

*А. А. Лубышев<sup>1</sup>✉, М. С. Зайцев<sup>1</sup>, Р. В. Гришин<sup>1</sup>*

## **Разработка системы управления контентом с целью создания веб-сайта, для организации АО «Балт АГП»**

<sup>1</sup>Сибирский государственный университет геосистем и технологий,  
г. Новосибирск, Российская Федерация  
e-mail: lubyshev.alexandr@ya.ru

**Аннотация.** В условиях стремительного развития цифровых технологий и роста запросов к качеству онлайн-представительства, наличие современного, многофункционального и удобного веб-сайта становится значимой составляющей имиджа каждой организации. Особенно актуально это для компаний, взаимодействующих с широкой аудиторией и предоставляющих информацию о своей деятельности посредством интернета. Компания «Балт АГП» столкнулась с проблемой отсутствия эффективной системы управления контентом на сайте, что затрудняло обновление информации, ведение новостей и актуализацию данных. Трудности в обновлении информации, ограниченная гибкость существующего решения и устаревший пользовательский интерфейс уменьшали скорость реакции на внешние изменения и затрудняли коммуникацию с посетителями сайта и партнерами. Все это могло отрицательно отражаться на восприятии компании и снижать доверие со стороны целевой аудитории. В связи с этим была разработана административная панель, позволяющая сотрудникам компании легко управлять контентом сайта без необходимости в технических знаниях. Новая система реализована с использованием новейших технологий, что обеспечивает высокую производительность, расширяемость и гибкость проекта. Кроме того, была создана современная публичная часть сайта с адаптивным дизайном, улучшенной навигацией и расширенными возможностями взаимодействия с пользователями. В совокупности это позволило обеспечить удобный и эффективный инструмент для поддержки и развития онлайн-присутствия компании «Балт АГП».

**Ключевые слова:** Vue 3, CMS, UI/UX, Java, REST API, интеграция frontend и backend

*A. A. Lubyshev<sup>1</sup>✉, M. S. Zaitsev<sup>1</sup>, R. V. Grishin<sup>1</sup>*

## **Development of a content management system to create a website for the organization "Balt AGP"**

<sup>1</sup>Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation  
e-mail: lubyshev.alexandr@ya.ru

**Annotation.** With the rapid development of digital technologies and growing demands for quality online representation, having a modern, multifunctional and user-friendly website is becoming a significant component of every organization's image. This is especially true for companies that interact with a wide audience and provide information about their activities via the Internet. The company "Battle AGP" faced a number of problems associated with the lack of an effective content management system. Difficulty in updating information, limited flexibility of the existing solution and outdated user interface reduced the speed of reaction to external changes and hindered communication with site visitors and partners. All of this could negatively affect the perception of the company and reduce the trust of the target audience. In this regard, an administrative panel was developed, allowing the company's employees to easily manage the content of the site without the need for technical knowledge. The new system was implemented using the latest technologies, which ensures high

performance, extensibility and flexibility of the project. In addition, a modern public part of the site was created with adaptive design, improved navigation and enhanced user interaction capabilities. Together, this provided a convenient and effective tool to support and develop Battle AGP's online presence.

**Keywords:** Vue 3, CMS, UI/UX, Java, REST API, frontend and backend integration

### ***Введение***

В современном мире каждой компании, ориентированной на клиентов, необходим современный сайт – удобный, эстетичный и с возможностью быстрого и простого редактирования контента. Перед нами была поставлена задача – обеспечить возможность публикации и редактирования информации о компании, управлять каталогом продукции, публиковать новости, обрабатывать заказы и отзывы, а также взаимодействовать с пользователями через формы обратной связи. 1С-Битрикс, WordPress и другие традиционные CMS ориентированы на широкий спектр задач, но при этом часто избыточны и сложны для гибкой кастомизации. Например, внесение изменений в структуру системы может привести к сбоям и конфликтам плагинов, а также в этих системах управления имеется высокая зависимость от сторонних расширений, что усиливает риск уязвимостей и неустойчивой работы. Цель данной работы – разработка системы управления контентом для создания современного и функционального веб-сайта для АО «Балт АГП».

Для этого была реализована CMS-панель, доступная администраторам, и публичный сайт, доступный пользователям. Приложение разработано как монолитная система, использующая современные технологии: Spring Boot, PostgreSQL, Kafka, Redis, JWT и Docker, а для frontend-составляющей: Vue 3, Vite, Tailwind CSS + SCSS, Axios. Благодаря использованию брокера Kafka обеспечена асинхронная обработка email-уведомлений, а Redis применяется для хранения токенов и управления сессиями, а с помощью Vue 3, появляется возможность создавать динамические компоненты, с целью переиспользования, что существенно упрощает работу с кодом. Безопасность доступа к административным данным обеспечивается с помощью Spring Security и механизма авторизации на основе JWT.

### ***Методы и материалы***

Для обеспечения полноценной работы сайта требуется создание как клиентской части (frontend), так и серверной части (backend) приложения.

Первое, с чего была начата разработка – это backend. Проектирование серверной части приложения началось с разработки многослойной архитектуры, позволяющей разделить ответственность между компонентами [1–3]. Основу составляет контроллерный слой, обеспечивающий обработку HTTP-запросов, сервисный слой, реализующий бизнес-логику, и слой доступа к данным, взаимодействующий с базой данных через JPA (рис. 1).

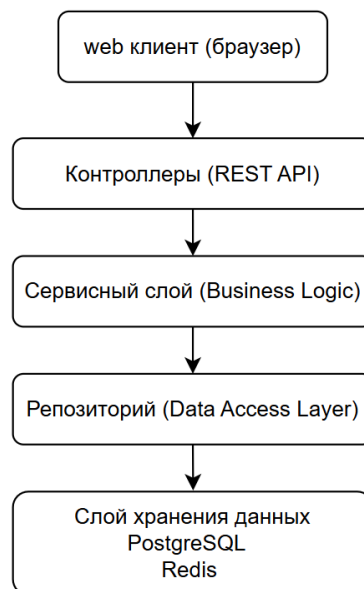


Рис. 1. Слои архитектуры приложения

Важной частью архитектурного решения стала система безопасности. Для защиты ресурсов и разграничения прав доступа используется Spring Security в связке с JWT-токенами. Это обеспечивает stateless аутентификацию: после входа пользователь получает access и refresh токены, а дальнейшие запросы проходят верификацию по заголовкам. Все JWT-токены хранятся в Redis, что позволяет быстро их проверять и отзывать при необходимости (рис. 2).

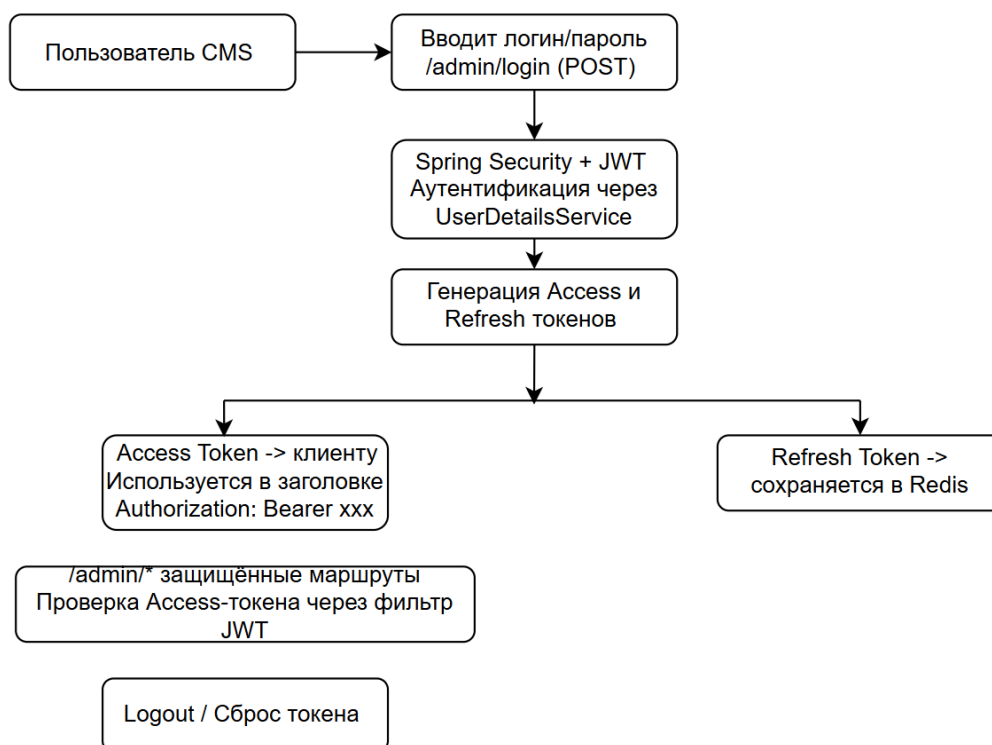


Рис. 2. Механизм авторизации и работы с токенами

Проектирование API осуществлялось с учетом принципов REST и разделения ролей. Все маршруты были логически разделены на публичную часть (/pub/) и административную (/admin/). Например, данные о продуктах и новостях доступны через публичные эндпоинты, а управление ими – только через защищенные административные маршруты. Для описания и тестирования API использовался OpenAPI с интеграцией SpringDoc.

Основу модели данных составляет набор сущностей, отражающих ключевые бизнес-объекты сайта: продукты, категории, новости, заказы, отзывы, пользователи и настройки сайта. Связи между ними представлены на ER-диаграмме основных сущностей (рис. 3), что позволило эффективно спроектировать реляционную схему и обеспечить целостность данных на уровне БД.

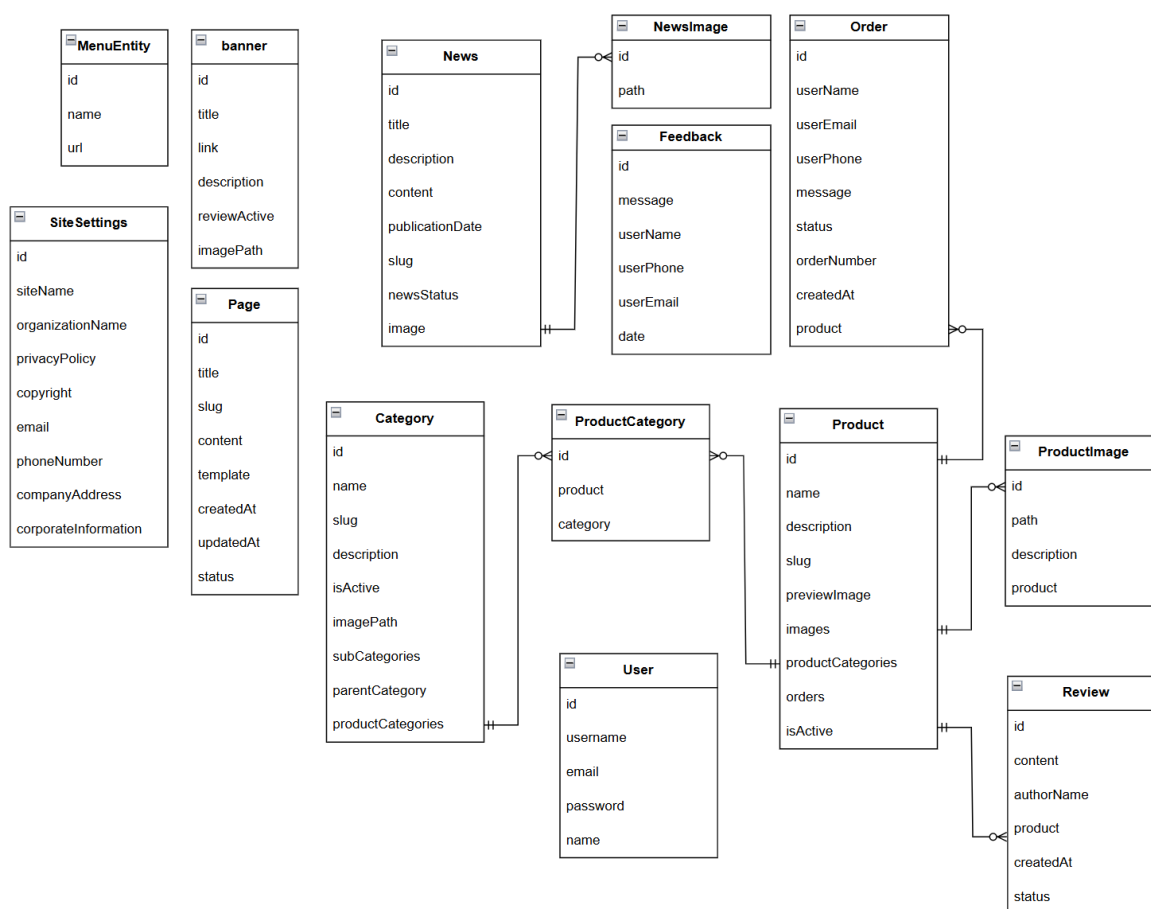


Рис. 3 ER–диаграмма основных сущностей

Реализация серверной части велась на базе Spring Boot с использованием Spring Data JPA, Security, Redis, Kafka и других модулей. Приложение реализовано как монолит, при этом для асинхронных задач, таких как email уведомления о заказах, используется Kafka как брокер сообщений. Благодаря этому удалось разделить ответственные процессы, повысив отказоустойчивость системы.

Функциональность CMS реализована через административную панель, работающую на эндпоинтах с префиксом /admin/. Через нее администратор сайта

может управлять продуктами, категориями, новостями, отзывами и настройками сайта. Также предусмотрены инструменты для просмотра и обработки заказов и обращений пользователей через формы обратной связи.

Особое внимание было уделено безопасности административных функций. Аутентификация пользователей реализована через JWT и систему ролей. Только авторизованные пользователи с правами администратора могут обращаться к маршрутам `/admin/**`. Все токены валидируются через Redis, что позволяет легко реализовать `logout` и блокировку доступа.

Для отображения данных и управления сущностями в CMS используются REST-контроллеры с четко структурированными DTO и мапперами на базе MapStruct. Это позволило отделить внутреннюю модель от внешнего представления и упростить поддержку проекта.

Таким образом, реализованная серверная часть обеспечивает полную поддержку функциональности сайта и административной панели, сохраняя модульность и соответствие современным требованиям к веб-приложениям.

Для реализации клиентской части сайта был выбран фреймворк Vue 3 [4]. Благодаря компонентам Vue, интерфейс разбит на независимые составляющие (страницы, карточки, таблицы, формы, модальные окна и т.д.), каждая из которых отвечает за свою функциональность и визуальное отображение. Это упрощает сопровождение и повторное использование кода. С помощью такого мощного функционала фреймворка, было создано дерево компонентов и исполнительного кода, позволяющее эффективно работать клиентской части (рис. 4)

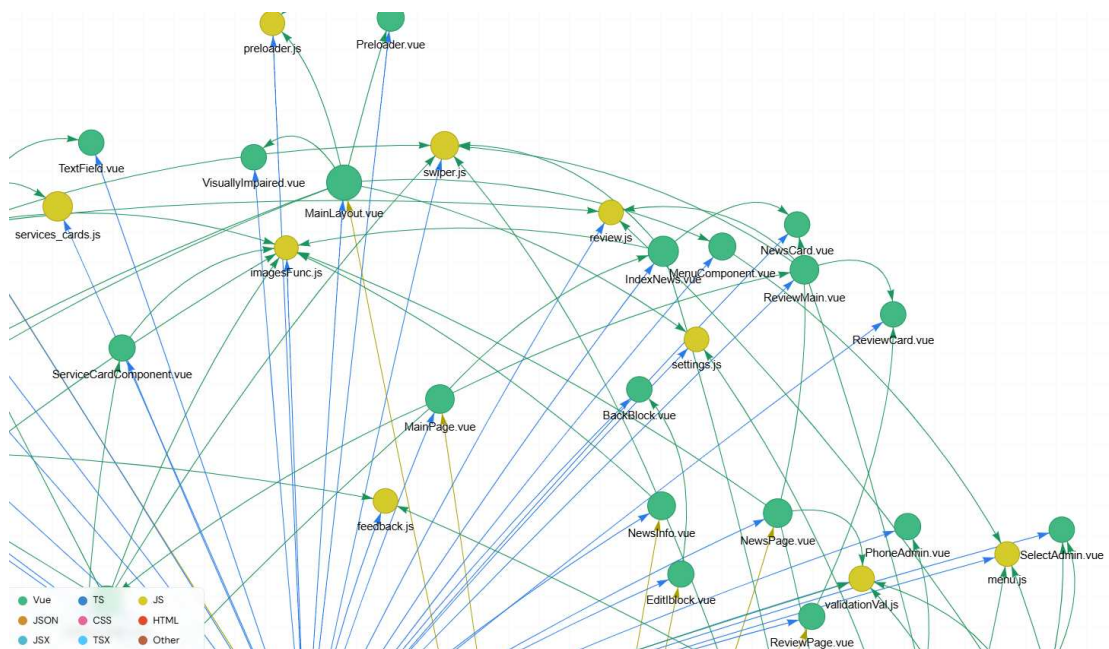


Рис. 4. Дерево компонентов и исполнительного кода

В качестве системы маршрутизации использован Vue Router, который обеспечивает навигацию между страницами без перезагрузки, что значительно повы-

шает отзывчивость интерфейса и удобство использования. Для стилизации применяются Tailwind CSS и SCSS. Такой подход позволил комбинировать утилитарные классы и кастомные стили, обеспечив быструю адаптацию интерфейса под дизайн-макет.

Дизайн сайта был разработан предварительно, а на его основе реализована адаптивная и семантически корректная верстка, соответствующая современным требованиям к UI/UX [5, 6]. Компоненты интерфейса реализованы с учетом повторного использования и возможности масштабирования проекта в будущем.

Обмен данными с серверной частью и клиентской выполняются с помощью Axios. Все запросы к API помещаются в json и отправляются на сервер в отдельные сервисы, что упрощает управление взаимодействием с backend-логикой и облегчает поддержку и масштабируемость проекта. Также реализована обработка ошибок, загрузок и статусов, что улучшает пользовательский опыт при действиях в приложениях.

### ***Результаты***

Исходя из проделанной работы можно видеть результат, что для посетителя сайта реализовано все что нужно, для полноценной работы. Пользователь может почитать информацию о компании, посмотреть список товаров и выбрать нужный по категории, посмотреть новости, оформить товар и оставить на него отзыв. Интерфейс получился простой и интуитивно понятный, все действия, которые выполняет пользователь обрабатываются на сервере, то есть обработка заказов и отправка отзывов. При этом отзывы на товар проходят модерацию перед публикацией.

Также была настроена авторизация с помощью JWT токена, для cms, это нужно для предоставления прав пользователю и контроля сессий. Механизм устроен так: администратор вводит логин и пароль, далее отправляется запрос на сервер, и если данные верны, то система выдает два токена, один для предоставления прав в закрытые участки сайта, второй нужен для его обновления. Механизм помимо тестов на самом сайте был протестирован в Postman, на рисунке 6 можно увидеть отправленный post запрос на /login, и что возвращается в ответ, это подтверждает, что механизм работает верно.

Далее, результат интерфейса и внешнего вида. Сайт сделан визуально приятно и интуитивно понятно как для пользователей, так и для администрации. В cms можно без проблем добавлять, редактировать и удалять нужный контент: новости, продукцию и т.д. Также имея права к панели администратора можно создавать собственные страницы с любым содержимым без знания программирования, с помощью реализованного функционала через интерфейс. Все это в сумме делает систему удобной и функциональной

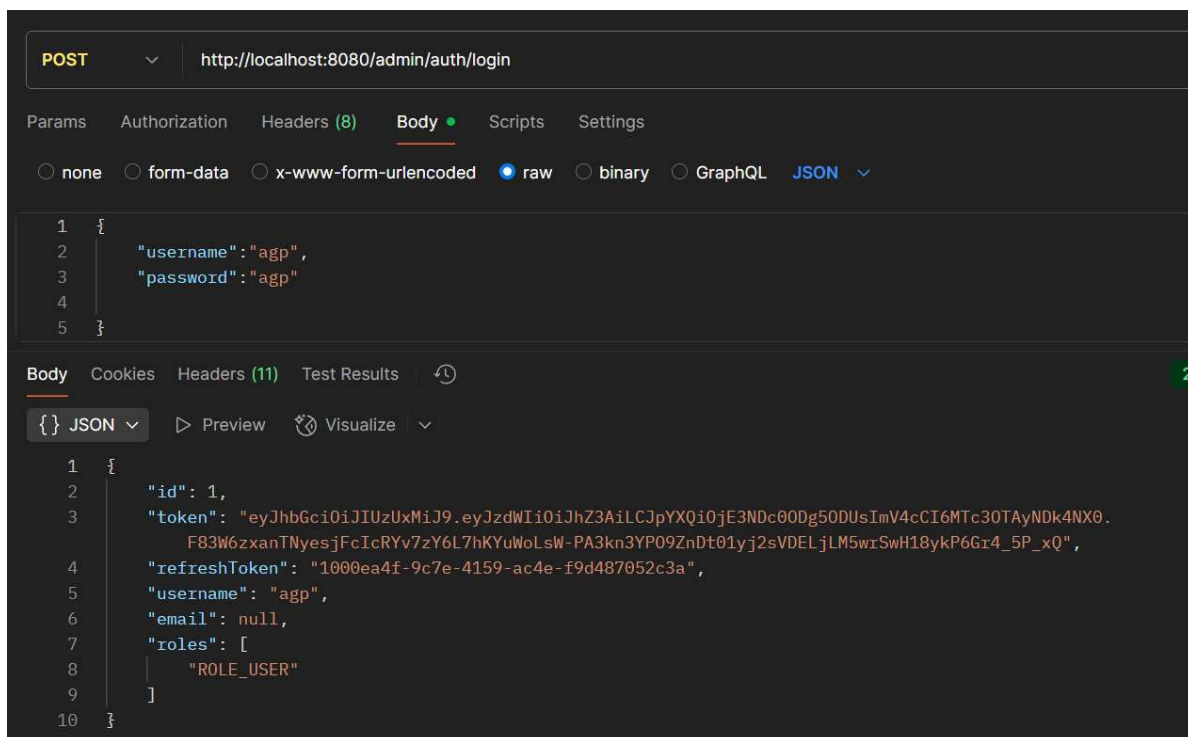


Рис. 6. Пример запроса на /login в Postman

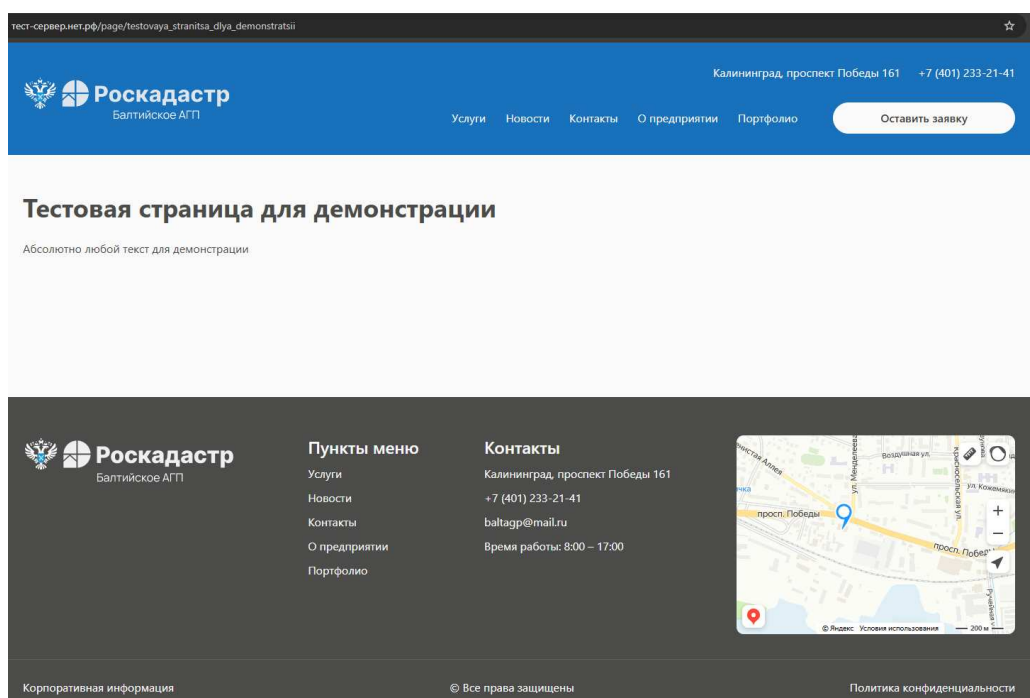


Рис. 7. Создание пользовательской страницы

## Заключение

В результате работы, нами была создана собственная CMS, которая позволяет создавать, изменять, и удалять контент. Frontend-часть без ошибок работает с backend, а также компоненты приложения соответствуют требованиям бренд-

бука компании. Более того, приложение было интегрировано на сервер АО «Балт АГП», и сейчас проходит этап заполнения контента.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Spring Boot – официальная документация. – URL: <https://docs.spring.io/spring-boot/documentation.html> (дата обращения 17.04.2025).
2. PostgreSQL – официальная документация. – URL: <https://www.postgresql.org/docs/current/index.html> (дата обращения 15.02.2025).
3. Apache Kafka – официальная документация. – URL: <https://kafka.apache.org/documentation> (дата обращения 21.02.2025).
4. Vue.js – официальная документация. – URL: <https://vuejs.org> (дата обращения 10.02.2025).
5. Tailwind CSS – официальная документация. – URL: <https://tailwindcss.com> (дата обращения 01.02.2025).
6. Tilda Education – как передавать макет из Figma в Tilda – URL: <https://tilda.education/articles-figma> (дата обращения 03.02.2025).
7. Vite – официальная документация. – URL: <https://vitejs.dev> (дата обращения 03.02.2025).
8. Spring Boot – официальная документация. – URL: <https://docs.spring.io/spring-boot/documentation.html> (дата обращения 17.04.2025).

© А. А. Лубышев, М. С. Зайцев, Р. В. Гришин, 2025