

В заголовке сайта пользователь может осуществить аутентификацию в системе для более комфортной работы. После входа пользователь, например, сможет сохранять результаты своих работ прямо на сайте и в последующем получать к ним доступ в любом месте, на котором он также осуществит вход. Также пользователь может сохранять информацию о роутере после входа, чтобы в будущем не вводить все данные снова.

В основном блоке была реализована опция загрузки файла местности. После загрузки изображения плана пользователь может установить роутер с заданными характеристиками в выбранную точку на плане. Далее происходит запрос на сервер, где высчитывается зона покрытия Wi-Fi и пользователю отображается результат. В левой части экрана реализован сайдбар, при активации которого отображаются предыдущие результаты работы программы. Сайдбар можно убирать и показывать, нажав на соответствующую кнопку.

Нижний колонтитул сайта содержит информацию об использованных технологиях, ссылки на GitHub аккаунты авторов, а также некоторые функциональные кнопки.

При разработке были задействованы следующие технологии:

- React – для создания разметки страницы, а также добавления интерактивности ее элементам.
- RESTful API – для получения информации о работе точек доступа Wi-Fi.
- GoLang – для функционирования Back-end приложения и обработки подаваемых к API запросов.
- CSS (в частности SCSS) – препроцессор для облегчения создания дизайна и оформления.
- Webpack – инструмент для сборки модулей проекта.

**Н. А. Миладовский**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **П. В. Бычков**, канд. физ.-мат. наук, доцент

## **РАЗРАБОТКА WEB-ПРИЛОЖЕНИЯ СБОРА И АНАЛИЗА СТАТИСТИЧЕСКИХ ДАННЫХ**

В настоящее время во всем мире возрастает интерес к сбору и анализу различной статистической информации.

С помощью удобного просмотра статистических данных разработчики могут узнать, как улучшить свое приложение. Такая статистика помогает понять, какие проблемы возникают при работе приложения и своевременно реагировать на события. Сбор статистики позволяет узнать поведение пользователей и усовершенствовать их взаимодействие с приложением.

С каждым годом все больше и больше набирает популярность модель оказания услуг SaaS (software as a service). При такой модели приложение создается в облаке, например в Amazon Cloud (AWS), Microsoft Azure, Google Cloud. Все вычисления и данные переносятся в облачную среду, а клиенты взаимодействуют с веб-интерфейсом.

Облачные приложения разрабатываются с учетом облачной архитектуры. Преимущества этой архитектуры заключается в большей гибкости, облачная среда предоставляет возможность развёртывать приложение по нескольким географическим областям и предоставляет защиту приложения и его инфраструктуры. Также облака предлагают множество полезных функций для управления развертыванием приложений, что упрощает разработку и поддержку приложения.

Одним из главных преимуществ такой архитектуры является возможность масштабирования. При использовании микросервисов, появляется возможность разработки веб-приложения, которое будет разбито на отдельные модули и будет располагаться на разных серверах. При разработке приложения по сбору статистических данных такая архитектура позволяет разрабатывать независимые модули и добавлять новые возможности. Также облачная архитектура предоставляет возможность уменьшать и увеличивать количество серверов, объем памяти и производительность системы, что позволяет гибко управлять модулями своего приложения.

**Б. А. Мирземетов**

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **В. Н. Леванцов**, ст. преподаватель

## **ФУНКЦИОНАЛ ТЕХНОЛОГИЙ ДЛЯ РЕАЛИЗАЦИИ ПОИСКА ВАРИАНТОВ АРЕНДЫ ЖИЛЬЯ**

За основной функционал в проекте отвечает React Framework. Этот фреймворк даёт нам возможность легко создавать клиентские