

ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»
ПО ПРОФИЛЮ «ИНЖЕНЕРНОЕ ДЕЛО»

16598

регистрационный номер

Компьютерные системы

название секции

Создание интерактивной среды московской школы

название работы

Автор:

Дубей Динеш

фамилия, имя, отчество

ГБОУ Школа №1155, 11

наименование учебного заведения, класс

Научный руководитель:

Юсупова Кристина Олеговна

фамилия, имя, отчество

ГБОУ Школа №1155

место работы

учитель информатики

звание, должность

подпись научного руководителя

Создание интерактивной среды московской школы

Аннотация

Целью данной работы является создание интерактивной карты школы, которая содержит удобный пользователю функционал для легкого ориентирования внутри здания, с возможностью поиска нужного внутреннего помещения, а также получение дополнительной информации по объекту. Для достижения поставленной цели были выполнены следующие задачи: создание интерактивной карты путем наложения интерактивных элементов поверх статичного изображения; внесение базы данных информации и структурирование ее в разрезе кабинетов; реализация перехода между этажами; создание плана эвакуации; создание панели администратора для изменения актуальной информации на сайте и в приложении; создание мобильного приложения с аналогичным функционалом, и реализация возможности создания событий, отображаемых на карте.

Данный проект создан на платформе Node.js (React.js) с использованием языков JavaScript, HTML, CSS. Реализована карта каждого этажа здания с возможностью переключения между этажами с помощью панели управления. На каждом этаже есть возможность по щелчку на элемент кабинета посмотреть информацию о нём: фамилию учителя, ответственного за кабинет, расписание кабинета на текущий день и его панораму. В программе создан поиск кабинета по его номеру или по фамилии учителя. Заложенный алгоритм поиска перебирает данные всех кабинетов, генерируя подсказки под поисковой строкой. Кроме того, в разработанной карте есть режим эвакуации при пожаре. При включении режима эвакуации на схеме отображаются вспомогательные линии, показывающие движение людей в здании от каждого кабинета согласно утвержденному плану эвакуации.

В работе представлено подробное описание взаимодействия всех частей проекта, представлена общая структура мобильного приложения и сайта, а также отличия в их реализации.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
ОСНОВНАЯ ЧАСТЬ.....	6
1 Архитектура проекта. Серверная часть (BACKEND).....	6
2 Клиентская часть сайта (FRONTEND)	9
2.1 Инициализация проекта (skaffolding). Интерфейс прикладного программирования (API)	9
2.2 Оформление сайта. Основные элементы интерфейса	11
3 Описание приложения.....	18
ЗАКЛЮЧЕНИЕ	20
Список использованных источников	21
Приложение А. Отзыв.....	22

ВВЕДЕНИЕ

В современном мире интерактивных технологий «умные» карты пользуются все большим интересом у пользователей. Существуют интерактивные карты местности с возможностью просматривать участки со спутника, а также видеть пробки и затруднение движения на дорогах в реальном времени. В торговых центрах используются карты для поиска необходимого магазина или кафе. Множество необходимых задач способны помочь решить пользователю данные средства. В рамках данного проекта была создана система инструментов для навигации в рамках выбранного образовательного учреждения: сайт и мобильное приложение. С учетом того, как быстро современные технологии внедряются в процесс школьного образования (электронный журнал, обучающие платформы и т.д.), то предложенный в проекте инструмент был бы уместным дополнением для быстрого ориентирования внутри здания школы, особенно здания нетипичного плана. Кроме того, карта оснащена дополнительными возможностями, которые также отсутствуют в других существующих школьных приложениях.

В данном проекте было решено создать интерактивную карту для корпуса №1 ГБОУ Школы №1566. Данный корпус имеет необычное архитектурное решение: здание отличается красивыми закругленными стенами с нестандартным остеклением и сложной планировкой этажей. Внутри здания помещения тоже располагаются оригинально: гостям школы или новым ученикам может быть сложно найти нужный кабинет. Именно поэтому создание удобной в использовании карты такого здания очень актуально. Актуальность работы заключается также в возможности использования созданного инструмента на интерактивной доске или на мобильных устройствах через сайт.

Удобство созданной карты обусловлено ее многофункциональностью. В карте совмещен поиск нужного кабинета с расписанием детей по классам. А также реализована возможность поиска кабинетов по их номерам или по фамилии учителей. Кроме того, в карту добавлена схема эвакуации, как необходимая функция для помещений с большим количеством людей.

Проект содержит две основные части с подробными подразделами: клиентская часть (FRONTEND), серверная часть (BACKEND).

Программные средства, используемые при реализации проекта:

- HTML – это язык разметки страницы («говорит» браузеру, какие элементы должны находиться на странице и в каком месте);
- CSS – это код для описания стилей веб-страниц, участвует на этапе оформления;
- JavaScript добавляет сайту функциональность и интерактивность;
- Node.js – программная платформа, основанная на движке V8, превращающая JavaScript из узкоспециализированного языка в язык общего назначения;
- React.js – JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов;
- Ant design – UI библиотека;
- WebStorm – интегрированная среда разработки на JavaScript, CSS & HTML от компании JetBrains;
- Express – веб-фреймворк для приложений Node.js;
- Expo – платформа для создания Android и IOS приложений с использованием JavaScript и React Native;
- MongoDB – нереляционная система управления базами данных.

Целью данной работы является создание веб-версии интерактивной карты школы и мобильного приложения, содержащего тот же функционал: схему здания для легкого ориентирования в нем с возможностью поиска нужного внутреннего помещения, а также получение дополнительной информации по каждому объекту карты.

Для достижения поставленной цели были выполнены следующие задачи:

1. Создание интерактивной карты путем наложения интерактивных элементов поверх статичного изображения;
2. Внесение базы данных информации и структурирование ее в разрезе кабинетов;
3. Реализация перехода между этажами с помощью специальной панели;
4. Создание плана эвакуации, а также внесение правил поведения при чрезвычайных происшествиях;
5. Создание мобильного приложения;
6. Создание панели администратора для изменения актуальной информации одновременно на сайте и в приложении;
7. Реализация возможности создания событий, отображаемых на карте.

ОСНОВНАЯ ЧАСТЬ

1 Архитектура проекта. Серверная часть (BACKEND)

Представим на рисунке 1 как взаимодействуют между собой основные элементы проекта. В схеме взаимодействия предоставлены независимые части системы, которые обмениваются данными при помощи интернета и обеспечивают их динамичность и безопасность. Администраторы при помощи админ-панели на сайте могут редактировать, создавать, удалять и читать расписания и события. Пользователи в свою очередь могут просматривать карту через сайт или через мобильное приложение, установка которого на устройство доступна для Android. Также с помощью приложения Expo Go карта школы доступна для Android и IOS.

Серверная часть проекта отвечает за обрабатывание и хранение данных (фотографий, расписания, информации о кабинетах), и при запросе клиента раздает данные для рендера (загрузки) карты. Кроме того, серверная часть обеспечивает защиту информации от посторонних лиц. Коллекция данных

хранится на облачном хранилище MongoDB. Для получения к ней доступа необходима ссылка, содержащая логин и пароль (рис. 2).

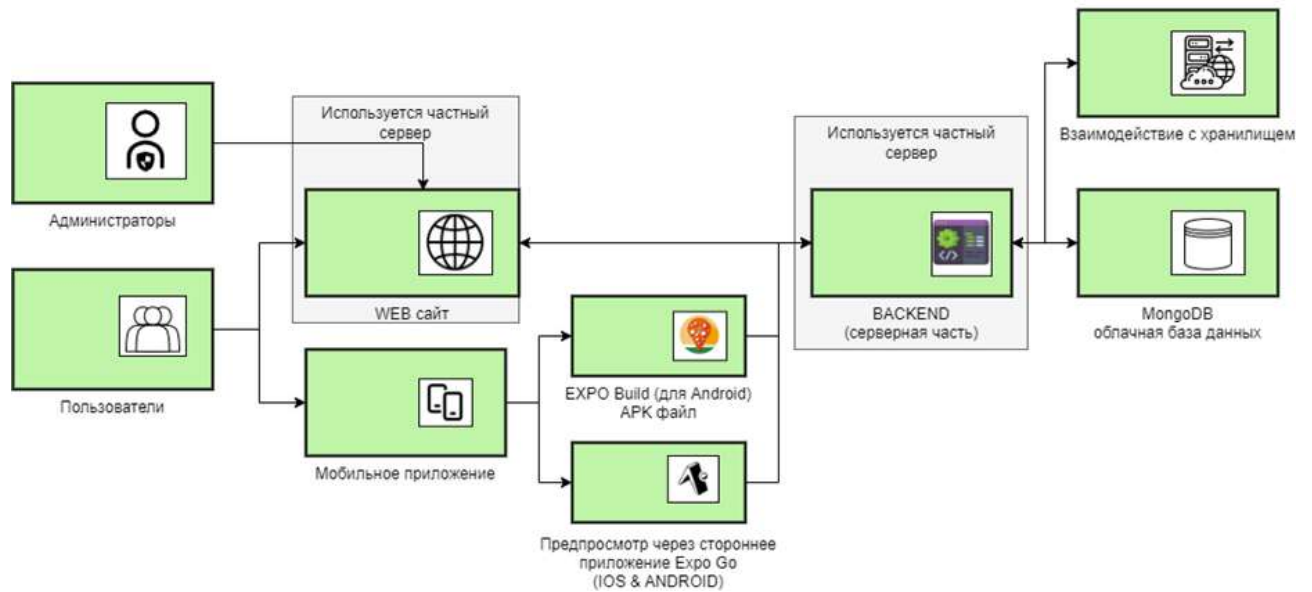


Рисунок 1 — Структурная схема проекта

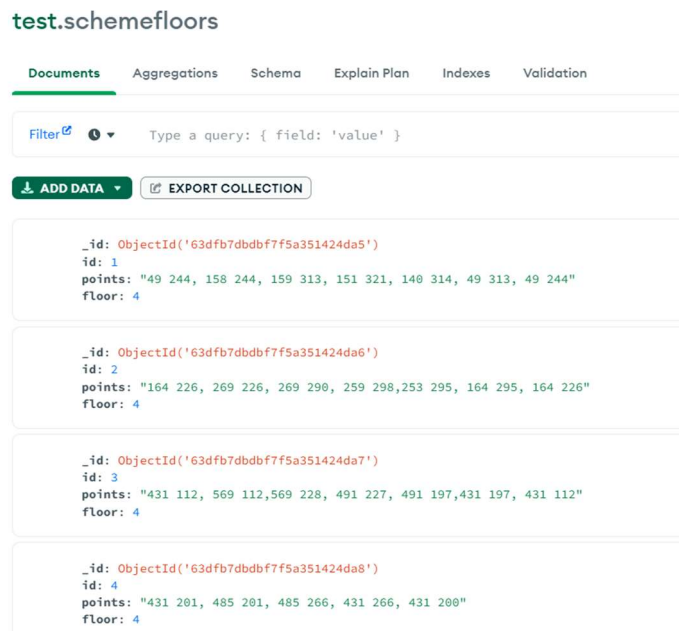


Рисунок 2 — Пример данных в программе MongoDB Compass

Сервер написан на той же платформе Node.js с использованием легкого фреймворка express. Рассмотрим подробно строение сервера. Приложение состоит из главного файла, роутера, контроллера и файла с объектами, описывающими базу данных. Первый из таких файлов Index.js – главный файл,

являющийся исполняемым файлом. К нему подключаются все нужные пакеты (библиотеки) и осуществляется запуск самого сервера. В файле `router.js` мы описываем все запросы, отправляемые пользователем при взаимодействии с сайтом (то есть при нажатии на какую-то кнопку для совершения одного из действий: создание, обновление, чтение или удаление), «эндпоинты» и передача функции для обработки запроса. В файле `Controller.js` создается класс, в который перемещают всю логику и ответы (`response`), которые вызываются в роутере. В `Schema.js` с помощью метода `mongoose.Schema` можно создать и описать поля и их настройки. Данный объект необходим для **CRUD**. **CRUD** – это аббревиатура, обозначающая основные функции работы с базой данных: **Create**, **Read**, **Update**, **Delete**.

Фреймворк **Express** отлично реализует инструменты для создания простого и легкого сервера. В работе удалось максимально просто без сложной архитектуры создать достаточно оптимальное приложение с точки зрения временных затрат.

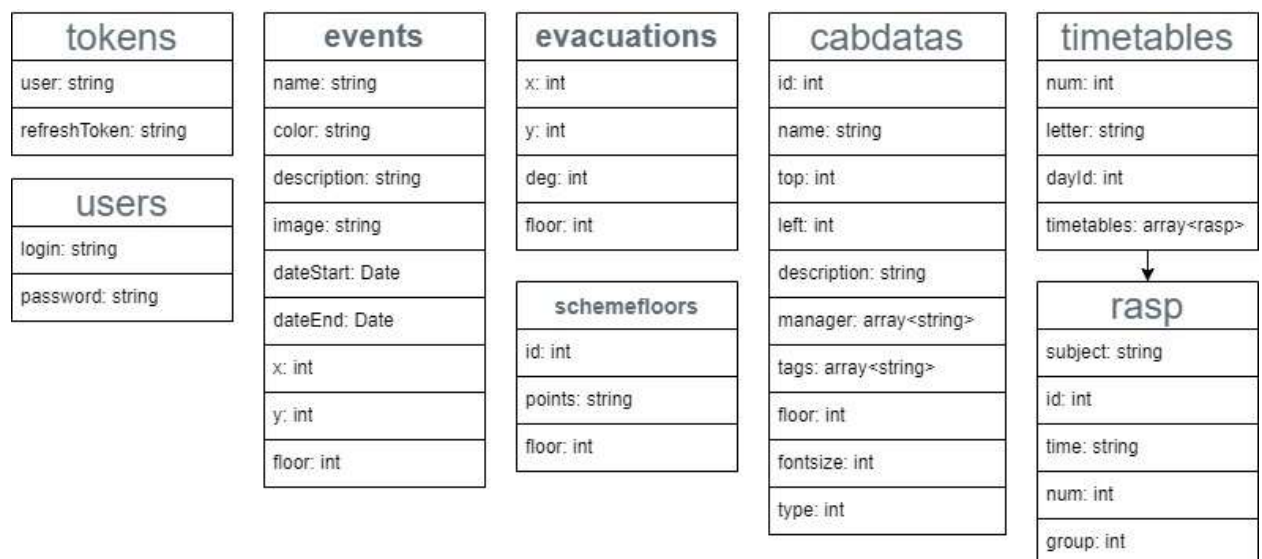


Рисунок 3 — Список моделей данных, хранящихся в базе данных

На рисунке 3 представлена схема, описывающая все модели данных, хранящихся в базе данных, такие как: JWT токены, учетные записи администраторов, события, стрелки эвакуации (на данный момент не

используется), схемы кабинетов (области, на которые можно нажать), информация о кабинетах и расписания.

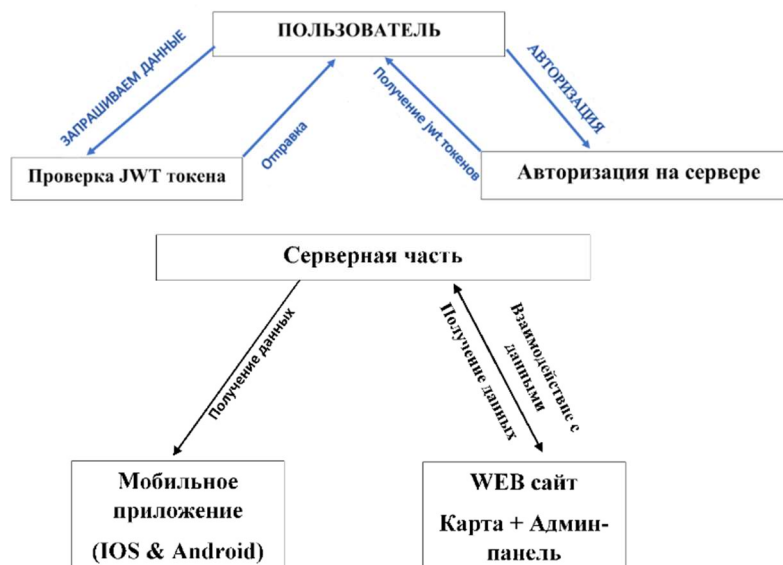


Рисунок 4 — Схема авторизации по JWT

Стоит отдельно пояснить этап авторизации пользователя, который упоминался ранее. На рисунке 4 представлена подробная схема данного процесса. Для авторизации клиент отправляет на определенный адрес запрос, содержащий логин и пароль от аккаунта администратора. При успешной валидации (логин и хэш пароля совпал), сервер возвращает два токена: access и refresh. Первый нужен для получения доступа к запрашиваемой информацией, а второй для генерации нового токена. Токены живут некоторое время. Если refresh токен устареет, то сайт запросит повторную авторизацию.

2 Клиентская часть сайта (FRONTEND)

2.1 Инициализация проекта (skaffolding). Интерфейс прикладного программирования (API)

Skaffolding — это процесс инициализации проекта, включающий развёртывание файловой структуры проекта и генерацию некоторого (иногда опционального) количества кода, требуемого пользователю [3].

Проект реализован на платформе Node.js с использованием библиотеки React. После установки Node.js на компьютер была создана рабочая директория и прописана команда инициализации проекта: `npm create-react-app my-app`. После окончания установки были удалены ненужные файлы.

API необходима для связи клиентской с серверной частью. В данном случае используется модель RESTAPI. Это http запрос (request) с получением ответа (response).

Методы HTTP-запроса:

- GET – получить подробную информацию о ресурсе;
- POST – создать новый ресурс;
- UPDATE – обновить существующий ресурс;
- DELETE – удалить ресурс.

Популярные коды response-ответов:

- 404 – Не найдено;
- 400 – Плохой запрос;
- 401 – Не авторизован;
- 403 – Запрещено;
- 500 – Ошибка сервера;
- 200 – ОК;
- 201 – Создано;
- 204 – Нет контента;
- 304 – Не изменен.

В работе было запланировано сделать в отдельно вынесенном файле функции, которые будут на заготовленные адреса отправлять запросы и возвращать данные.

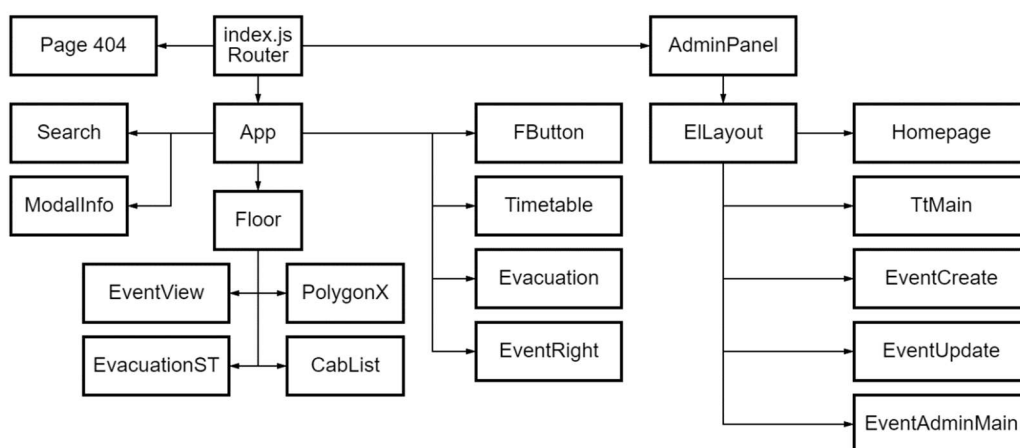


Рисунок 5 — Схема React компонентов

Стартовый файл – index.js. В зависимости от URL адреса определяется его дальнейший рендер. Если URL адреса не существует, то сайт «рендерит» (рисует) стилизованную страницу с ошибкой 404. Отрисовка идет от родительского компонента к дочернему. На схеме рисунка 5 представлено дерево компонентов на сайте проекта.

2.2 Оформление сайта. Основные элементы интерфейса

Основу разработанной карты составляют изображения схем этажей здания от -1 до 4 этажа. Чтобы связать данные изображения, был сделан переключатель, который меняет карту по нажатию на необходимую цифру на панели переключения.

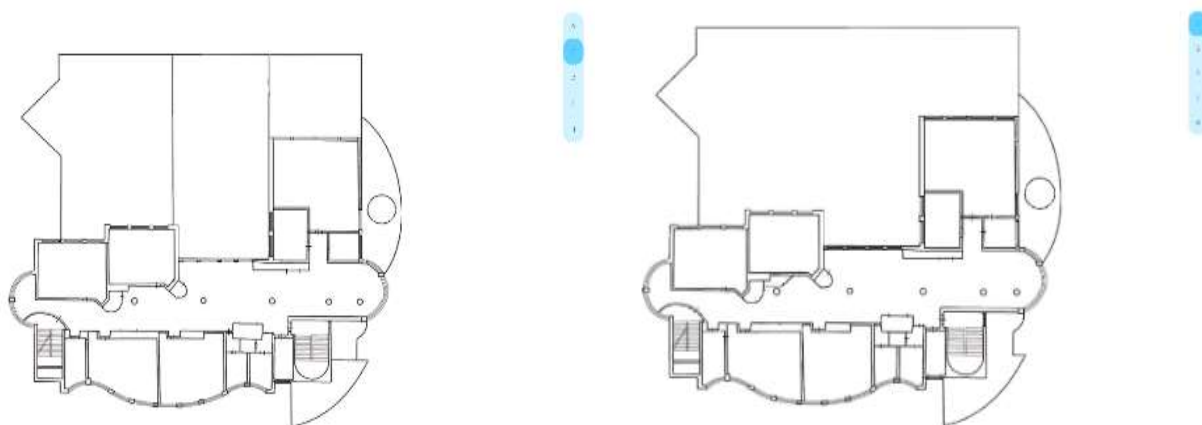


Рисунок 6 — Пример переключения между 2 и 4 этажами

При нажатии на виртуальные кнопки этажей, картинка меняется на то изображение, которому присвоено значение этой кнопки. На рисунке 6 показаны разные этажи, которые соответствуют своим значениям на панели переключения. Текущий этаж записывается в динамическую переменную (State). При изменении состояния обновляется весь компонент, в котором он использовался. Алгоритм представлен на рисунке 7.

```

32 // создаем состояние floor и функцию обновления состояния setFloor
33 const [floor, setFloor] = useState(Number(localStorage.getItem( key: 'started_floor')));
34 // ^ берем начальное состояние из локального хранилища на компьютере
35 const editFloor = (floor) => {
36   setFloor(Number(floor)); // функция, которая изменяет состояние
37 }

```

Рисунок 7 — Пример кода App.js

Стиль кнопок был написан вручную с помощью языка стилей CSS. Цвет и форма может видоизменяться в коде.

На карте также выделены учебные кабинеты и прочие помещения путем наложения интерактивных элементов поверх статичного изображения карты (рис. 9). Координаты этого участка прописываю в коде для дальнейшей привязки информации к этому объекту (рис. 8).

```

<svg> == $0
<polygon class=" " points="49 244, 158 244, 159 313, 151 321, 140 314, 49 313, 49 244"></polygon>
<polygon class=" " points="164 226, 269 226, 269 290, 259 298,253 295, 164 295, 164 226"></polygon>
<polygon class=" " points="431 112, 569 112,569 228, 491 227, 491 197,431 197, 431 112"></polygon>
<polygon class=" " points="431 201, 485 201, 485 266, 431 266, 431 200"></polygon>
<polygon class=" " points="522 231, 569 231,569 266, 522 266, 522 231"></polygon>
<polygon class=" " points="434 365, 462 365, 462 419,433 418 ,434 365"></polygon>
<polygon class=" " points="245 359, 256 358,257 366, 319 367,319 359, 349 359, 350 432, 345 437, 340 438, 335 439, 330 441, 325 443, 320 44
3, 315 444, 310 445, 305 446.5, 300 447, 295 447.5, 290 448, 285 448.5, 280 448, 275 448.5, 270 449.5, 265 449, 260 449, 255 449, 250 449, 2
47 448"></polygon>
<polygon class=" " points="130 357, 143 357, 143 367,242 366,242 448,235 448,230 447,225 447,220 446, 215 446, 210 445, 205 444, 200 443, 19
5 442, 190 440,185 439, 180 437, 175 434,170 431, 165 429, 160 427,155 425,150 423,145 422,140 420, 135 420,135 416, 130 416, 130 357">
</polygon>
<polygon class=" " points="92 367, 125 366,125 421,120 420,115 422, 110 423,105 425,100 428,97 429,92 429,92 367"></polygon>
<polygon class=" " points="355.14,358.57 364,358.57 364,365.57 375.43,365.57 375.43,381.14 375.43,384 386.86,384 386.86,419.43 382.86,420.86
380.57,420.86 374.86,422.29 371,423.71 365.43,425.14 360,427.43 355.14,429.43 "></polygon>
<polygon class=" " points="393 384,402 384,402 367,412 367,413 359,429 360,429 431, 421 431, 415 428, 410 426, 405 424, 400 423, 395 422.5,
393.3 422.5"></polygon>
</svg>

```

Рисунок 8 — Координаты <polygon>-элемента 4 этажа

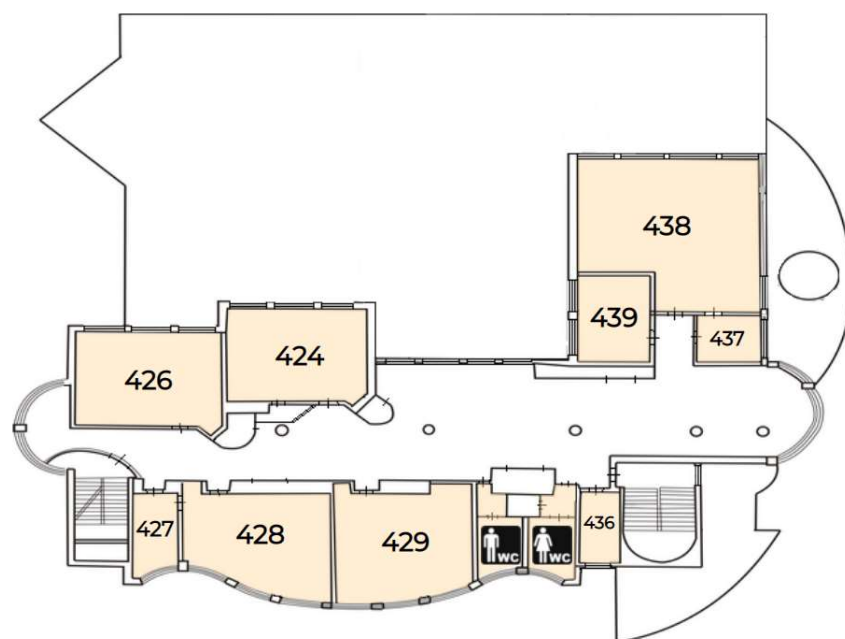


Рисунок 9 — Схема 4 этажа с элементами кабинетов

С помощью отслеживания событий, на сайте настроен механизм масштабирования. Это сделано для удобства просмотра деталей карты.

На рисунке 10 показано модальное окно для одного из кабинетов. Модальное окно – это окно, наложенное либо на документ, либо на другие окна. При этом, любой контент под модальным окном является недоступным для взаимодействия. Например, при щелчке по кабинету на карте открывается модальное окно, которое выводит следующую информацию:

- номер кабинета;
- описание (например, кабинет математики);
- ФИО: ответственного за кабинет учителя;
- расписание кабинета на сегодняшний день.

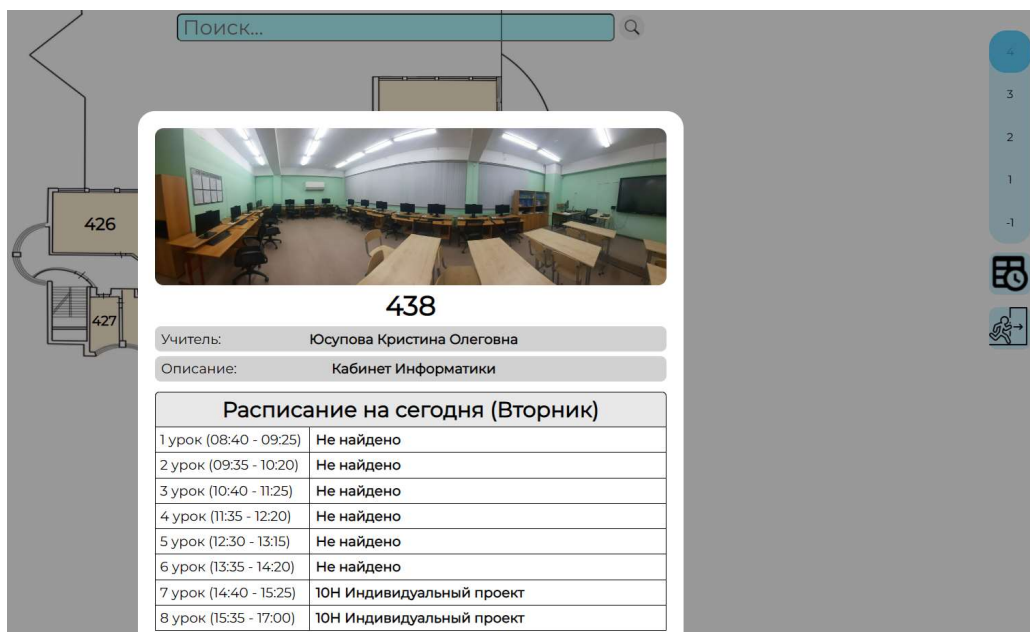


Рисунок 10 — Модальное окно для кабинета 438

Поисковая строка – это поле в системе, куда пользователь вводит свой запрос. После ввода запроса следует нажать значок «лупа» и выбирать кабинет по подсказке. После ввода запросы можно уточнить результаты, выбрав кнопку расширенного поиска. Для поиска необходимо ввести также ключевой запрос. Ключевым запросом является номер кабинета или ФИО педагога. На рисунке 11 продемонстрирована работа с возможностью поиска нужного кабинета.

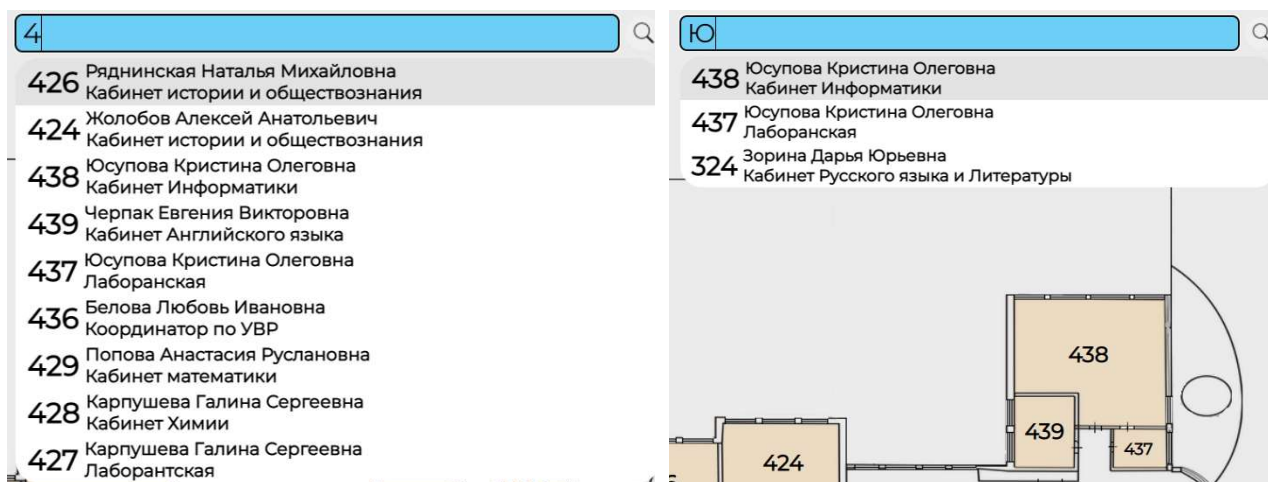


Рисунок 11 — Поиск кабинета

Кроме того, на сайте созданы дополнительные режимы такие как «Расписание» и «Эвакуация». При нажатии на соответствующую кнопку,

находящуюся на боковой панели, открывается меню с расписанием (рис. 12). Расписание можно менять на для каждого класса, используя select-меню. Расписание отображается по дням недели. Менять их можно, перелистывая таблицу с расписанием движением мыши с нажатой левой кнопкой, или при помощи кнопок расположенной на верхней панели. Чтобы открыть кабинет, нужно сделать двойной клик по строке с предметом в таблице расписания. Закрыть окно с расписанием можно нажатием на крестик, находящийся в правом верхнем углу.

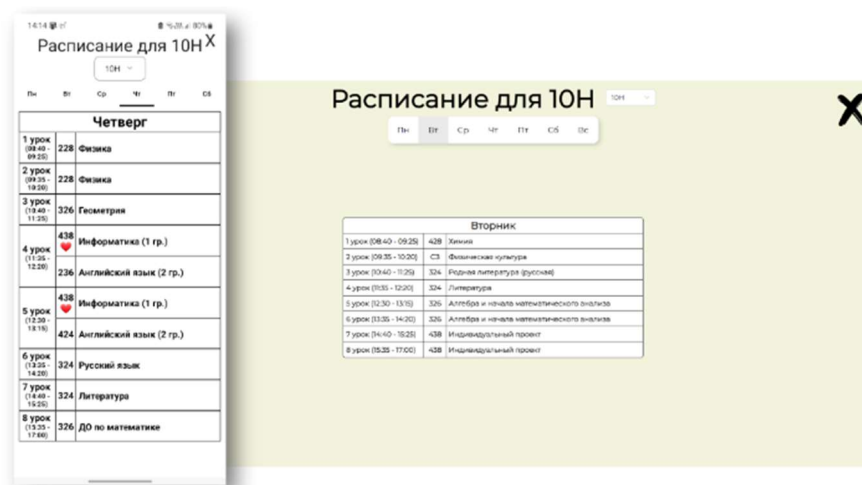


Рисунок 12 — Пример отображения расписания для одного из классов

Для ознакомления с планом эвакуации можно нажать на кнопку с иконкой «Пожарный выход». После чего на карте появятся стрелки, описывающие маршрут от каждого кабинета к пожарным выходам (рис. 13).



Рисунок 13 — Отображение плана эвакуации

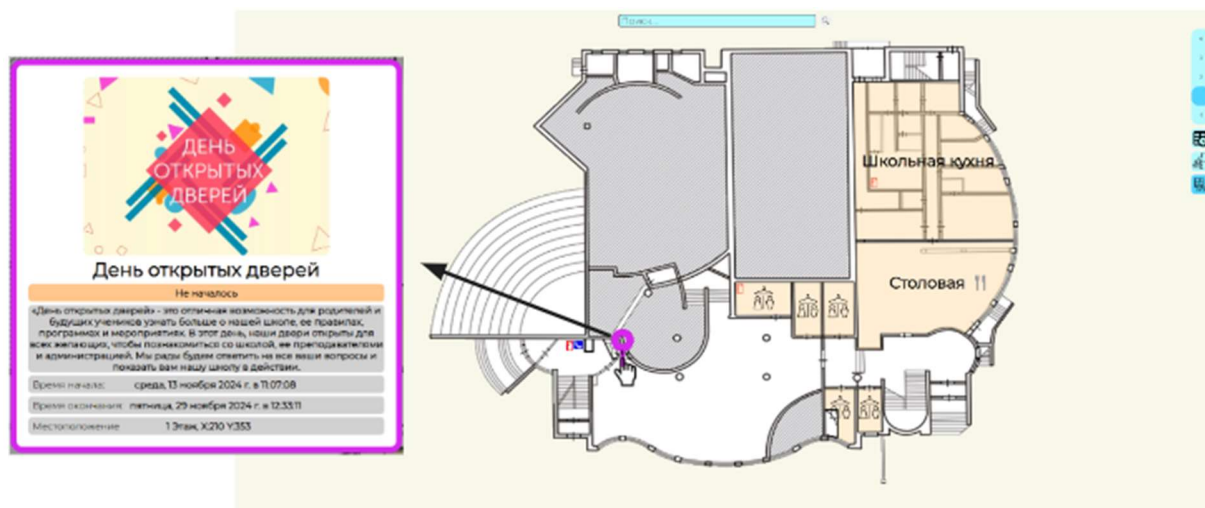


Рисунок 14 — Добавление события на карту

На карте также реализован механизм добавления локальных событий на карту (рис. 14). Для добавления события администратору надо кликнуть левой кнопкой мышки по месту, где будет располагаться значок и нажать «Создать событие». Авторизовавшись, осталось заполнить форму данных о событии и нажать кнопку.

Адаптация сайта под мобильные устройства также является важной частью работы, так как это самое популярное устройство, которым пользуются все люди (рис. 18). Школьники также являются активными пользователями, и именно мобильный вариант программы им более интересен.

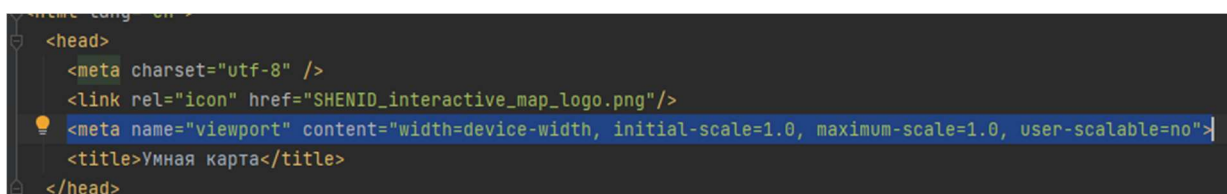
Адаптивная вёрстка сайта – это способ обеспечить оптимальное отображение интерфейсов на устройствах с разным разрешением экранов. В адаптивной вёрстке выделяют следующие принципы [4].

Контент десктопной и мобильной версии сайта дублируется. Всё, что пользователь может увидеть с компьютера, должно быть доступно и с телефона. Если какой-то информации не хватает в мобильной версии, придётся включать компьютер, искать сайт, а на нём – нужную информацию. Чтобы не вынуждать пользователя совершать столько действий, дизайнер должен перенести всю информации из десктопной версии в мобильную, ничего не выбрасывая [4].

Дизайн не меняется. Шрифты, цвета, логотипы должны быть одинаковыми в десктопной и мобильной версиях. Например, если в десктопной версии сайта использован голубой цвет, не стоит делать главным цветом мобильной версии зелёный. Если пользователь уже заходил на сайт с компьютера, то не узнает его в телефоне и подумает, что неправильно ввёл запрос или попал не на тот сайт [4].

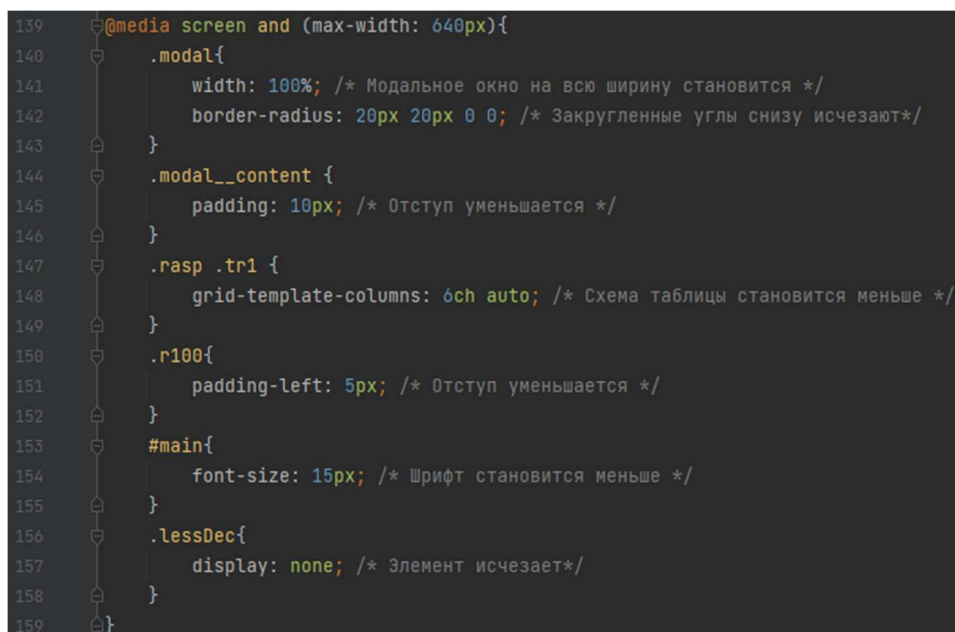
Сохраняется иерархия элементов. Иерархия заголовков, подзаголовков, текстовых блоков, иллюстраций и кнопок выстраивается на основе цели сайта.

Для адаптивной вёрстки сайта, были использованы meta-теги в html и css.



```
<head>
  <meta charset="utf-8" />
  <link rel="icon" href="SHENID_interactive_map_logo.png"/>
  <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
  <title>Умная карта</title>
</head>
```

Рисунок 15 — Пример meta -тега в html-файле



```
@media screen and (max-width: 640px){
  .modal{
    width: 100%; /* Модальное окно на всю ширину становится */
    border-radius: 20px 20px 0 0; /* Закругленные углы снизу исчезают*/
  }
  .modal__content {
    padding: 10px; /* Отступ уменьшается */
  }
  .rasp .tr1 {
    grid-template-columns: 6ch auto; /* Схема таблицы становится меньше */
  }
  .r100{
    padding-left: 5px; /* Отступ уменьшается */
  }
  #main{
    font-size: 15px; /* Шрифт становится меньше */
  }
  .lessDec{
    display: none; /* Элемент исчезает*/
  }
}
```

Рисунок 16 — Пример meta-тега в css-файле

На рисунках 15–16 показаны примеры meta-тегов в файлах html и css.

Ссылка на сайт: <https://interactive-school-map.ru/>

3 Описание приложения

Так как язык программирования во всех частях проекта был одинаковый (JavaScript), то по архитектуре мобильная часть мало чем отличается от браузерной. Основное отличие идёт в способах исполнения визуализации.

Для разработки мобильного приложения была использована платформа Expo основанная на React и на компонентах из React Native. Логика обработки данных почти не изменялась. Основное различие легло в отображении. Для сайта использовалась разметка HTML и CSS, а для приложения – нативные компоненты. На схеме рисунка 17 представлены различия базовых элементов разметки и стилей.

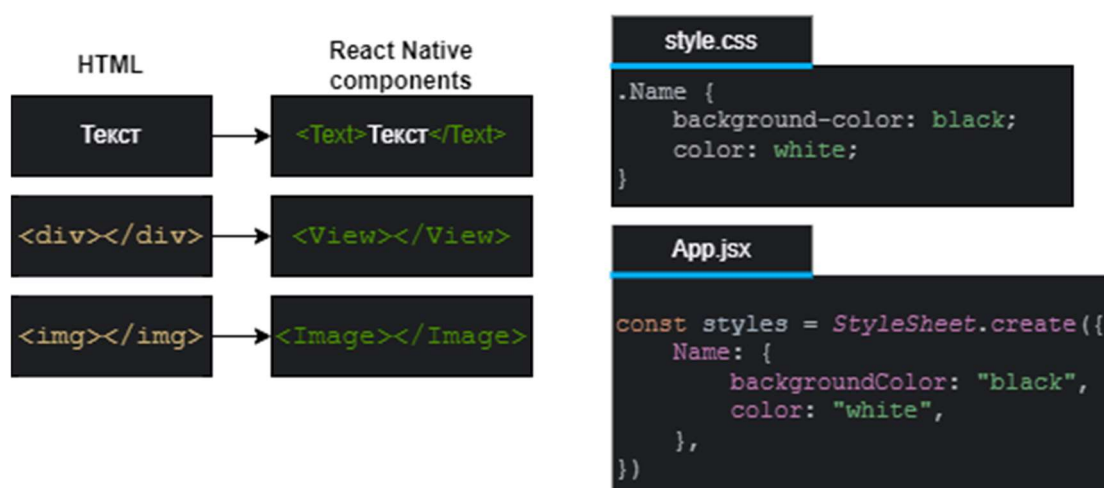


Рисунок 17 — Базовые отличия кода между приложением и сайтом



Рисунок 18 — Приложение с картой школы на различных устройствах

Для работы масштабирования карты был полностью переписан алгоритм вычисления координат. Анимации и остальные компоненты, например, компонент `<Svg>`, которого изначально не было в библиотеке React Native. Но тем не менее, приложение работает стабильнее и быстрее браузерной версии. Его данные, например, события также меняются через админ-панель и синхронизированы с сайтом (рис. 18).

Ссылка на приложение: <https://apps.rustore.ru/app/com.shenid.mobileApp>

ЗАКЛЮЧЕНИЕ

Целью данной работы являлось создание подробного интерактивного поэтажного плана школы, который содержит удобный пользователю функционал для легкого ориентирования внутри здания. Поставленная цель достигнута – реализована карта каждого этажа здания с возможностью переключения между этажами с помощью панели управления. На каждом этаже есть возможность по щелчку на элемент кабинета посмотреть информацию о нём: фамилию учителя, ответственного за кабинет, расписание кабинета на текущий день и его панораму.

В программе создан поиск кабинета по его номеру или по фамилии учителя. Присутствует удобное ученическое расписание с быстрой и понятной навигацией по классам и возможностью поиска кабинета по щелчку на нужный урок. На карте также присутствует режим отображения плана эвакуации: визуализация маршрута от каждого кабинета к пожарным выходам с помощью стрелочек в соответствии с утвержденным планом эвакуации школы. Данный режим также содержит краткую инструкцию о действиях обучающихся при пожаре. Через роль администратора можно создать на карте метки с событиями по щелчку на необходимую область его расположения на плане школы. Данный функционал необходим для уведомления пользователя о нерегулярных локальных мероприятиях, таких как живая музыка на переменах в холе одного из этажей.

Разработанная карта реализована в двух вариантах: сайт и мобильное приложение. Приложение удобно использовать как посетителям школы, например, на дне открытых дверей, так и для обучающихся школы, которые в определенное время ищут нужный кабинет по встроенному расписанию. Сайт может служить основой для интерактивного стенда при входе в школу для поиска нужного кабинета гостями. Удобство обусловлено, в первую очередь, подробным и структурированным представлением всей информации об устройстве здания, а также удобным интерфейсом с возможностью масштабирования и адаптацией под мобильные устройства.

Список использованных источников

1. И.В. Горячая, Пользовательский интерфейс: назначение, принципы разработки и практические рекомендации [Электронный ресурс] URL: <https://al.cs.msu.ru/files/goryachaya.user.interface.2010.pdf> (дата обращения: 20 декабря 2022)
2. Архитектура проектов [Электронный ресурс] URL: <https://dev.1c-bitrix.ru/learning/course/...> (дата обращения: 22 декабря 2022)
3. Frontender Magazine [Электронный ресурс] URL: <https://frontender.info/skaffolding-dlya-frontenderov/> (дата обращения: 13 января 2023)
4. Как адаптировать сайт для мобильных: методы и примеры [Электронный ресурс] URL: <https://practicum.yandex.ru/blog/kak-adaptirovat-sayt-pod-mobilnye-ustroystva/#id1/> (дата обращения: 3 марта 2023)

Приложение А. Отзыв



Федеральное государственное
бюджетное образовательное учреждение
высшего образования
«Национальный исследовательский
университет «МЭИ» (ФГБОУ «МЭИ»)
111250, г. Москва,
вн.тер.г. муниципальный округ Лефортово,
ул. Красноказарменная, д. 14, стр. 1
Тел.: (495) 362-75-60, факс: (495) 362-89-38
E-mail: universe@mpei.ac.ru
<http://www.mpei.ac.ru>

№ _____

« ____ » _____ 20 ____ г.

ОТЗЫВ

на проектную работу
(проектную / исследовательскую)

по теме «Создание интерактивной среды московской школы»
(наименование работы)

учащегося 11 класса ГБОУ Школа №1155
(класс) (наименование образовательной организации)

Дубей Динеш
(фамилия, имя, отчество (при наличии), заполняется на каждого участника)

В рамках проекта «Создание интерактивной среды московской школы» автор создал уникальную систему, которая включает в себя сайт и мобильное приложение с заложенной в них картой школы для удобной навигации внутри здания. Кроме того, функционал приложения и сайта дополнены планом эвакуации, быстрым поиском нужного кабинета, а также расписанием школы и созданием меток на карте в зоне локальных событий.

В работе подробно изложены и качественно проиллюстрированы этапы разработки сайта и приложения, присутствует описание клиентской части (внешнее оформление, адаптация под различные устройства, основные элементы интерфейса), также большое внимание уделяется в работе серверной части (роль администратора, работа в панели администратора, изменение и обновление содержимого на сайте и в приложении). Автор владеет программными средствами профессиональной разработки сайтов и приложений на высоком уровне, что отражено в описании задействованного в проекте программного обеспечения.

Цели работы автором достигнуты, актуальность представленного в работе продукта и его практическая значимость подтверждены. Автор создал удобный инструмент для поиска внутренних помещений внутри школы и предлагает свою систему как единую среду для навигации во всех московских школах.

ассистент
(должность)


(подпись)



Шейкин М.О.
(И. О. Фамилия)