

# ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ АДМИНИСТРИРОВАНИЯ САЙТА ОРГАНИЗАЦИИ ООО «АРЛАЙТ РУС»

*Тумановский В.С.*

*Белорусский государственный университет информатики и радиоэлектроники,  
Институт информационных технологий,  
г. Минск, Республика Беларусь*

*Шведова О.А. – маг. техн. наук, ст. препод. каф. ИСиТ*

В рамках данного проекта было разработано веб-приложение для администрирования сайта организации ООО "Арлайт РУС". В работе также приведены описания современных методов разработки программного обеспечения и представлены применяемые технологии, использованные при создании данного приложения.

С ростом числа веб-сайтов и их сложности возникает потребность в эффективном управлении содержимым и функциональностью. Веб-приложение для администрирования сайта предоставляет инструменты, позволяющие администраторам сайта легко добавлять, редактировать и удалять контент, управлять пользователями, настраивать параметры сайта и многое другое. Это значительно упрощает процесс управления сайтом и позволяет оперативно вносить изменения без необходимости обращения к разработчикам.

Основные функций администрирования сайта, которые реализовываются с помощью веб-приложения:

- управление контентом: Приложение позволяет администраторам добавлять, редактировать и удалять различные типы контента на сайте, такие как статьи, новости, продукты, услуги и другие. Они могут изменять текст, изображения, видео и другие медиафайлы, чтобы обновлять информацию и поддерживать актуальность сайта;

- управление пользователями: Приложение позволяет администраторам управлять пользователями сайта. Они могут создавать новые учетные записи пользователей, устанавливать права доступа, управлять ролями и привилегиями. Это обеспечит контроль над тем, кто имеет доступ к различным разделам и функциям сайта;

- управление SEO-оптимизацией: Приложение может предоставлять инструменты для оптимизации сайта для поисковых систем (SEO). Администраторам доступны функции для настройки мета-тегов, URL-адресов, ключевых слов и других параметров, которые помогут улучшить видимость сайта в поисковых результатах;

– управление безопасностью: Приложение включает функции для обеспечения безопасности сайта, такие как управление доступом, защита от несанкционированного доступа, резервное копирование данных и другие меры безопасности.

ПО разработано с использованием клиент-серверной архитектуры - модель организации распределенных систем, в которой приложение делится на два основных компонента: клиент и сервер [1]. В результате проектирования и разработки программного средства:

- разработана схема алгоритма разграничения пользовательских прав доступа;
- разработана схема алгоритма добавления контента. На которой видно, как пользователь будет добавлять контент на администрируемый сайт;
- разработана схема алгоритма авторизации, на которой видно, что авторизация будет происходить при помощи JWT-токенов;
- построена физическая модель базы данных. В ходе построения были определены связи между таблицами и типы данных у полей;
- разработан прототип пользовательского интерфейса, где было определено расположение основных элементов пользовательского интерфейса на странице контента и странице добавления контента веб-приложения;
- разработана диаграмма развертывания, где была определена архитектура веб-приложения, составные части сервера: Nuxt.js, REST API Nuxt.js, СУБД MongoDB и как клиент будет взаимодействовать с сервером и показана на рисунке 1..

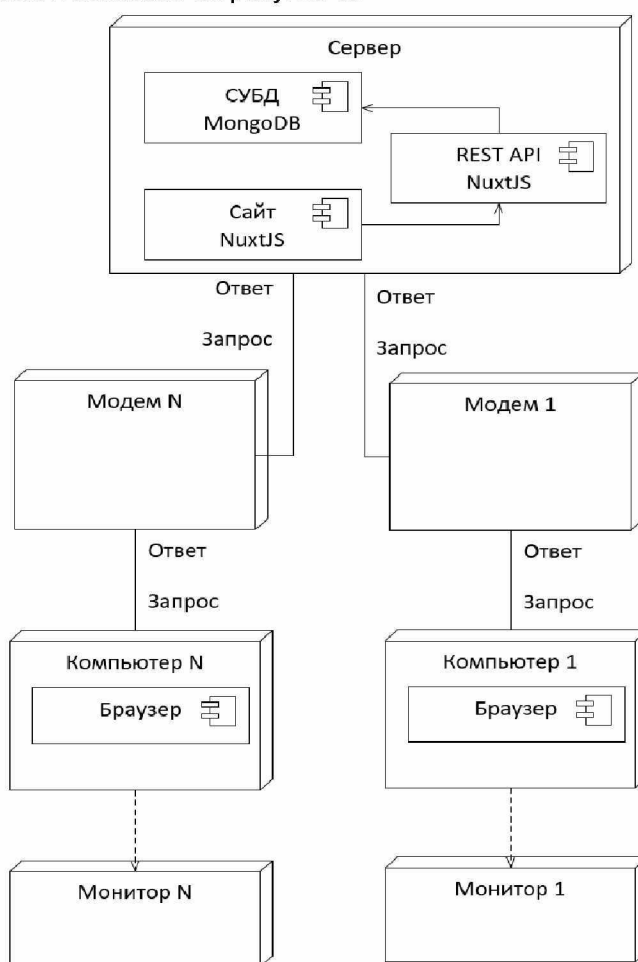


Рисунок 1 – Диаграмма развертывания

На диаграмме развертывания выделен блок сервер, который состоит из трех основных компонентов: Nuxt.js [2] REST API (Node.js [3]), СУБД MongoDB [4] и сайта Nuxt.js. Каждый из этих компонентов играет ключевую роль в функционировании веб-приложения.

Nuxt.js Rest API:

– этот компонент представляет собой серверную часть приложения, разработанную с использованием Nuxt.js, который обеспечивает взаимодействие между клиентской частью приложения и базой данных;

– Nuxt.js Rest API отвечает за обработку HTTP-запросов, аутентификацию и авторизацию пользователей, обработку бизнес-логики и предоставление данных клиентскому приложению;

– он взаимодействует с СУБД MongoDB для получения и сохранения данных.

СУБД MongoDB:

– СУБД MongoDB является основной базой данных для хранения информации, связанной с администрированием сайта;

– она обеспечивает надежное хранение данных, поддерживает операции чтения, записи и запросы к данным;

– Nuxt.js Rest API взаимодействует с СУБД MongoDB с использованием соответствующего драйвера или библиотеки для выполнения операций с данными;

Сайт Nuxt.js:

– сайт, разработанный с использованием фреймворка Nuxt.js, представляет собой клиентскую часть веб-приложения, которая обеспечивает интерфейс и пользовательский опыт для администраторов сайта;

– он взаимодействует с Nuxt.js Rest API для получения данных и отправки запросов на сервер;

– сайт Nuxt.js отображает информацию, позволяет администраторам управлять контентом, пользователями и настройками сайта.

Диаграмма развертывания демонстрирует, что серверный блок, состоящий из Nuxt.js Rest API и СУБД MongoDB, развертывается на серверной инфраструктуре, в то время как клиентская часть веб-приложения, сайт Nuxt.js, доступна для использования через веб-браузеры администраторов.

Веб-приложение для администрирования сайта способствует повышению эффективности работы. Оно предоставляет удобный пользовательский интерфейс, интуитивно понятные инструменты и функциональность, которая оптимизирована для выполнения задач администрирования сайта. Благодаря этому, администраторы могут быстро выполнять свои обязанности, управлять сайтом более эффективно и сосредотачиваться на стратегически важных задачах. Также оно обеспечивает безопасность и контроль доступа к административным функциям. С помощью такого приложения можно устанавливать различные уровни доступа для администраторов, определять их права и ограничения, а также отслеживать и контролировать действия пользователей. Это позволяет обеспечить безопасность сайта и предотвратить несанкционированный доступ к административным функциям, что особенно важно для сайтов, содержащих конфиденциальную информацию или обрабатывающих персональные данные.

**Список использованных источников:**

1. Hassan Goma'a. "Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures". Cambridge University Press, 2011.

2. Основные преимущества Nuxt.js – [Электронный ресурс]. – Режим доступа: <https://nuxtjs.org/>.

3. Документация Node.js – [Электронный ресурс]. – Режим доступа: <https://nodejs.org/docs/latest/api/>.

4. Документация Mongo DB – [Электронный ресурс]. – Режим доступа: <https://www.mongodb.com/docs/>.