Чупеев А.Д. Автоматизация среды и создание административного интерфейса системы для сбора отзывов на учебные курсы: проект «ОТЗЫВУС» // Международный журнал информационных технологий и энергоэффективности.– 2025. – Т. 10 № 3(53) с. 142–148

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.3:37

АВТОМАТИЗАЦИЯ СРЕДЫ И СОЗДАНИЕ АДМИНИСТРАТИВНОГО ИНТЕРФЕЙСА СИСТЕМЫ ДЛЯ СБОРА ОТЗЫВОВ НА УЧЕБНЫЕ КУРСЫ: ПРОЕКТ «ОТЗЫВУС»

Чупеев А.Д.
ФГБОУ ВО "ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ", Тюмень, Россия (625000, Тюменская область, город Тюмень, ул. Володарского, д. 38), e-mail: cenz1217@gmail.com

В статье рассматривается модернизация информационной системы «Отзывус», направленной на сбор и анализ отзывов студентов на элективные и основные учебные курсы. Основной задачей проекта было расширение функционала системы для автоматического обновления данных о курсах и повышения удобства взаимодействия пользователей с системой. Проведена, автоматизация среды разработки с использованием Docker, а также разработан административный интерфейс для управления учебными подразделениями и курсами. В результате данных изменений удалось существенно повысить функциональность системы, улучшить её производительность и удобство использования.

Ключевые слова: Информационная система, учебные курсы, обратная связь, автоматизация.

AUTOMATING THE ENVIRONMENT AND CREATING AN ADMINISTRATIVE INTERFACE OF THE SYSTEM FOR COLLECTING FEEDBACK ON TRAINING COURSES: THE “OTZYVUS” PROJECT

Chupeev A.D.
TYUMEN INDUSTRIAL UNIVERSITY, Tyumen, Russia (625000, Tyumen Region, Tyumen, Volodarskogo St., 38), e-mail: cenz1217@gmail.com

The article deals with the modernization of the information system “Otzavus”, aimed at collecting and analyzing student feedback on elective and core courses. The main task of the project was to expand the functionality of the system to automatically update the data on courses and improve the convenience of user interaction with the system. The development environment was automated using Docker, and an administrative interface was developed for managing academic departments and courses. As a result of these changes it was possible to significantly increase the functionality of the system, improve its performance and usability.

Keywords: Information system, educational courses, feedback, automation.

Введение

Информационные системы для сбора отзывов студентов играют важную роль в совершенствовании образовательных программ и повышении качества преподавания. Проект «Отзывус» изначально был разработан для сбора отзывов на элективные курсы, но возникла необходимость расширения функциональности системы для обработки отзывов как по элективным, так и по основным курсам. Это расширение привело к модернизации архитектуры базы данных и внедрению новых возможностей для автоматизации процесса обновления данных о курсах.

Задачи проекта включали:

1. Внедрение автоматизированной среды разработки и тестирования с использованием Docker.
2. Создание административного интерфейса для управления учебными подразделениями и курсами.

Создание конфигурации автоматизированной среды разработки и тестирования

Изначально проект не включал в себя использование Docker. Чтобы развернуть проект локально, требовалось настраивать конфигурационные файлы PHP, прописывать 14 команд для установки зависимостей, настраивать .env файл, создавать базу данных и прописывать команду миграций. При осуществлении этого процесса заказчик принял решение заказать интегра

цию Docker [1,7]. Цель заключается в настройке автоматизированной среды разработки и тестирования с использованием Docker [1,7]. Главной задачей является создание среды для эффективной разработки и реализации тестов, направленных на проверку успешной загрузки страниц, стабильности сортировок и корректной работы пагинации в разделах "Элективы" и "Отзывы".

Для конфигурации Docker-среды были разработаны Docker-контейнеры, включающий в себя все необходимые зависимости, библиотеки и сервисы для успешного локального развертывания и тестирования веб-приложения. Эти контейнеры могут быть воспроизведены на любой совместимой с Docker платформе. Кроме того, были предусмотрены задокументированные инструкции по настройке и запуску Docker-контейнеров, что обеспечивает ясность воспроизведения окружения для разработчиков. Конфигурация Docker-среды включала создание основного образа с поддержкой PHP и необходимых расширений, а также для веб-приложения с настроенным веб-сервером и базой данных. Это обеспечило легкость воспроизведения окружения на различных платформах и стабильность в процессе разработки и тестирования. Дополнительно был создан контейнер в котором был механизм логирования для регистрации ошибок, предупреждений и других событий в CI/CD логах.

Реализация Dockerfile.dev:

Таблица 1 - Реализация Dockerfile.dev

Шаг	Описание
FROM php:8.2-cli-alpine AS final	Использует официальный образ PHP версии 8.2 с Alpine Linux в качестве базового образа для конечного образа.
RUN apk --no-cache add ...	Устанавливает необходимые зависимости, такие как git, zip, unzip, libpng-dev, и другие, используя apk.
RUN docker-php-ext-install ...	Устанавливает расширения PHP, такие как mbstring, exif, pcntl, bcmath, gd.
RUN curl -sS https://getcomposer.org/installer ...	Устанавливает Composer глобально.
WORKDIR /otzyvus/web/	Копирует все файлы из текущего контекста

	(локальной папки) в /otzyvus/web/ внутри контейнера.
RUN composer install --ignore-platform-reqs	Устанавливает зависимости Composer, игнорируя платформенные требования.
RUN npm install	Устанавливает зависимости Node.js с помощью npm.
RUN php artisan key:generate	Генерирует ключ приложения для Laravel.
RUN php artisan migrate	Выполняет миграции базы данных.
CMD npm run dev -- --host=0.0.0.0 --port=5173 & php artisan serve --host=0.0.0.0 --port=8000	Команда для запуска npm и веб-сервера с приложением внутри контейнера.

Реализация docker-compose.dev.yml:

Таблица 2 - Реализация docker-compose.dev.yml

Сервис	Описание
app	Контейнер с веб-приложением otzyvus-app и настроенным веб-сервером и базой данных
build	Контекст: ./web, Dockerfile: Dockerfile.dev
ports	Порты: 8000 для веб-приложения, 5173 для механизма логирования
volumes	Привязка локальной папки ./web к контейнеру
environment	Переменные окружения: APP_ENV=local, DB_CONNECTION=sqlite

Аспект тестирования включал написание тестов на PHP, охватывающих успешную загрузку страниц, обработку ошибок и предупреждений, а также проверку стабильности сортировок и пагинаций для элективов и отзывов. Тесты были интегрированы в автоматизированный режим в рамках процесса CI/CD, обеспечивая регулярное тестирование функциональности приложения.

Тестирование автоматизированной среды разработки:

Таблица 3 - Тестирование автоматизированной среды разработки

Название теста	Описание
testCallback	Тестирование обработчика обратного вызова, проверяющего создание и сохранение пользователя
testEditFeedback	Тестирование редактирования отзыва, проверяющее существование и отсутствие отзывов
testUpdateFeedback	Тестирование обновления отзыва, проверяющее существование и отсутствие отзывов
testDeleteFeedback	Тестирование удаления отзыва, проверяющее существование и отсутствие отзывов
testRestoreFeedback	Тестирование восстановления отзыва, проверяющее существование и отсутствие отзывов

Документационная работа включала в себя описание постановки задачи, используемых технологий и результатов разработки. Также предоставлено подробное руководство по конфигурации Docker-среды с описанием созданных тестов. Это позволило ясно представить контекст разработки и облегчило интеграцию компонентов в проект.

Разработка интерфейса для конфигурации подразделений университета

Администратор “Отзывус” может изменить информацию об подразделений ВУЗа только импортируя файл в котором содержиться данные о подразделениях, при этом информация которая есть в БД удаляется и заменяется данными из файла, это увеличивает потенциальный риск удаления данных, что не удовлетворяет заказчика. Поэтому необходимо спроектировать и разработать пользовательский интерфейс для страницы конфигурации подразделений университета. Интерфейс должен обеспечивать функциональность поиска, добавления, редактирования и удаления подразделений. Также должна быть предоставлена функция импорта данных подразделений из файла. Выбор технологий обусловлен стеком разработки “Отзывус”: Laravel – фреймворк PHP [3,4], Blade - шаблонизатор, встроенный в Laravel [3,4] , Livewire – инструмент для создания интерактивных пользовательских интерфейсов на основе PHP [3,4].

В админ-панели администратор имеет возможность управлять подразделениями. Он может добавить новое подразделение, предоставив полное и краткое название. При необходимости пользователь может отредактировать название подразделения, нажав на кнопку "редактировать". В случае удаления подразделения пользователю будет предложено выбрать, куда перевести всех привязанных к этому подразделению пользователей, что обеспечивает более гибкое и безопасное удаление. Кроме того, оставлена возможность импорта подразделений через файл (Рисунок 1).

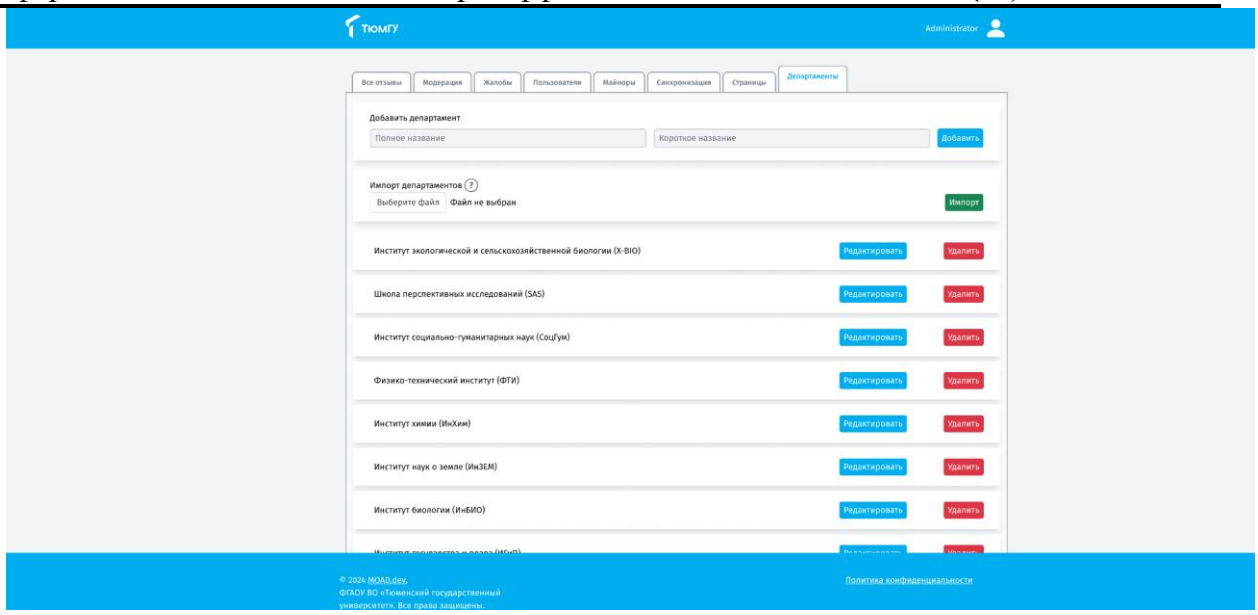

Рисунок 1 - Интерфейс

Был реализован DepartmentController.php. Этот компонент переносит действия пользователя “Отзывус” на БД. Он реализует методы описанные в таблице 4

Таблица 4 – Методы реализации DepartmentController.php.

Название метода	**Описание метода**
index	Метод для отображения списка подразделений. Загружает данные из базы данных и передает их в представление.
update	Обрабатывает запрос на обновление данных конкретного подразделения в базе данных.
edit	Отображает форму редактирования информации о подразделении.
delete	Удаляет выбранное подразделение из базы данных.
store	Обрабатывает запрос на создание нового подразделения и сохраняет его данные в базе.

Следующем этапом была реализация Livewire компонентов - ShowDepartment.php и EditDepartment.php. Эти компоненты реализует метод render, который отвечает за отображение соответствующего представления (view) Livewire, передавая данные для отображения в этих представлениях.

Далее следовало создание Blade-представлений. departments-edit.blade.php - представление для отображения формы редактирования информации о подразделении. departments.blade.php - Представление для отображения списка всех подразделений. Эти представления содержат разметку HTML, которая будет использоваться для отображения данных, а также формы для редактирования или удаления подразделений.

Последний шаг создание Livewire представлений - show-departments.blade.php и edit-department.blade.php: они используются для отображения данных и обработки действий пользователей, которые могут быть вызваны во время использования компонентов Livewire.

Каждый из этих шагов представляет часть функционала для управления подразделениями. Контроллер отвечает за маршрутизацию запросов и обработку действий пользователя. Livewire компоненты позволяют создавать динамические интерфейсы без использования JavaScript, а Blade-представления отображают данные и обеспечивают пользовательский интерфейс для управления подразделениями.

Заключение

В рамках проекта по модернизации информационной системы «Отзывус», нацеленного на сбор отзывов на различные типы учебных курсов, было выполнено следующее:

1. Создана и настроена конфигурация автоматизированной среды разработки и тестирования Docker, предназначенная для стандартизации процессов разработки и тестирования.
2. Разработан интерфейс для конфигурации подразделений университета, позволяющий управлять подразделениями университета.

Перспективы развития включают усовершенствование пользовательского интерфейса и расширение функциональности сервиса для улучшения взаимодействия пользователей с системой. Эти шаги направлены на обеспечение более эффективного управления учебным процессом и оптимизацию работы сервиса.

Список литературы

1. Буренков, И. А. Применение виртуальных контейнеров Docker для запуска сервисов: [Электронный ресурс]. CyberLeninka. URL: https://cyberleninka.ru/article/n/primenenie-virtualnyh-konteynerov-docker-dlya-zapuska-servisov (дата обращения: 22.11.2023).
2. PHP Manual: [Электронный ресурс]. URL: https://www.php.net/manual/en/index.php (дата обращения: 02.11.2023).
3. Laravel - The PHP Framework For Web Artisans: [Электронный ресурс] // Laravel : [сайт]. URL: https://laravel.com/docs/10.x (дата обращения: 02.11.2023).
4. Стаффер, М. Laravel. Полное руководство. Санкт-Петербург: Питер, 2020. 512 с.: ил. ISBN 978-5-4461-1396-5.
5. PhpMyAdmin: [Электронный ресурс] // PhpMyAdmin: [сайт]. URL: https://www.phpmyadmin.net (дата обращения: 02.11.2023).
6. Миграции в Laravel: [Электронный ресурс] // Laravel: [сайт]. URL: https://laravel.com/docs/10.x/migrations (дата обращения: 02.11.2023).
7. Эдриен Моуэт. Использование Docker (2017): [Электронный ресурс]. URL: https://vtome.ru/knigi/programming/534325-ispolzovanie-docker.html (дата обращения: 02.11.2023).

References

1. Burenkov, I.A. Application of Docker virtual containers for launching services: [Electronic resource]. CyberLeninka. URL: https://cyberleninka.ru/article/n/primenenie-virtualnyh-konteynerov-docker-dlya-zapuska-servisov (accessed: 22.11.2023).

2. PHP Manual: [Electronic resource]. URL: https://www.php.net/manual/en/index.php (accessed: 02.11.2023).
3. Laravel - The PHP Framework For Web Artisans: [Electronic resource] // Laravel: [website]. URL: https://laravel.com/docs/10.x (accessed: 02.11.2023).
4. Stauffer, M. Laravel: The Definitive Guide. St. Petersburg: Piter, 2020. 512 p.: ill. ISBN 978-5-4461-1396-5.
5. PhpMyAdmin: [Electronic resource] // PhpMyAdmin: [website]. URL: https://www.phpmyadmin.net (accessed: 02.11.2023).
6. Migrations in Laravel: [Electronic resource] // Laravel: [website]. URL: https://laravel.com/docs/10.x/migrations (accessed: 02.11.2023).
7. Mouat A. Using Docker (2017): [Electronic resource]. URL: https://vtome.ru/knigi/programming/534325-ispolzovanie-docker.html (accessed: 02.11.2023).