.twig

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>{{ this }}</title>  
 {% block stylesheet %}  
 <link href="{{ asset('css/\_header.css') }}" rel="stylesheet"/>  
 <link href="{{ asset('css/test\_list-main.css') }}" rel="stylesheet"/>  
 <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"  
 integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"  
 crossorigin="anonymous">  
 {% endblock %}  
 {% block javascript %}  
 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"  
 integrity="sha384-MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"  
 crossorigin="anonymous"></script>  
 {% endblock %}  
</head>  
<body>  
<div class="wrapper montserrat">  
 {{ block('header', '\_header.html.twig') }}  
 <div class="main">  
 <ul>  
 {% if is\_granted('ROLE\_EDITOR') %}  
 <li class="list-group-item">  
 <a href="{{ path('addTest') }}">Сделать тест</a>  
 </li>  
 {% endif %}  
 {% for key, test in tests %}  
 <li class="lecture border list-group-item">  
 <a href="{{ path('test', {id: test.id}) }}">  
 {{ test.name }}  
 </a>  
 {% if score != "" or size > key %}  
 <span>  
 {{ score[key].score }} баллов из 5  
 </span>  
 {% endif %}  
 <div class="actions float-end">  
 {% if is\_granted('ROLE\_EDITOR') %}  
 <form action="{{ path('deleteTest', {id: test.id}) }}">  
 <button class="button-delete" type="submit">  
 <svg viewBox="0 0 24 24" xmlns="http://www.w3.org/2000/svg">  
 <g id="SVGRepo\_bgCarrier" stroke-width="0"></g>  
 <g id="SVGRepo\_tracerCarrier" stroke-linecap="round" stroke-linejoin="round"></g>  
 <g id="SVGRepo\_iconCarrier">  
 <path d="M19 5L5 19M5.00001 5L19 19" stroke-width="1.5"  
 stroke-linecap="round" stroke-linejoin="round"></path>  
 </g>  
 </svg>  
 </button>  
 </form>  
 {% endif %}  
 </div>  
 </li>  
 {% endfor %}  
 </ul>  
 </div>  
</div>  
</body>  
</html>

templates/userCab.html.twig

<!DOCTYPE html>  
<html lang="">  
<head>  
 <meta charset='UTF-8'>  
 <title>{{ this }}</title>  
 <link href="{{ asset('css/\_header.css') }}" rel="stylesheet"/>  
 <link href="{{ asset('css/userCab-main.css') }}" rel="stylesheet"/>  
 {{ block('stylesheet', 'auth.html.twig') }}  
 {{ block('javascript', 'auth.html.twig') }}  
</head>  
<body>  
<div class="wrapper montserrat">  
 {{ block('header', '\_header.html.twig') }}  
 <div class="container main darkText border-top">  
 <div class="container text-center">  
 <div class="row align-items-start">  
 <div class="col">  
 <span>  
 Имя пользователя: {{ username }}  
 </span>  
 </div>  
 </div>  
 <div class="row align-items-start">  
 <div class="col">  
 <span>  
 Ваш статус: {{ role }}  
 </span>  
 </div>  
 <div class="col">  
 <span>  
 пройдено тестов: {{ tests }}  
 </span>  
 </div>  
 </div>  
 <div class="row align-items-start">  
 {% if is\_granted('ROLE\_ADMIN') %}  
 <div class="col m m-5">  
 <form action="{{ path('admin') }}">  
 <button class="btn btn-light btn-outline-primary btn-lg">  
 Админ панель  
 </button>  
 </form>  
 </div>  
 {% endif %}  
 <div class="col m m-5">  
 <form action="{{ path('logout') }}">  
 <button class="btn btn-light btn-outline-danger btn-lg">  
 Выйти из аккаунта  
 </button>  
 </form>  
 </div>  
 </div>  
 </div>  
 </div>  
</div>  
</body>  
</html>

.env

# In all environments, the following files are loaded if they exist,  
# the latter taking precedence over the former:  
#  
# \* .env contains default values for the environment variables needed by the app  
# \* .env.local uncommitted file with local overrides  
# \* .env.$APP\_ENV committed environment-specific defaults  
# \* .env.$APP\_ENV.local uncommitted environment-specific overrides  
#  
# Real environment variables win over .env files.  
#  
# DO NOT DEFINE PRODUCTION SECRETS IN THIS FILE NOR IN ANY OTHER COMMITTED FILES.  
# https://symfony.com/doc/current/configuration/secrets.html  
#  
# Run "composer dump-env prod" to compile .env files for production use (requires symfony/flex >=1.2).  
# https://symfony.com/doc/current/best\_practices.html#use-environment-variables-for-infrastructure-configuration  
  
###> symfony/framework-bundle ###  
APP\_ENV=prod  
APP\_SECRET=d7d1ea5b2a1d9920354e7f49334ba91e  
###< symfony/framework-bundle ###  
  
###> doctrine/doctrine-bundle ###  
# Format described at https://www.doctrine-project.org/projects/doctrine-dbal/en/latest/reference/configuration.html#connecting-using-a-url  
# IMPORTANT: You MUST configure your server version, either here or in config/packages/doctrine.yaml  
#  
# DATABASE\_URL="sqlite:///%kernel.project\_dir%/var/data.db"  
 DATABASE\_URL="mysql://okUser:!QwE@AsD3zXc@127.0.0.1:3306/techlandbd?serverVersion=mariadb-10.4.11&charset=utf8mb4"  
# DATABASE\_URL="postgresql://app:!ChangeMe!@127.0.0.1:5432/app?serverVersion=15&charset=utf8"  
###< doctrine/doctrine-bundle ###  
  
###> symfony/messenger ###  
# Choose one of the transports below  
# MESSENGER\_TRANSPORT\_DSN=amqp://guest:guest@localhost:5672/%2f/messages  
# MESSENGER\_TRANSPORT\_DSN=redis://localhost:6379/messages  
MESSENGER\_TRANSPORT\_DSN=doctrine://default?auto\_setup=0  
###< symfony/messenger ###  
  
###> symfony/mailer ###  
# MAILER\_DSN=null://null  
###< symfony/mailer ###  
  
  
###> google/recaptcha ###  
# To use Google Recaptcha, you must register a site on Recaptcha's admin panel:  
# https://www.google.com/recaptcha/admin  
GOOGLE\_RECAPTCHA\_SITE\_KEY=6LchkaAkAAAAAN0rwxcJ0v\_JBK0nbfKPaIjiarof  
GOOGLE\_RECAPTCHA\_SECRET=6LchkaAkAAAAAFpsrHsLNRdN2Y\_p2jUqwzYB2g7v  
###< google/recaptcha ###  
  
###> victor-prdh/recaptcha-bundle ###  
GOOGLE\_RECAPTCHA\_SITE\_KEY=6LchkaAkAAAAAN0rwxcJ0v\_JBK0nbfKPaIjiarof  
GOOGLE\_RECAPTCHA\_SECRET\_KEY=6LchkaAkAAAAAFpsrHsLNRdN2Y\_p2jUqwzYB2g7v  
###< victor-prdh/recaptcha-bundle ###

.env.test

# define your env variables for the test env here  
KERNEL\_CLASS='App\Kernel'  
APP\_SECRET='$ecretf0rt3st'  
SYMFONY\_DEPRECATIONS\_HELPER=999999  
PANTHER\_APP\_ENV=panther  
PANTHER\_ERROR\_SCREENSHOT\_DIR=./var/error-screenshots

composer.json

{  
 "type": "project",  
 "license": "proprietary",  
 "minimum-stability": "stable",  
 "prefer-stable": true,  
 "require": {  
 "php": ">=8.1",  
 "ext-ctype": "\*",  
 "ext-iconv": "\*",  
 "ext-intl": "\*",  
 "captcha-com/symfony-captcha-bundle": "4.\*",  
 "cboden/ratchet": "^0.4.4",  
 "doctrine/annotations": "^1.0",  
 "doctrine/doctrine-bundle": "^2.8",  
 "doctrine/doctrine-migrations-bundle": "^3.2",  
 "doctrine/orm": "^2.14",  
 "mhujer/breadcrumbs-bundle": "^1.5",  
 "phpdocumentor/reflection-docblock": "^5.3",  
 "phpstan/phpdoc-parser": "^1.16",  
 "sensio/framework-extra-bundle": "^6.2",  
 "symfony/asset": "6.2.\*",  
 "symfony/console": "6.2.\*",  
 "symfony/doctrine-messenger": "6.2.\*",  
 "symfony/dotenv": "6.2.\*",  
 "symfony/expression-language": "6.2.\*",  
 "symfony/flex": "^2",  
 "symfony/form": "6.2.\*",  
 "symfony/framework-bundle": "6.2.\*",  
 "symfony/http-client": "6.2.\*",  
 "symfony/http-foundation": "6.2.\*",  
 "symfony/intl": "6.2.\*",  
 "symfony/mime": "6.2.\*",  
 "symfony/monolog-bundle": "^3.0",  
 "symfony/notifier": "6.2.\*",  
 "symfony/process": "6.2.\*",  
 "symfony/property-access": "6.2.\*",  
 "symfony/property-info": "6.2.\*",  
 "symfony/runtime": "6.2.\*",  
 "symfony/security-bundle": "6.2.\*",  
 "symfony/security-csrf": "6.2.\*",  
 "symfony/serializer": "6.2.\*",  
 "symfony/string": "6.2.\*",  
 "symfony/translation": "6.2.\*",  
 "symfony/twig-bundle": "6.2.\*",  
 "symfony/validator": "6.2.\*",  
 "symfony/web-link": "6.2.\*",  
 "symfony/yaml": "6.2.\*",  
 "twig/extra-bundle": "^2.12|^3.0",  
 "twig/twig": "^2.12|^3.0",  
 "victor-prdh/recaptcha-bundle": "^1.6"  
 },  
 "config": {  
 "allow-plugins": {  
 "php-http/discovery": true,  
 "symfony/flex": true,  
 "symfony/runtime": true  
 },  
 "sort-packages": true  
 },  
 "autoload": {  
 "psr-4": {  
 "App\\": "src/"  
 }  
 },  
 "autoload-dev": {  
 "psr-4": {  
 "App\\Tests\\": "tests/"  
 }  
 },  
 "replace": {  
 "symfony/polyfill-ctype": "\*",  
 "symfony/polyfill-iconv": "\*",  
 "symfony/polyfill-php72": "\*",  
 "symfony/polyfill-php73": "\*",  
 "symfony/polyfill-php74": "\*",  
 "symfony/polyfill-php80": "\*",  
 "symfony/polyfill-php81": "\*"  
 },  
 "scripts": {  
 "auto-scripts": {  
 "cache:clear": "symfony-cmd",  
 "assets:install %PUBLIC\_DIR%": "symfony-cmd"  
 },  
 "post-install-cmd": [  
 "@auto-scripts"  
 ],  
 "post-update-cmd": [  
 "@auto-scripts"  
 ]  
 },  
 "conflict": {  
 "symfony/symfony": "\*"  
 },  
 "extra": {  
 "symfony": {  
 "allow-contrib": false,  
 "require": "6.2.\*"  
 }  
 },  
 "require-dev": {  
 "phpunit/phpunit": "^9.5",  
 "symfony/browser-kit": "6.2.\*",  
 "symfony/css-selector": "6.2.\*",  
 "symfony/debug-bundle": "6.2.\*",  
 "symfony/maker-bundle": "^1.48",  
 "symfony/phpunit-bridge": "^6.2",  
 "symfony/stopwatch": "6.2.\*",  
 "symfony/web-profiler-bundle": "6.2.\*"  
 }  
}

composer.lock

{  
 "\_readme": [  
 "This file locks the dependencies of your project to a known state",  
 "Read more about it at https://getcomposer.org/doc/01-basic-usage.md#installing-dependencies",  
 "This file is @generated automatically"  
 ],  
 "content-hash": "489646cc037e83b5dd42e69dcdb8d3a3",  
 "packages": [  
 {  
 "name": "captcha-com/captcha",  
 "version": "4.2.5",  
 "source": {  
 "type": "git",  
 "url": "https://git.captcha.com/botdetect-php-captcha-library.git",  
 "reference": "4b8b642509bfc63d2f295dccf1c1b4d0e2266a15"  
 },  
 "require": {  
 "php": ">=5.3.0"  
 },  
 "suggest": {  
 "captcha-com/cakephp-captcha": "CakePHP Captcha Plugin -- BotDetect PHP CAPTCHA generator integration for the CakePHP framework.",  
 "captcha-com/laravel-captcha": "Laravel Captcha Package -- BotDetect PHP CAPTCHA generator integration for the Laravel framework.",  
 "captcha-com/symfony-captcha-bundle": "Symfony Captcha Bundle -- BotDetect PHP CAPTCHA generator integration for the Symfony framework."  
 },  
 "type": "library",  
 "extra": {  
 "branch-alias": {  
 "dev-master": "4.x-dev"  
 }  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "license": [  
 "proprietary"  
 ],  
 "authors": [  
 {  
 "name": "Captcha Support",  
 "email": "botdetect.support@captcha.com",  
 "homepage": "https://captcha.com/php-captcha.html"  
 }  
 ],  
 "description": "Captcha Generator -- BotDetect PHP CAPTCHA Free composer package.",  
 "homepage": "https://captcha.com/php-captcha.html",  
 "keywords": [  
 "captcha",  
 "captcha generator",  
 "captcha library",  
 "php captcha"  
 ],  
 "support": {  
 "email": "botdetect.support@captcha.com",  
 "forum": "https://captcha.com/support",  
 "issues": "https://captcha.com/support",  
 "source": "https://captcha.com/php-captcha.html"  
 },  
 "time": "2019-07-22T16:36:53+00:00"  
 },  
 {  
 "name": "captcha-com/symfony-captcha-bundle",  
 "version": "4.2.12",  
 "source": {  
 "type": "git",  
 "url": "https://github.com/captcha-com/symfony-captcha-bundle.git",  
 "reference": "fd071c3f36d646ff87b591ddb3ea531c5f3678c2"  
 },  
 "dist": {  
 "type": "zip",  
 "url": "https://api.github.com/repos/captcha-com/symfony-captcha-bundle/zipball/fd071c3f36d646ff87b591ddb3ea531c5f3678c2",  
 "reference": "fd071c3f36d646ff87b591ddb3ea531c5f3678c2",  
 "shasum": ""  
 },  
 "require": {  
 "captcha-com/captcha": "4.\*",  
 "php": ">=5.3.9"  
 },  
 "type": "symfony-bundle",  
 "extra": {  
 "branch-alias": {  
 "dev-master": "4.x-dev"  
 }  
 },  
 "autoload": {  
 "psr-4": {  
 "Captcha\\Bundle\\CaptchaBundle\\": "/"  
 }  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "authors": [  
 {  
 "name": "Symfony Captcha Bundle Support",  
 "email": "botdetect.support@captcha.com",  
 "homepage": "https://captcha.com/doc/php/symfony-captcha-bundle-quickstart.html"  
 }  
 ],  
 "description": "Symfony Captcha Bundle -- BotDetect PHP Captcha generator integration for the Symfony framework.",  
 "homepage": "https://captcha.com/doc/php/symfony-captcha-bundle-quickstart.html",  
 "keywords": [  
 "botdetect",  
 "captcha",  
 "captcha bundle",  
 "captcha generator",  
 "symfony captcha",  
 "symfony captcha bundle"  
 ],  
 "support": {  
 "email": "botdetect.support@captcha.com",  
 "forum": "https://captcha.com/support",  
 "issues": "https://captcha.com/support",  
 "source": "https://captcha.com/doc/php/symfony-captcha-bundle-quickstart.html"  
 },  
 "time": "2019-08-08T16:56:37+00:00"  
 },  
 {  
 "name": "cboden/ratchet",  
 "version": "v0.4.4",  
 "source": {  
 "type": "git",  
 "url": "https://github.com/ratchetphp/Ratchet.git",  
 "reference": "5012dc954541b40c5599d286fd40653f5716a38f"  
 },  
 "dist": {  
 "type": "zip",  
 "url": "https://api.github.com/repos/ratchetphp/Ratchet/zipball/5012dc954541b40c5599d286fd40653f5716a38f",  
 "reference": "5012dc954541b40c5599d286fd40653f5716a38f",  
 "shasum": ""  
 },  
 "require": {  
 "guzzlehttp/psr7": "^1.7|^2.0",  
 "php": ">=5.4.2",  
 "ratchet/rfc6455": "^0.3.1",  
 "react/event-loop": ">=0.4",  
 "react/socket": "^1.0 || ^0.8 || ^0.7 || ^0.6 || ^0.5",  
 "symfony/http-foundation": "^2.6|^3.0|^4.0|^5.0|^6.0",  
 "symfony/routing": "^2.6|^3.0|^4.0|^5.0|^6.0"  
 },  
 "require-dev": {  
 "phpunit/phpunit": "~4.8"  
 },  
 "type": "library",  
 "autoload": {  
 "psr-4": {  
 "Ratchet\\": "src/Ratchet"  
 }  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "license": [  
 "MIT"  
 ],  
 "authors": [  
 {  
 "name": "Chris Boden",  
 "email": "cboden@gmail.com",  
 "role": "Developer"  
 },  
 {  
 "name": "Matt Bonneau",  
 "role": "Developer"  
 }  
 ],  
 "description": "PHP WebSocket library",  
 "homepage": "http://socketo.me",  
 "keywords": [  
 "Ratchet",  
 "WebSockets",  
 "server",  
 "sockets",  
 "websocket"  
 ],  
 "support": {  
 "chat": "https://gitter.im/reactphp/reactphp",  
 "issues": "https://github.com/ratchetphp/Ratchet/issues",  
 "source": "https://github.com/ratchetphp/Ratchet/tree/v0.4.4"  
 },  
 "time": "2021-12-14T00:20:41+00:00"  
 },  
 {  
 "name": "doctrine/annotations",  
 "version": "1.14.3",  
 "source": {  
 "type": "git",  
 "url": "https://github.com/doctrine/annotations.git",  
 "reference": "fb0d71a7393298a7b232cbf4c8b1f73f3ec3d5af"  
 },  
 "dist": {  
 "type": "zip",  
 "url": "https://api.github.com/repos/doctrine/annotations/zipball/fb0d71a7393298a7b232cbf4c8b1f73f3ec3d5af",  
 "reference": "fb0d71a7393298a7b232cbf4c8b1f73f3ec3d5af",  
 "shasum": ""  
 },  
 "require": {  
 "doctrine/lexer": "^1 || ^2",  
 "ext-tokenizer": "\*",  
 "php": "^7.1 || ^8.0",  
 "psr/cache": "^1 || ^2 || ^3"  
 },  
 "require-dev": {  
 "doctrine/cache": "^1.11 || ^2.0",  
 "doctrine/coding-standard": "^9 || ^10",  
 "phpstan/phpstan": "~1.4.10 || ^1.8.0",  
 "phpunit/phpunit": "^7.5 || ^8.5 || ^9.5",  
 "symfony/cache": "^4.4 || ^5.4 || ^6",  
 "vimeo/psalm": "^4.10"  
 },  
 "suggest": {  
 "php": "PHP 8.0 or higher comes with attributes, a native replacement for annotations"  
 },  
 "type": "library",  
 "autoload": {  
 "psr-4": {  
 "Doctrine\\Common\\Annotations\\": "lib/Doctrine/Common/Annotations"  
 }  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "license": [  
 "MIT"  
 ],  
 "authors": [  
 {  
 "name": "Guilherme Blanco",  
 "email": "guilhermeblanco@gmail.com"  
 },  
 {  
 "name": "Roman Borschel",  
 "email": "roman@code-factory.org"  
 },  
 {  
 "name": "Benjamin Eberlei",  
 "email": "kontakt@beberlei.de"  
 },  
 {  
 "name": "Jonathan Wage",  
 "email": "jonwage@gmail.com"  
 },  
 {  
 "name": "Johannes Schmitt",  
 "email": "schmittjoh@gmail.com"  
 }  
 ],  
 "description": "Docblock Annotations Parser",  
 "homepage": "https://www.doctrine-project.org/projects/annotations.html",  
 "keywords": [  
 "annotations",  
 "docblock",  
 "parser"  
 ],  
 "support": {  
 "issues": "https://github.com/doctrine/annotations/issues",  
 "source": "https://github.com/doctrine/annotations/tree/1.14.3"  
 },  
 "time": "2023-02-01T09:20:38+00:00"  
 },  
 {  
 "name": "doctrine/cache",  
 "version": "2.2.0",  
 "source": {  
 "type": "git",  
 "url": "https://github.com/doctrine/cache.git",  
 "reference": "1ca8f21980e770095a31456042471a57bc4c68fb"  
 },  
 "dist": {  
 "type": "zip",  
 "url": "https://api.github.com/repos/doctrine/cache/zipball/1ca8f21980e770095a31456042471a57bc4c68fb",  
 "reference": "1ca8f21980e770095a31456042471a57bc4c68fb",  
 "shasum": ""  
 },  
 "require": {  
 "php": "~7.1 || ^8.0"  
 },  
 "conflict": {  
 "doctrine/common": ">2.2,<2.4"  
 },  
 "require-dev": {  
 "cache/integration-tests": "dev-master",  
 "doctrine/coding-standard": "^9",  
 "phpunit/phpunit": "^7.5 || ^8.5 || ^9.5",  
 "psr/cache": "^1.0 || ^2.0 || ^3.0",  
 "symfony/cache": "^4.4 || ^5.4 || ^6",  
 "symfony/var-exporter": "^4.4 || ^5.4 || ^6"  
 },  
 "type": "library",  
 "autoload": {  
 "psr-4": {  
 "Doctrine\\Common\\Cache\\": "lib/Doctrine/Common/Cache"  
 }  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "license": [  
 "MIT"  
 ],  
 "authors": [  
 {  
 "name": "Guilherme Blanco",  
 "email": "guilhermeblanco@gmail.com"  
 },  
 {  
 "name": "Roman Borschel",  
 "email": "roman@code-factory.org"  
 },  
 {  
 "name": "Benjamin Eberlei",  
 "email": "kontakt@beberlei.de"  
 },  
 {  
 "name": "Jonathan Wage",  
 "email": "jonwage@gmail.com"  
 },  
 {  
 "name": "Johannes Schmitt",  
 "email": "schmittjoh@gmail.com"  
 }  
 ],  
 "description": "PHP Doctrine Cache library is a popular cache implementation that supports many different drivers such as redis, memcache, apc, mongodb and others.",  
 "homepage": "https://www.doctrine-project.org/projects/cache.html",  
 "keywords": [  
 "abstraction",  
 "apcu",  
 "cache",  
 "caching",  
 "couchdb",  
 "memcached",  
 "php",  
 "redis",  
 "xcache"  
 ],  
 "support": {  
 "issues": "https://github.com/doctrine/cache/issues",  
 "source": "https://github.com/doctrine/cache/tree/2.2.0"  
 },  
 "funding": [  
 {  
 "url": "https://www.doctrine-project.org/sponsorship.html",  
 "type": "custom"  
 },  
 {  
 "url": "https://www.patreon.com/phpdoctrine",  
 "type": "patreon"  
 },  
 {  
 "url": "https://tidelift.com/funding/github/packagist/doctrine%2Fcache",  
 "type": "tidelift"  
 }  
 ],  
 "time": "2022-05-20T20:07:39+00:00"  
 },  
 {  
 "name": "doctrine/collections",  
 "version": "2.1.2",  
 "source": {  
 "type": "git",  
 "url": "https://github.com/doctrine/collections.git",  
 "reference": "db8cda536a034337f7dd63febecc713d4957f9ee"  
 },  
 "dist": {  
 "type": "zip",  
 "url": "https://api.github.com/repos/doctrine/collections/zipball/db8cda536a034337f7dd63febecc713d4957f9ee",  
 "reference": "db8cda536a034337f7dd63febecc713d4957f9ee",  
 "shasum": ""  
 },  
 "require": {  
 "doctrine/deprecations": "^1",  
 "php": "^8.1"  
 },  
 "require-dev": {  
 "doctrine/coding-standard": "^10.0",  
 "ext-json": "\*",  
 "phpstan/phpstan": "^1.8",  
 "phpstan/phpstan-phpunit": "^1.0",  
 "phpunit/phpunit": "^9.5",  
 "vimeo/psalm": "^4.22"  
 },  
 "type": "library",  
 "autoload": {  
 "psr-4": {  
 "Doctrine\\Common\\Collections\\": "src"  
 }  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "license": [  
 "MIT"  
 ],  
 "authors": [  
 {  
 "name": "Guilherme Blanco",  
 "email": "guilhermeblanco@gmail.com"  
 },  
 {  
 "name": "Roman Borschel",  
 "email": "roman@code-factory.org"  
 },  
 {  
 "name": "Benjamin Eberlei",  
 "email": "kontakt@beberlei.de"  
 },  
 {  
 "name": "Jonathan Wage",  
 "email": "jonwage@gmail.com"  
 },  
 {  
 "name": "Johannes Schmitt",  
 "email": "schmittjoh@gmail.com"  
 }  
 ],  
 "description": "PHP Doctrine Collections library that adds additional functionality on top of PHP arrays.",  
 "homepage": "https://www.doctrine-project.org/projects/collections.html",  
 "keywords": [  
 "array",  
 "collections",  
 "iterators",  
 "php"  
 ],  
 "support": {  
 "issues": "https://github.com/doctrine/collections/issues",  
 "source": "https://github.com/doctrine/collections/tree/2.1.2"  
 },  
 "funding": [  
 {  
 "url": "https://www.doctrine-project.org/sponsorship.html",  
 "type": "custom"  
 },  
 {  
 "url": "https://www.patreon.com/phpdoctrine",  
 "type": "patreon"  
 },  
 {  
 "url": "https://tidelift.com/funding/github/packagist/doctrine%2Fcollections",  
 "type": "tidelift"  
 }  
 ],  
 "time": "2022-12-27T23:41:38+00:00"  
 },  
 {  
 "name": "doctrine/common",  
 "version": "3.4.3",  
 "source": {  
 "type": "git",  
 "url": "https://github.com/doctrine/common.git",  
 "reference": "8b5e5650391f851ed58910b3e3d48a71062eeced"  
 },  
 "dist": {  
 "type": "zip",  
 "url": "https://api.github.com/repos/doctrine/common/zipball/8b5e5650391f851ed58910b3e3d48a71062eeced",  
 "reference": "8b5e5650391f851ed58910b3e3d48a71062eeced",  
 "shasum": ""  
 },  
 "require": {  
 "doctrine/persistence": "^2.0 || ^3.0",  
 "php": "^7.1 || ^8.0"  
 },  
 "require-dev": {  
 "doctrine/coding-standard": "^9.0 || ^10.0",  
 "doctrine/collections": "^1",  
 "phpstan/phpstan": "^1.4.1",  
 "phpstan/phpstan-phpunit": "^1",  
 "phpunit/phpunit": "^7.5.20 || ^8.5 || ^9.0",  
 "squizlabs/php\_codesniffer": "^3.0",  
 "symfony/phpunit-bridge": "^6.1",  
 "vimeo/psalm": "^4.4"  
 },  
 "type": "library",  
 "autoload": {  
 "psr-4": {  
 "Doctrine\\Common\\": "src"  
 }  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "license": [  
 "MIT"  
 ],  
 "authors": [  
 {  
 "name": "Guilherme Blanco",  
 "email": "guilhermeblanco@gmail.com"  
 },  
 {  
 "name": "Roman Borschel",  
 "email": "roman@code-factory.org"  
 },  
 {  
 "name": "Benjamin Eberlei",  
 "email": "kontakt@beberlei.de"  
 },  
 {  
 "name": "Jonathan Wage",  
 "email": "jonwage@gmail.com"  
 },  
 {  
 "name": "Johannes Schmitt",  
 "email": "schmittjoh@gmail.com"  
 },  
 {  
 "name": "Marco Pivetta",  
 "email": "ocramius@gmail.com"  
 }  
 ],  
 "description": "PHP Doctrine Common project is a library that provides additional functionality that other Doctrine projects depend on such as better reflection support, proxies and much more.",  
 "homepage": "https://www.doctrine-project.org/projects/common.html",  
 "keywords": [  
 "common",  
 "doctrine",  
 "php"  
 ],  
 "support": {  
 "issues": "https://github.com/doctrine/common/issues",  
 "source": "https://github.com/doctrine/common/tree/3.4.3"  
 },  
 "funding": [  
 {  
 "url": "https://www.doctrine-project.org/sponsorship.html",  
 "type": "custom"  
 },  
 {  
 "url": "https://www.patreon.com/phpdoctrine",  
 "type": "patreon"  
 },  
 {  
 "url": "https://tidelift.com/funding/github/packagist/doctrine%2Fcommon",  
 "type": "tidelift"  
 }  
 ],  
 "time": "2022-10-09T11:47:59+00:00"  
 },  
 {  
 "name": "doctrine/dbal",  
 "version": "3.6.0",  
 "source": {  
 "type": "git",  
 "url": "https://github.com/doctrine/dbal.git",  
 "reference": "85b98cb23c8af471a67abfe14485da696bcabc2e"  
 },  
 "dist": {  
 "type": "zip",  
 "url": "https://api.github.com/repos/doctrine/dbal/zipball/85b98cb23c8af471a67abfe14485da696bcabc2e",  
 "reference": "85b98cb23c8af471a67abfe14485da696bcabc2e",  
 "shasum": ""  
 },  
 "require": {  
 "composer-runtime-api": "^2",  
 "doctrine/cache": "^1.11|^2.0",  
 "doctrine/deprecations": "^0.5.3|^1",  
 "doctrine/event-manager": "^1|^2",  
 "php": "^7.4 || ^8.0",  
 "psr/cache": "^1|^2|^3",  
 "psr/log": "^1|^2|^3"  
 },  
 "require-dev": {  
 "doctrine/coding-standard": "11.1.0",  
 "fig/log-test": "^1",  
 "jetbrains/phpstorm-stubs": "2022.3",  
 "phpstan/phpstan": "1.9.14",  
 "phpstan/phpstan-strict-rules": "^1.4",  
 "phpunit/phpunit": "9.6.3",  
 "psalm/plugin-phpunit": "0.18.4",  
 "squizlabs/php\_codesniffer": "3.7.1",  
 "symfony/cache": "^5.4|^6.0",  
 "symfony/console": "^4.4|^5.4|^6.0",  
 "vimeo/psalm": "4.30.0"  
 },  
 "suggest": {  
 "symfony/console": "For helpful console commands such as SQL execution and import of files."  
 },  
 "bin": [  
 "bin/doctrine-dbal"  
 ],  
 "type": "library",  
 "autoload": {  
 "psr-4": {  
 "Doctrine\\DBAL\\": "src"  
 }  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "license": [  
 "MIT"  
 ],  
 "authors": [  
 {  
 "name": "Guilherme Blanco",  
 "email": "guilhermeblanco@gmail.com"  
 },  
 {  
 "name": "Roman Borschel",  
 "email": "roman@code-factory.org"  
 },  
 {  
 "name": "Benjamin Eberlei",  
 "email": "kontakt@beberlei.de"  
 },  
 {  
 "name": "Jonathan Wage",  
 "email": "jonwage@gmail.com"  
 }  
 ],  
 "description": "Powerful PHP database abstraction layer (DBAL) with many features for database schema introspection and management.",  
 "homepage": "https://www.doctrine-project.org/projects/dbal.html",  
 "keywords": [  
 "abstraction",  
 "database",  
 "db2",  
 "dbal",  
 "mariadb",  
 "mssql",  
 "mysql",  
 "oci8",  
 "oracle",  
 "pdo",  
 "pgsql",  
 "postgresql",  
 "queryobject",  
 "sasql",  
 "sql",  
 "sqlite",  
 "sqlserver",  
 "sqlsrv"  
 ],  
 "support": {  
 "issues": "https://github.com/doctrine/dbal/issues",  
 "source": "https://github.com/doctrine/dbal/tree/3.6.0"  
 },  
 "funding": [  
 {  
 "url": "https://www.doctrine-project.org/sponsorship.html",  
 "type": "custom"  
 },  
 {  
 "url": "https://www.patreon.com/phpdoctrine",  
 "type": "patreon"  
 },  
 {  
 "url": "https://tidelift.com/funding/github/packagist/doctrine%2Fdbal",  
 "type": "tidelift"  
 }  
 ],  
 "time": "2023-02-07T22:52:03+00:00"  
 },  
 {  
 "name": "doctrine/deprecations",  
 "version": "v1.0.0",  
 "source": {  
 "type": "git",  
 "url": "https://github.com/doctrine/deprecations.git",  
 "reference": "0e2a4f1f8cdfc7a92ec3b01c9334898c806b30de"  
 },  
 "dist": {  
 "type": "zip",  
 "url": "https://api.github.com/repos/doctrine/deprecations/zipball/0e2a4f1f8cdfc7a92ec3b01c9334898c806b30de",  
 "reference": "0e2a4f1f8cdfc7a92ec3b01c9334898c806b30de",  
 "shasum": ""  
 },  
 "require": {  
 "php": "^7.1|^8.0"  
 },  
 "require-dev": {  
 "doctrine/coding-standard": "^9",  
 "phpunit/phpunit": "^7.5|^8.5|^9.5",  
 "psr/log": "^1|^2|^3"  
 },  
 "suggest": {  
 "psr/log": "Allows logging deprecations via PSR-3 logger implementation"  
 },  
 "type": "library",  
 "autoload": {  
 "psr-4": {  
 "Doctrine\\Deprecations\\": "lib/Doctrine/Deprecations"  
 }  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "license": [  
 "MIT"  
 ],  
 "description": "A small layer on top of trigger\_error(E\_USER\_DEPRECATED) or PSR-3 logging with options to disable all deprecations or selectively for packages.",  
 "homepage": "https://www.doctrine-project.org/",  
 "support": {  
 "issues": "https://github.com/doctrine/deprecations/issues",  
 "source": "https://github.com/doctrine/deprecations/tree/v1.0.0"  
 },  
 "time": "2022-05-02T15:47:09+00:00"  
 },  
 {  
 "name": "doctrine/doctrine-bundle",  
 "version": "2.8.3",  
 "source": {  
 "type": "git",  
 "url": "https://github.com/doctrine/DoctrineBundle.git",  
 "reference": "fd67ba64db3c806f626a33dcab15a4db0c77652e"  
 },  
 "dist": {  
 "type": "zip",  
 "url": "https://api.github.com/repos/doctrine/DoctrineBundle/zipball/fd67ba64db3c806f626a33dcab15a4db0c77652e",  
 "reference": "fd67ba64db3c806f626a33dcab15a4db0c77652e",  
 "shasum": ""  
 },  
 "require": {  
 "doctrine/cache": "^1.11 || ^2.0",  
 "doctrine/dbal": "^3.4.0",  
 "doctrine/persistence": "^2.2 || ^3",  
 "doctrine/sql-formatter": "^1.0.1",  
 "php": "^7.4 || ^8.0",  
 "symfony/cache": "^5.4 || ^6.0",  
 "symfony/config": "^5.4 || ^6.0",  
 "symfony/console": "^5.4 || ^6.0",  
 "symfony/dependency-injection": "^5.4 || ^6.0",  
 "symfony/deprecation-contracts": "^2.1 || ^3",  
 "symfony/doctrine-bridge": "^5.4.19 || ^6.0.7",  
 "symfony/framework-bundle": "^5.4 || ^6.0",  
 "symfony/service-contracts": "^1.1.1 || ^2.0 || ^3"  
 },  
 "conflict": {  
 "doctrine/annotations": ">=3.0",  
 "doctrine/orm": "<2.11 || >=3.0",  
 "twig/twig": "<1.34 || >=2.0,<2.4"  
 },  
 "require-dev": {  
 "doctrine/annotations": "^1 || ^2",  
 "doctrine/coding-standard": "^9.0",  
 "doctrine/orm": "^2.11 || ^3.0",  
 "friendsofphp/proxy-manager-lts": "^1.0",  
 "phpunit/phpunit": "^9.5.26 || ^10.0",  
 "psalm/plugin-phpunit": "^0.18.4",  
 "psalm/plugin-symfony": "^4",  
 "psr/log": "^1.1.4 || ^2.0 || ^3.0",  
 "symfony/phpunit-bridge": "^6.1",  
 "symfony/property-info": "^5.4 || ^6.0",  
 "symfony/proxy-manager-bridge": "^5.4 || ^6.0",  
 "symfony/security-bundle": "^5.4 || ^6.0",  
 "symfony/twig-bridge": "^5.4 || ^6.0",  
 "symfony/validator": "^5.4 || ^6.0",  
 "symfony/web-profiler-bundle": "^5.4 || ^6.0",  
 "symfony/yaml": "^5.4 || ^6.0",  
 "twig/twig": "^1.34 || ^2.12 || ^3.0",  
 "vimeo/psalm": "^4.30"  
 },  
 "suggest": {  
 "doctrine/orm": "The Doctrine ORM integration is optional in the bundle.",  
 "ext-pdo": "\*",  
 "symfony/web-profiler-bundle": "To use the data collector."  
 },  
 "type": "symfony-bundle",  
 "autoload": {  
 "psr-4": {  
 "Doctrine\\Bundle\\DoctrineBundle\\": ""  
 }  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "license": [  
 "MIT"  
 ],  
 "authors": [  
 {  
 "name": "Fabien Potencier",  
 "email": "fabien@symfony.com"  
 },  
 {  
 "name": "Benjamin Eberlei",  
 "email": "kontakt@beberlei.de"  
 },  
 {  
 "name": "Symfony Community",  
 "homepage": "https://symfony.com/contributors"  
 },  
 {  
 "name": "Doctrine Project",  
 "homepage": "https://www.doctrine-project.org/"  
 }  
 ],  
 "description": "Symfony DoctrineBundle",  
 "homepage": "https://www.doctrine-project.org",  
 "keywords": [  
 "database",  
 "dbal",  
 "orm",  
 "persistence"  
 ],  
 "support": {  
 "issues": "https://github.com/doctrine/DoctrineBundle/issues",  
 "source": "https://github.com/doctrine/DoctrineBundle/tree/2.8.3"  
 },  
 "funding": [  
 {  
 "url": "https://www.doctrine-project.org/sponsorship.html",  
 "type": "custom"  
 },  
 {  
 "url": "https://www.patreon.com/phpdoctrine",  
 "type": "patreon"  
 },  
 {  
 "url": "https://tidelift.com/funding/github/packagist/doctrine%2Fdoctrine-bundle",  
 "type": "tidelift"  
 }  
 ],  
 "time": "2023-02-03T09:32:42+00:00"  
 },  
 {  
 "name": "doctrine/doctrine-migrations-bundle",  
 "version": "3.2.2",  
 "source": {  
 "type": "git",  
 "url": "https://github.com/doctrine/DoctrineMigrationsBundle.git",  
 "reference": "3393f411ba25ade21969c33f2053220044854d01"  
 },  
 "dist": {  
 "type": "zip",  
 "url": "https://api.github.com/repos/doctrine/DoctrineMigrationsBundle/zipball/3393f411ba25ade21969c33f2053220044854d01",  
 "reference": "3393f411ba25ade21969c33f2053220044854d01",  
 "shasum": ""  
 },  
 "require": {  
 "doctrine/doctrine-bundle": "~1.0|~2.0",  
 "doctrine/migrations": "^3.2",  
 "php": "^7.2|^8.0",  
 "symfony/framework-bundle": "~3.4|~4.0|~5.0|~6.0"  
 },  
 "require-dev": {  
 "doctrine/coding-standard": "^8.0",  
 "doctrine/orm": "^2.6",  
 "doctrine/persistence": "^1.3||^2.0",  
 "phpstan/phpstan": "^0.12",  
 "phpstan/phpstan-deprecation-rules": "^0.12",  
 "phpstan/phpstan-phpunit": "^0.12",  
 "phpstan/phpstan-strict-rules": "^0.12",  
 "phpunit/phpunit": "^8.0|^9.0",  
 "vimeo/psalm": "^4.11"  
 },  
 "type": "symfony-bundle",  
 "autoload": {  
 "psr-4": {  
 "Doctrine\\Bundle\\MigrationsBundle\\": ""  
 },  
 "exclude-from-classmap": [  
 "/Tests/"  
 ]  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "license": [  
 "MIT"  
 ],  
 "authors": [  
 {  
 "name": "Fabien Potencier",  
 "email": "fabien@symfony.com"  
 },  
 {  
 "name": "Doctrine Project",  
 "homepage": "https://www.doctrine-project.org"  
 },  
 {  
 "name": "Symfony Community",  
 "homepage": "https://symfony.com/contributors"  
 }  
 ],  
 "description": "Symfony DoctrineMigrationsBundle",  
 "homepage": "https://www.doctrine-project.org",  
 "keywords": [  
 "dbal",  
 "migrations",  
 "schema"  
 ],  
 "support": {  
 "issues": "https://github.com/doctrine/DoctrineMigrationsBundle/issues",  
 "source": "https://github.com/doctrine/DoctrineMigrationsBundle/tree/3.2.2"  
 },  
 "funding": [  
 {  
 "url": "https://www.doctrine-project.org/sponsorship.html",  
 "type": "custom"  
 },  
 {  
 "url": "https://www.patreon.com/phpdoctrine",  
 "type": "patreon"  
 },  
 {  
 "url": "https://tidelift.com/funding/github/packagist/doctrine%2Fdoctrine-migrations-bundle",  
 "type": "tidelift"  
 }  
 ],  
 "time": "2022-02-01T18:08:07+00:00"  
 },  
 {  
 "name": "doctrine/event-manager",  
 "version": "2.0.0",  
 "source": {  
 "type": "git",  
 "url": "https://github.com/doctrine/event-manager.git",  
 "reference": "750671534e0241a7c50ea5b43f67e23eb5c96f32"  
 },  
 "dist": {  
 "type": "zip",  
 "url": "https://api.github.com/repos/doctrine/event-manager/zipball/750671534e0241a7c50ea5b43f67e23eb5c96f32",  
 "reference": "750671534e0241a7c50ea5b43f67e23eb5c96f32",  
 "shasum": ""  
 },  
 "require": {  
 "php": "^8.1"  
 },  
 "conflict": {  
 "doctrine/common": "<2.9"  
 },  
 "require-dev": {  
 "doctrine/coding-standard": "^10",  
 "phpstan/phpstan": "^1.8.8",  
 "phpunit/phpunit": "^9.5",  
 "vimeo/psalm": "^4.28"  
 },  
 "type": "library",  
 "autoload": {  
 "psr-4": {  
 "Doctrine\\Common\\": "src"  
 }  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "license": [  
 "MIT"  
 ],  
 "authors": [  
 {  
 "name": "Guilherme Blanco",  
 "email": "guilhermeblanco@gmail.com"  
 },  
 {  
 "name": "Roman Borschel",  
 "email": "roman@code-factory.org"  
 },  
 {  
 "name": "Benjamin Eberlei",  
 "email": "kontakt@beberlei.de"  
 },  
 {  
 "name": "Jonathan Wage",  
 "email": "jonwage@gmail.com"  
 },  
 {  
 "name": "Johannes Schmitt",  
 "email": "schmittjoh@gmail.com"  
 },  
 {  
 "name": "Marco Pivetta",  
 "email": "ocramius@gmail.com"  
 }  
 ],  
 "description": "The Doctrine Event Manager is a simple PHP event system that was built to be used with the various Doctrine projects.",  
 "homepage": "https://www.doctrine-project.org/projects/event-manager.html",  
 "keywords": [  
 "event",  
 "event dispatcher",  
 "event manager",  
 "event system",  
 "events"  
 ],  
 "support": {  
 "issues": "https://github.com/doctrine/event-manager/issues",  
 "source": "https://github.com/doctrine/event-manager/tree/2.0.0"  
 },  
 "funding": [  
 {  
 "url": "https://www.doctrine-project.org/sponsorship.html",  
 "type": "custom"  
 },  
 {  
 "url": "https://www.patreon.com/phpdoctrine",  
 "type": "patreon"  
 },  
 {  
 "url": "https://tidelift.com/funding/github/packagist/doctrine%2Fevent-manager",  
 "type": "tidelift"  
 }  
 ],  
 "time": "2022-10-12T20:59:15+00:00"  
 },  
 {  
 "name": "doctrine/inflector",  
 "version": "2.0.6",  
 "source": {  
 "type": "git",  
 "url": "https://github.com/doctrine/inflector.git",  
 "reference": "d9d313a36c872fd6ee06d9a6cbcf713eaa40f024"  
 },  
 "dist": {  
 "type": "zip",  
 "url": "https://api.github.com/repos/doctrine/inflector/zipball/d9d313a36c872fd6ee06d9a6cbcf713eaa40f024",  
 "reference": "d9d313a36c872fd6ee06d9a6cbcf713eaa40f024",  
 "shasum": ""  
 },  
 "require": {  
 "php": "^7.2 || ^8.0"  
 },  
 "require-dev": {  
 "doctrine/coding-standard": "^10",  
 "phpstan/phpstan": "^1.8",  
 "phpstan/phpstan-phpunit": "^1.1",  
 "phpstan/phpstan-strict-rules": "^1.3",  
 "phpunit/phpunit": "^8.5 || ^9.5",  
 "vimeo/psalm": "^4.25"  
 },  
 "type": "library",  
 "autoload": {  
 "psr-4": {  
 "Doctrine\\Inflector\\": "lib/Doctrine/Inflector"  
 }  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "license": [  
 "MIT"  
 ],  
 "authors": [  
 {  
 "name": "Guilherme Blanco",  
 "email": "guilhermeblanco@gmail.com"  
 },  
 {  
 "name": "Roman Borschel",  
 "email": "roman@code-factory.org"  
 },  
 {  
 "name": "Benjamin Eberlei",  
 "email": "kontakt@beberlei.de"  
 },  
 {  
 "name": "Jonathan Wage",  
 "email": "jonwage@gmail.com"  
 },  
 {  
 "name": "Johannes Schmitt",  
 "email": "schmittjoh@gmail.com"  
 }  
 ],  
 "description": "PHP Doctrine Inflector is a small library that can perform string manipulations with regard to upper/lowercase and singular/plural forms of words.",  
 "homepage": "https://www.doctrine-project.org/projects/inflector.html",  
 "keywords": [  
 "inflection",  
 "inflector",  
 "lowercase",  
 "manipulation",  
 "php",  
 "plural",  
 "singular",  
 "strings",  
 "uppercase",  
 "words"  
 ],  
 "support": {  
 "issues": "https://github.com/doctrine/inflector/issues",  
 "source": "https://github.com/doctrine/inflector/tree/2.0.6"  
 },  
 "funding": [  
 {  
 "url": "https://www.doctrine-project.org/sponsorship.html",  
 "type": "custom"  
 },  
 {  
 "url": "https://www.patreon.com/phpdoctrine",  
 "type": "patreon"  
 },  
 {  
 "url": "https://tidelift.com/funding/github/packagist/doctrine%2Finflector",  
 "type": "tidelift"  
 }  
 ],  
 "time": "2022-10-20T09:10:12+00:00"  
 },  
 {  
 "name": "doctrine/instantiator",  
 "version": "1.5.0",  
 "source": {  
 "type": "git",  
 "url": "https://github.com/doctrine/instantiator.git",  
 "reference": "0a0fa9780f5d4e507415a065172d26a98d02047b"  
 },  
 "dist": {  
 "type": "zip",  
 "url": "https://api.github.com/repos/doctrine/instantiator/zipball/0a0fa9780f5d4e507415a065172d26a98d02047b",  
 "reference": "0a0fa9780f5d4e507415a065172d26a98d02047b",  
 "shasum": ""  
 },  
 "require": {  
 "php": "^7.1 || ^8.0"  
 },  
 "require-dev": {  
 "doctrine/coding-standard": "^9 || ^11",  
 "ext-pdo": "\*",  
 "ext-phar": "\*",  
 "phpbench/phpbench": "^0.16 || ^1",  
 "phpstan/phpstan": "^1.4",  
 "phpstan/phpstan-phpunit": "^1",  
 "phpunit/phpunit": "^7.5 || ^8.5 || ^9.5",  
 "vimeo/psalm": "^4.30 || ^5.4"  
 },  
 "type": "library",  
 "autoload": {  
 "psr-4": {  
 "Doctrine\\Instantiator\\": "src/Doctrine/Instantiator/"  
 }  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "license": [  
 "MIT"  
 ],  
 "authors": [  
 {  
 "name": "Marco Pivetta",  
 "email": "ocramius@gmail.com",  
 "homepage": "https://ocramius.github.io/"  
 }  
 ],  
 "description": "A small, lightweight utility to instantiate objects in PHP without invoking their constructors",  
 "homepage": "https://www.doctrine-project.org/projects/instantiator.html",  
 "keywords": [  
 "constructor",  
 "instantiate"  
 ],  
 "support": {  
 "issues": "https://github.com/doctrine/instantiator/issues",  
 "source": "https://github.com/doctrine/instantiator/tree/1.5.0"  
 },  
 "funding": [  
 {  
 "url": "https://www.doctrine-project.org/sponsorship.html",  
 "type": "custom"  
 },  
 {  
 "url": "https://www.patreon.com/phpdoctrine",  
 "type": "patreon"  
 },  
 {  
 "url": "https://tidelift.com/funding/github/packagist/doctrine%2Finstantiator",  
 "type": "tidelift"  
 }  
 ],  
 "time": "2022-12-30T00:15:36+00:00"  
 },  
 {  
 "name": "doctrine/lexer",  
 "version": "2.1.0",  
 "source": {  
 "type": "git",  
 "url": "https://github.com/doctrine/lexer.git",  
 "reference": "39ab8fcf5a51ce4b85ca97c7a7d033eb12831124"  
 },  
 "dist": {  
 "type": "zip",  
 "url": "https://api.github.com/repos/doctrine/lexer/zipball/39ab8fcf5a51ce4b85ca97c7a7d033eb12831124",  
 "reference": "39ab8fcf5a51ce4b85ca97c7a7d033eb12831124",  
 "shasum": ""  
 },  
 "require": {  
 "doctrine/deprecations": "^1.0",  
 "php": "^7.1 || ^8.0"  
 },  
 "require-dev": {  
 "doctrine/coding-standard": "^9 || ^10",  
 "phpstan/phpstan": "^1.3",  
 "phpunit/phpunit": "^7.5 || ^8.5 || ^9.5",  
 "psalm/plugin-phpunit": "^0.18.3",  
 "vimeo/psalm": "^4.11 || ^5.0"  
 },  
 "type": "library",  
 "autoload": {  
 "psr-4": {  
 "Doctrine\\Common\\Lexer\\": "src"  
 }  
 },  
 "notification-url": "https://packagist.org/downloads/",  
 "license": [  
 "MIT"  
 ],  
 "authors": [  
 {  
 "name": "Guilherme Blanco",  
 "email": "guilhermeblanco@gmail.com"  
 },  
 {  
 "name": "Roman Borschel",  
 "email": "roman@code-factory.org"  
 },  
 {  
 "name": "Johannes Schmitt",  
 "email": "schmittjoh@gmail.com"  
 }  
 ],  
 "description": "PHP Doctrine Lexer parser library that can be used in Top-Down, Recursive Descent Parsers.",  
 "homepage": "https://www.doctrine-project.org/projects/lexer.html",  
 "keywords": [  
 "annotations",  
 "docblock",  
 "lexer",  
 "parser",  
 "php"  
 ],  
 "support": {  
 "issues": "https://github.com/doctrine/lexer/issues"

docker-compose.override.yml

version: '3'  
  
services:  
*###> doctrine/doctrine-bundle ###* database:  
 ports:  
 - "5432"  
*###< doctrine/doctrine-bundle ###  
  
###> symfony/mailer ###* mailer:  
 image: schickling/mailcatcher  
 ports: ["1025", "1080"]  
*###< symfony/mailer ###*

docker-compose.yml

version: '3'  
  
services:  
*###> doctrine/doctrine-bundle ###* database:  
 image: postgres:${POSTGRES\_VERSION:-15}-alpine  
 environment:  
 POSTGRES\_DB: ${POSTGRES\_DB:-app}  
 *# You should definitely change the password in production* POSTGRES\_PASSWORD: ${POSTGRES\_PASSWORD:-!ChangeMe!}  
 POSTGRES\_USER: ${POSTGRES\_USER:-app}  
 volumes:  
 - database\_data:/var/lib/postgresql/data:rw  
 *# You may use a bind-mounted host directory instead, so that it is harder to accidentally remove the volume and lose all your data!  
 # - ./docker/db/data:/var/lib/postgresql/data:rw  
###< doctrine/doctrine-bundle ###*volumes:  
*###> doctrine/doctrine-bundle ###* database\_data:  
*###< doctrine/doctrine-bundle ###*

phpunit.xml.dist

<?xml version="1.0" encoding="UTF-8"?>  
  
<!-- https://phpunit.readthedocs.io/en/latest/configuration.html -->  
<phpunit xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:noNamespaceSchemaLocation="vendor/phpunit/phpunit/phpunit.xsd"  
 backupGlobals="false"  
 colors="true"  
 bootstrap="tests/bootstrap.php"  
 convertDeprecationsToExceptions="false"  
>  
 <php>  
 <ini name="display\_errors" value="1" />  
 <ini name="error\_reporting" value="-1" />  
 <server name="APP\_ENV" value="test" force="true" />  
 <server name="SHELL\_VERBOSITY" value="-1" />  
 <server name="SYMFONY\_PHPUNIT\_REMOVE" value="" />  
 <server name="SYMFONY\_PHPUNIT\_VERSION" value="9.5" />  
 </php>  
  
 <testsuites>  
 <testsuite name="Project Test Suite">  
 <directory>tests</directory>  
 </testsuite>  
 </testsuites>  
  
 <coverage processUncoveredFiles="true">  
 <include>  
 <directory suffix=".php">src</directory>  
 </include>  
 </coverage>  
  
 <listeners>  
 <listener class="Symfony\Bridge\PhpUnit\SymfonyTestsListener" />  
 </listeners>  
  
 <!-- Run `composer require symfony/panther` before enabling this extension -->  
 <!--  
 <extensions>  
 <extension class="Symfony\Component\Panther\ServerExtension" />  
 </extensions>  
 -->  
</phpunit>

symfony.lock

{  
 "captcha-com/symfony-captcha-bundle": {  
 "version": "4.2.12"  
 },  
 "doctrine/annotations": {  
 "version": "1.14",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "1.10",  
 "ref": "64d8583af5ea57b7afa4aba4b159907f3a148b05"  
 }  
 },  
 "doctrine/doctrine-bundle": {  
 "version": "2.8",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "2.8",  
 "ref": "719d615af3150a59a02acc97c285d689e5dcf43f"  
 },  
 "files": [  
 "./config/packages/doctrine.yaml",  
 "./src/Entity/.gitignore",  
 "./src/Repository/.gitignore"  
 ]  
 },  
 "doctrine/doctrine-migrations-bundle": {  
 "version": "3.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "3.1",  
 "ref": "1d01ec03c6ecbd67c3375c5478c9a423ae5d6a33"  
 },  
 "files": [  
 "./config/packages/doctrine\_migrations.yaml",  
 "./migrations/.gitignore"  
 ]  
 },  
 "google/recaptcha": {  
 "version": "1.3",  
 "recipe": {  
 "repo": "github.com/symfony/recipes-contrib",  
 "branch": "main",  
 "version": "1.1",  
 "ref": "e5a4aa21f2e98d7440ae9aab6b56e307f99dd084"  
 },  
 "files": [  
 "./config/packages/google\_recaptcha.yaml"  
 ]  
 },  
 "mhujer/breadcrumbs-bundle": {  
 "version": "1.5.8"  
 },  
 "phpunit/phpunit": {  
 "version": "9.6",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "9.3",  
 "ref": "a6249a6c4392e9169b87abf93225f7f9f59025e6"  
 },  
 "files": [  
 "./.env.test",  
 "./phpunit.xml.dist",  
 "./tests/bootstrap.php"  
 ]  
 },  
 "sensio/framework-extra-bundle": {  
 "version": "6.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "5.2",  
 "ref": "fb7e19da7f013d0d422fa9bce16f5c510e27609b"  
 },  
 "files": [  
 "./config/packages/sensio\_framework\_extra.yaml"  
 ]  
 },  
 "symfony/console": {  
 "version": "6.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "5.3",  
 "ref": "da0c8be8157600ad34f10ff0c9cc91232522e047"  
 },  
 "files": [  
 "./bin/console"  
 ]  
 },  
 "symfony/debug-bundle": {  
 "version": "6.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "5.3",  
 "ref": "5aa8aa48234c8eb6dbdd7b3cd5d791485d2cec4b"  
 },  
 "files": [  
 "./config/packages/debug.yaml"  
 ]  
 },  
 "symfony/flex": {  
 "version": "2.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "1.0",  
 "ref": "146251ae39e06a95be0fe3d13c807bcf3938b172"  
 },  
 "files": [  
 "./.env"  
 ]  
 },  
 "symfony/framework-bundle": {  
 "version": "6.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "6.2",  
 "ref": "af47254c5e4cd543e6af3e4508298ffebbdaddd3"  
 },  
 "files": [  
 "./config/packages/cache.yaml",  
 "./config/packages/framework.yaml",  
 "./config/preload.php",  
 "./config/routes/framework.yaml",  
 "./config/services.yaml",  
 "./public/index.php",  
 "./src/Controller/.gitignore",  
 "./src/Kernel.php"  
 ]  
 },  
 "symfony/maker-bundle": {  
 "version": "1.48",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "1.0",  
 "ref": "fadbfe33303a76e25cb63401050439aa9b1a9c7f"  
 }  
 },  
 "symfony/messenger": {  
 "version": "6.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "6.0",  
 "ref": "ba1ac4e919baba5644d31b57a3284d6ba12d52ee"  
 },  
 "files": [  
 "./config/packages/messenger.yaml"  
 ]  
 },  
 "symfony/monolog-bundle": {  
 "version": "3.8",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "3.7",  
 "ref": "213676c4ec929f046dfde5ea8e97625b81bc0578"  
 },  
 "files": [  
 "./config/packages/monolog.yaml"  
 ]  
 },  
 "symfony/notifier": {  
 "version": "6.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "5.0",  
 "ref": "c31585e252b32fe0e1f30b1f256af553f4a06eb9"  
 },  
 "files": [  
 "./config/packages/notifier.yaml"  
 ]  
 },  
 "symfony/phpunit-bridge": {  
 "version": "6.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "5.3",  
 "ref": "819d3d2ffa4590eba0b8f4f3e5e89415ee4e45c3"  
 },  
 "files": [  
 "./.env.test",  
 "./bin/phpunit",  
 "./phpunit.xml.dist",  
 "./tests/bootstrap.php"  
 ]  
 },  
 "symfony/routing": {  
 "version": "6.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "6.2",  
 "ref": "e0a11b4ccb8c9e70b574ff5ad3dfdcd41dec5aa6"  
 },  
 "files": [  
 "./config/packages/routing.yaml",  
 "./config/routes.yaml"  
 ]  
 },  
 "symfony/security-bundle": {  
 "version": "6.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "6.0",  
 "ref": "8a5b112826f7d3d5b07027f93786ae11a1c7de48"  
 },  
 "files": [  
 "./config/packages/security.yaml"  
 ]  
 },  
 "symfony/translation": {  
 "version": "6.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "5.3",  
 "ref": "da64f5a2b6d96f5dc24914517c0350a5f91dee43"  
 },  
 "files": [  
 "./config/packages/translation.yaml",  
 "./translations/.gitignore"  
 ]  
 },  
 "symfony/twig-bundle": {  
 "version": "6.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "5.4",  
 "ref": "bb2178c57eee79e6be0b297aa96fc0c0def81387"  
 },  
 "files": [  
 "./config/packages/twig.yaml",  
 "./templates/base.html.twig"  
 ]  
 },  
 "symfony/validator": {  
 "version": "6.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "5.3",  
 "ref": "c32cfd98f714894c4f128bb99aa2530c1227603c"  
 },  
 "files": [  
 "./config/packages/validator.yaml"  
 ]  
 },  
 "symfony/web-profiler-bundle": {  
 "version": "6.2",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "6.1",  
 "ref": "e42b3f0177df239add25373083a564e5ead4e13a"  
 },  
 "files": [  
 "./config/packages/web\_profiler.yaml",  
 "./config/routes/web\_profiler.yaml"  
 ]  
 },  
 "symfony/webapp-pack": {  
 "version": "1.1",  
 "recipe": {  
 "repo": "github.com/symfony/recipes",  
 "branch": "main",  
 "version": "1.0",  
 "ref": "aece95c8a188f6e6d04f01ccb8678d1764fd2642"  
 },  
 "files": [  
 "./config/packages/messenger.yaml"  
 ]  
 },  
 "twig/extra-bundle": {  
 "version": "v3.5.1"  
 },  
 "victor-prdh/recaptcha-bundle": {  
 "version": "1.6",  
 "recipe": {  
 "repo": "github.com/symfony/recipes-contrib",  
 "branch": "main",  
 "version": "1.5",  
 "ref": "f604d862b3ce4434d131a0e9b3a2eb7bb1784aff"  
 },  
 "files": [  
 "./config/packages/recaptcha.yaml"  
 ]  
 }  
}