

печного оборудования), воздухонагреватели и справочники (печи). В дальнейшем программа будет совершенствоваться, а её функциональные возможности расширяться.

Список использованных источников

1. Математическое моделирование металлургических процессов в АСУ ТП: учебное пособие / Н.А. Спирин, В.В. Лавров, В.Ю. Рыболовлев, Л.Ю. Гилева, А.В. Краснобаев, В.С. Швыдкий, О.П. Онорин, К.А. Щипанов, А.А. Бурыкин; под ред. Н.А. Спирина. – Екатеринбург: УрФУ, 2014. – 558 с. Зональная научная библиотека УрФУ: <http://elar.urfu.ru/handle/10995/27839>.

2. Программная реализация автоматизированного рабочего места «Технический отчет доменного цеха» на платформе ASP.NET CORE MVC / К.Р. Перетыкина, В.В. Лавров, И.А. Гурин, Н.А. Спирин // Системы автоматизации в образовании, науке и производстве. AS'2019: труды XII Всероссийской научно-практической конференции (с международным участием) (28–30 ноября 2019 г.) – Новокузнецк: СибГИУ, 2019. С. 161-164.

3. Разработка web-приложения автоматизированного рабочего места «Технический отчет доменного цеха» / К.Р. Перетыкина, В.В. Лавров, И.А. Гурин, Н.А. Спирин // Теплотехника и информатика в образовании, науке и производстве: сборник докладов VIII Всероссийской научно-практической конференции студентов, аспирантов и молодых учёных (ТИМ'2019) с международным участием (16–17 мая 2019 г.). – Екатеринбург: УрФУ, 2019. С. 286–290.

4. Совершенствование программного обеспечения АРМ «Технический отчет доменного цеха» при переходе на платформу ASP.NET MVC / Перетыкина К.Р., Лавров В.В., Гурин И.А., Спирин Н.А. // Актуальные проблемы современной науки, техники и образования: тезисы докладов 77-й международной научно-технической конференции. – Магнитогорск: МГТУ, 2019. Т. 1. – С. 343.

УДК 004.22

А. А. Першин, А. С. Блинков

ФГАОУ ВО «Уральский федеральный университет

имени первого Президента России Б.Н. Ельцина», г. Екатеринбург, Россия

РАЗРАБОТКА СИСТЕМЫ ВИЗУАЛИЗАЦИИ ДАННЫХ С ГЕОРАСПРЕДЕЛЕННЫХ ДАТЧИКОВ

Аннотация. Представлено описание информационной системы визуализации, основной функцией которой является отображение данных, полученных из базы данных. Основными входными данными системы являются: коллекция значений температуры, влажности, даты и времени, снятых с датчиков, установленных по месту. Отражены основные этапы

разработки программного обеспечения: постановка задачи, разработка структурной схемы, разработка архитектуры веб-приложения.

Ключевые слова: система визуализации, датчики, программирование, веб-приложение, интерфейс, карта.

Abstract. A description of the information visualization system, whose main function is to display data obtained. The main input data of the system is: a collection of temperature, humidity, date and time values taken from sensors installed locally. The main stages of software development are reflected: problem statement, development of the block diagram, development of the architecture of the web application.

Key words: visualization system, sensors, programming, web application, interface, map.

Введение. Тенденция развития современных информационных технологий характеризуется переходом к облачным вычислениям и хранению данных в облачных сетевых хранилищах. Под облачными вычислениями понимаются как приложения, предоставляемые в виде услуг через Интернет, так и аппаратное и системное программное обеспечение в центрах обработки данных, которые предоставляют эти услуги. Основная идея облачных вычислений – это предоставление потребителям готовой распределенной инфраструктуры, которая является прозрачной для выполняющихся приложений. Таким образом, при разработке облачных приложений можно игнорировать наиболее сложные для имплементации моменты, связанные с организацией распределенных вычислений, и вместо этого использовать высокоуровневые программные интерфейсы. Другая особенность облачной архитектуры – эластичность, то есть возможность быстрого увеличения вычислительных мощностей, например, при увеличении нагрузки на систему. При этом подобные манипуляции автоматизированы с помощью системы управления облаком, а в некоторых случаях могут вообще происходить без вмешательства человека.

Основными компонентами системы визуализации данных с геораспределенных датчиков являются: датчики сбора данных, контроллеры обработки информации на серверной части и система визуализации данных. Архитектура системы визуализации представлена на рисунке 1.

Рассмотрим вопрос построения системы с точки зрения разработки полноценного веб-приложения, использующего на Frontend JavaScript реактивный фреймворк Vue.js и php в качестве Backend серверной части.

Язык программирования JavaScript предназначен для написания сценариев для активных HTML-страниц. Программа на JavaScript встраивается непосредственно в исходный текст HTML-документа и интерпретируется браузером по мере загрузки этого документа. С помощью JavaScript можно динамически изменять текст загружаемого HTML-документа и реагировать на события, связанные с действиями посетителя или изменениями состояния документа или окна.

Vue.js – это фреймворк, который используется для решения задач именно уровня представления, его просто интегрировать с другими библиотеками и проектами. Иными словами, Vue.js – это инструмент, который можно внедрять постепенно. Возможность внедрения связана с тем, что Vue.js стремится к прогрессивности: поддержку Vue можно добавить в уже существующий проект,

благодаря чему его функциональность будет значительно расширена. И это отличает его от других фреймворков.

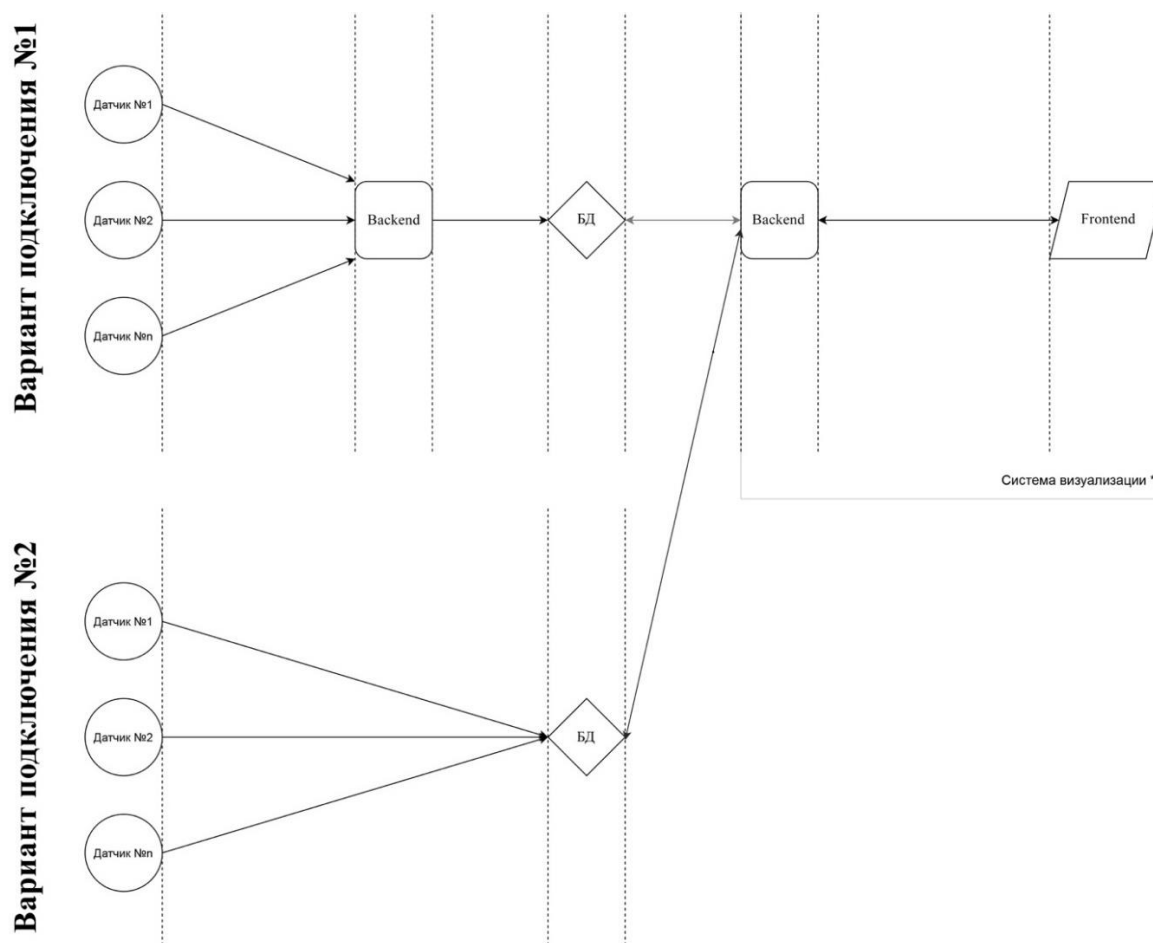


Рис. 1. Архитектура системы визуализации

RНР – это серверный язык создания сценариев. Конструкции RНР, вставленные в HTML-текст, выполняются сервером при каждом посещении страницы. Результат их обработки вместе с обычным HTML-текстом передается браузеру. В ходе исполнения RНР может изменить или динамически создать любой HTML-код, который и является результатом исполнения сценария. Затем сервер отправляет этот код браузеру. При этом браузеру не известно, как была сформирована данная страница — статично сверстана верстальщиком, или динамически создана при участии RНР.

Реализация пользовательского интерфейса. В веб-сервисе присутствует возможность настроить подключение к источнику или подключиться к уже имеющейся конфигурации (рис. 2), чтобы в дальнейшем была возможность просмотра всех активных датчиков на карте (рис. 3). Для реализации был выбран Vue.js так как он идеально подходит для создания малых и средних, по величине, проектов, так как нет необходимости установки различных библиотек и компонентов – фреймворк имеет множество встроенных функций. Главной из таких компонентов, при реализации реактивного приложения, является Vuex.

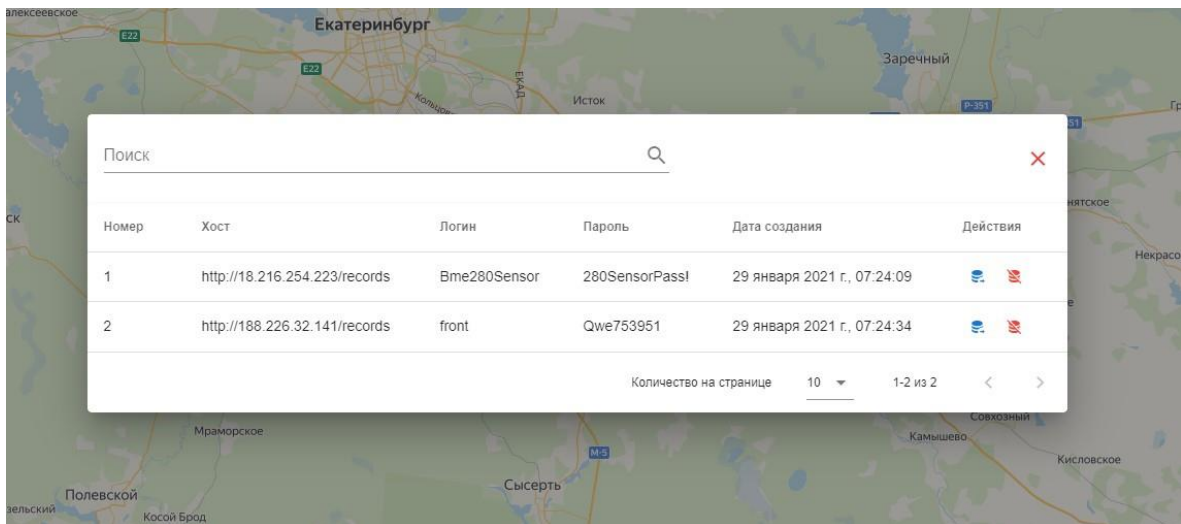


Рис. 2. Окно подключения баз данных к веб-приложению

Vueх – паттерн управления состоянием вместе с библиотекой для приложений на **Vue.js**. Он служит централизованным хранилищем данных для всех компонентов приложения с правилами, гарантирующими, что состояние может быть изменено только предсказуемым образом.

Преимущества хранилища **Vueх** от простого глобального объекта:

1. Хранилище **Vueх** реактивно. Когда компоненты **Vue** полагаются на его состояние, то они будут реактивно и эффективно обновляться, если состояние хранилища изменяется.
2. Нельзя напрямую изменять состояние хранилища. Единственный способ внести изменения – явно вызвать мутацию. Это гарантирует, что любое изменение состояния оставляет след и позволяет использовать инструментарий, чтобы лучше понимать ход работы приложения.

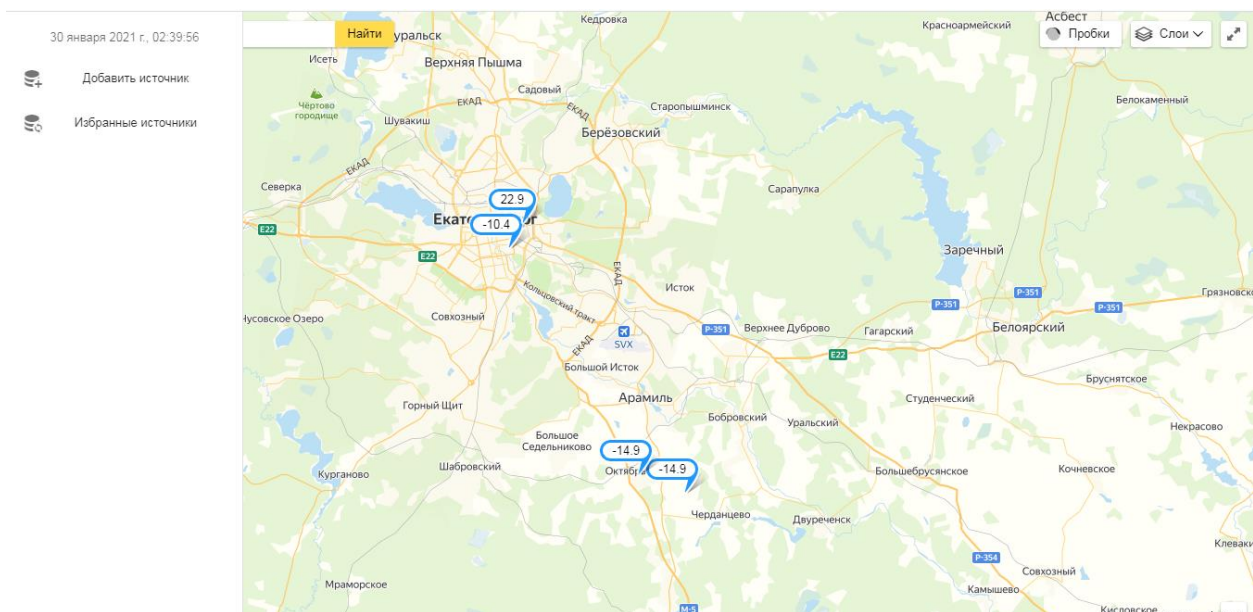


Рис. 3. Главная страница веб-сервиса

Для реализации отображения выбрана технология Yandex Map API. Yandex Map API – это инструмент, который поможет встроить на сайт или в приложение карту с поиском по топонимам и организациям, с возможностью строить маршруты и смотреть панорамы, а также с другими функциями, доступными на Яндекс.Картах. Картографические данные Яндекса постоянно обновляются, поэтому созданные с помощью API карты всегда актуальны — на них отображаются все изменения. Есть возможность настроить нужную логику взаимодействия пользователя с картой и определить, как эта карта будет выглядеть. Чтобы задать внешний вид объектов на карте, можно выбрать стандартные элементы или создать собственный макет.

Формат передачи данных с датчиков на клиентское приложение – JSON. JSON – это стандартный текстовый формат для представления структурированных данных на основе синтаксиса объекта JavaScript. Он обычно используется для передачи данных в веб-приложениях (например, отправка некоторых данных с сервера клиенту, таким образом чтобы это могло отображаться на веб-странице или наоборот).

Основные возможности веб-приложения:

- сортировка данных;
- графическое отображение данных из БД, при помощи сторонней библиотеки Chart.js. Данная библиотека позволяет создавать диаграммы JavaScript. Она имеет много опций, с помощью которых можно настроить все необходимые аспекты графиков. В данном случае использовались линейные графики с данными даты и температуры, для повышения наглядности полученных результатов (рис. 4);
- отображение в реальном времени. Все данные передаются и отображаются в приложении в реальном времени. Частота обновления напрямую зависит от конкретных настроек и возможностей каждого датчика.

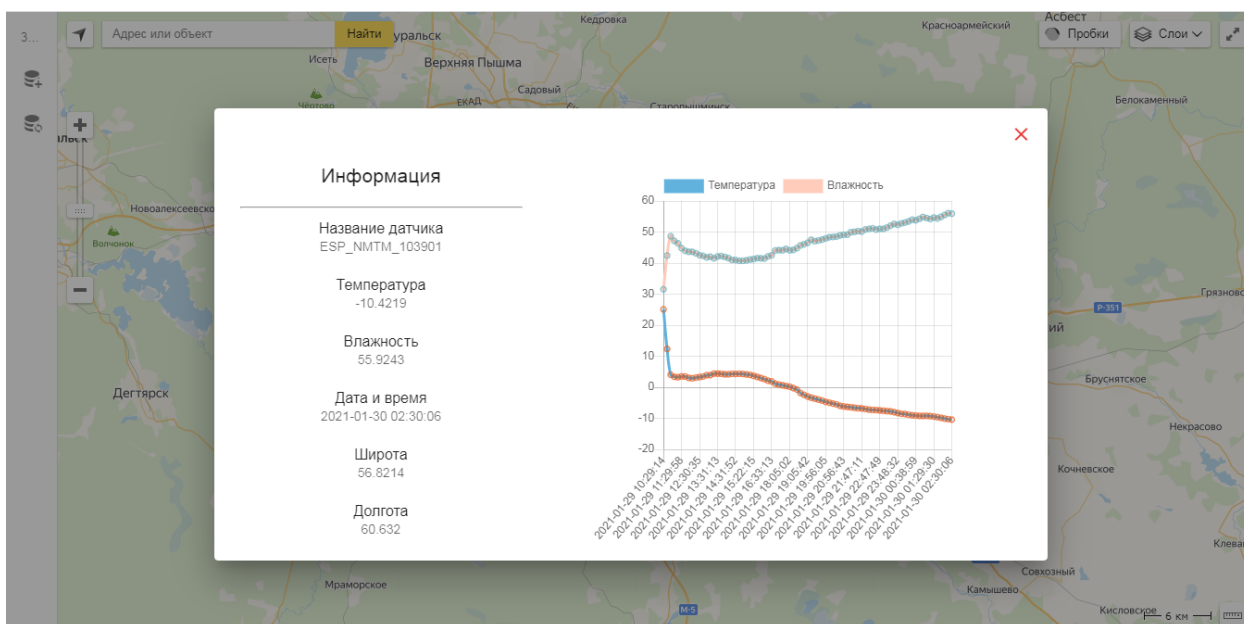


Рис. 4. Графическое отображение данных с датчиков

Выводы. Спроектировано и разработано веб-приложение для визуализации данных, полученных с датчиков, с использованием современных инструментов разработки. Данное web-приложение, при необходимости, легко масштабируется.

Список использованных источников

1. Стасышин, В.М. Базы данных: технологии доступа: учеб. Пособие для СПО / В.М. Стасышин, Т.Л. Стасышина, - 2-е изд., испр. И доп. – М.: Издательство Юрайт, 2018. – 164 с.
2. Гаевский, А.Ю. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. – М.: Наука, 2015. – 464 с.
3. Дронов, В. JavaScript в Web-дизайне / В. Дронов. – М.: СПб: БХВ, 2014. – 880 с.
4. Кингсли-Хью, К.Э. JavaScript 1.5: учебный курс / К.Э. Кингсли-Хью. – М.: СПб: Питер, 2013. – 272 с.
5. Иго Т. Arduino, датчики и сети для связи устройства. – СПб.: БХВ-Петербург, 2011. – 822 с.
6. Айвалиотс Д. Администрирование NGINX. – М.: ДМК Пресс-М, 2017. – 288 с.

УДК 378.147

А. В. Пономарев, А. А. Дерябина

ФГБОУ ВО «Череповецкий государственный университет», г. Череповец, Россия

ИНФОРМАЦИОННАЯ СИСТЕМА ОБУЧЕНИЯ БЕЗОПАСНОЙ РАБОТЕ НА ВЫСОТЕ

Аннотация. В настоящее время системы виртуальной реальности успешно используются на предприятиях для обучения персонала, занятого на опасных участках производства, а также в качестве виртуальных руководств по ремонту различного оборудования, техники. Одной из самых опасных работ является работа на высоте. Это вид человеческой деятельности, который связан с определенными рисками, требует от работников четкого соблюдения мер безопасности, знаний по охране труда. В данном докладе представлена разработка информационной системы обучения сотрудников безопасной работе на высоте с применением технологии виртуальной реальности, которая позволит сократить затраты предприятия и повысить эффективность обучения. Особенностью разработки является клиент-серверная архитектура и облачное хранение данных на выделенном сервере, а также многопользовательский доступ.

Ключевые слова: информационная система, виртуальная реальность, работа на высоте.

Abstract. Currently, virtual reality systems are successfully used in enterprises to train personnel employed in hazardous areas of production, as well as as virtual guides for the repair of