

**МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ**

**ОТЧЕТ
по научно-исследовательской работе
Тема: Приложение для расписания занятий университетов**

Студент гр. 5303

Десятов А.С.

Руководитель

Заславский М.М.

Санкт-Петербург
2020

**ЗАДАНИЕ
НА НАУЧНО-ИССЛЕДОВАТЕЛЬСКУЮ РАБОТУ**

Студент Десятов А.С.

Группа 5303

Тема НИР: Разработка онлайн расписания для учебных заведений

Задание на НИР:

Разработать онлайн приложение для составления расписания учебными заведениями для его последующего просмотра студентами

Сроки выполнения НИР: 09.12.2020 – 25.12.2020

Дата сдачи отчета: 25.12.2020

Дата защиты отчета: 25.12.2020

Студент

Десятов А.С.

Руководитель

Заславский М.М.

АННОТАЦИЯ

Цель научно исследовательской работы – создание онлайн приложения для составления и просмотра расписания студентами университета. Для этого необходимо изучить современные технологии работы с web, лучшие практики и особенности работы университетов.

SUMMARY

The purpose of the research work is to create an application for scheduling and viewing the schedule by university students. To do this, it is necessary to study modern web technologies, best practices and features of universities.

ВВЕДЕНИЕ

Целью исследовательской работы была разработка внешней части приложения. Основная работа была сделана в области front-end. Были изучены такие технологии как React, React-redux, sass, Nodejs. Был создан сервер для обработки запросов и клиент для показа данных для пользователя. Была создана небольшая база данных с занятиями для университета и создана логика для показа этих данных.

1. ПОСТАНОВКА ЗАДАЧИ

1.1. Актуальность

Задачей научно исследовательской работы является создание онлайн приложения для составления и просмотра расписания студентами университета. В текущее время, возможность просмотра расписания занятий для студентов используя интернет является довольно привлекательной. К сожалению, университеты имеют разную структуру своих сайтов и поиск такой информации как время занятия, преподаватели, местоположение (аудитория), дополнительные материалы представляет сложность. Имея приложение, где студент сможет просмотреть необходимую ему информацию, представленную в определенном виде, используя при этом один ресурс и такие устройства как телефон или компьютер, уменьшит время, необходимое для поиска этой информации в другом месте и положительно скажется на технологической составляющей университета.

1.1. Проблема

Информация о занятиях университета имеет разный формат и структуру.

1.2. Цель

Создание онлайн приложения для составления и просмотра расписания студентами университета.

1.3. Задачи

Задачи научно исследовательской работы:

- Создать прототип серверной части приложения с REST архитектурой
- Создать прототип клиентской части приложения с использованием технологии React и React-redux
- Стилизовать приложение для лучшего восприятия информации

1.4. Описание метода решения

При просмотре существующих вариантов реализации расписания в различных вузах города Санкт-Петербург не было найдено альтернатив, поэтому разрабатываемое решение можно назвать уникальным.

Решение было разработано с помощью веб-технологий. В связи с тем, что ресурсы мобильной сети ограничены и не каждый студент может позволить себе тратить их на загрузку приложения для определенной платформы, было принято решение разработать website.

Преимущества такого выбора это:

- Кросс-платформенность
- Единая кодовая база для всех платформ
- Доступность приложения для различных контекстов

Разработанное приложение может быть использовано в следующих сценариях:

- Использование студентом предполагает просмотр информации о интересующей его паре
- Использование преподавателем предполагает изменение информации о предмете, к которому у него есть доступ

- Использование администрацией университет предполагает изменение информации о всех предметах

В качестве используемых технологий был выбран framework React для разработки интерфейса пользователя и Node.js для разработки серверной части. Для стилизации приложения был выбран препроцессор Sass.

Структуру интерфейса приложения составляет множество компонентов из которых этот интерфейс в последствии и состоит. Эта структура предполагает передачу данных от компонентов родителя в компоненты детей (child-component). На рисунке 1 представлена структура одного из компонентов.

```
class Navigation extends React.Component {
  constructor(props) {
    super(props);
    this.state = {
      loginModalVisible: false
    };
  }

  renderLinkToProfile() {
    return (
      <span className="navigation__link">
        <Link to="/profile">Profile</Link>
      </span>
    );
  }

  render() {
    return (
      <React.Fragment>
        <nav className="navigation">
          {this.props.user && this.props.user.login ? (
            this.renderLinkToProfile()
          ) : (
            <span className="navigation__link" onClick={this.props.toggleLoginModal}>
              Login
            </span>
          )}
          <span className="navigation__link">
            <Link to="/gallery">Gallery</Link>
          </span>
          {this.props.user && this.props.user.login ? (
            <span className="navigation__link" onClick={this.props.logout}>
              Logout
            </span>
          ) : null}
        </nav>
      </React.Fragment>
    );
  }
}
```

Рис. 1. Описание класса Navigation

Интерфейс пользователя выглядит следующим образом:

После перехода по адресу, на котором запущенно приложения, пользователю дается выбор его университета (см. рис. 2).



Рис. 2. Окно выбора университета пользователем

После выбора университета, пользователь будет уведомлен о том, что необходимо выбрать свой курс, факультет и группу. При нажатии кнопки меню в верхнем правом углу, пользователю откроется конфигурационное меню, где он сможет выбрать курс, факультет и группу (рисунок 3).

X

University Schedule

University year:

▼

Select faculty:

▼

Select group:

▼

Reset

Рис. 3. Конфигурационное меню

После выбора группы, на странице появится расписание по выбранным данным и кнопка для переключения между 1-ой неделей и 2-ой (рисунок 4).

		week 1		week 2		
Time	8:00-9:20	9:50-11:20	11:40-13:15	13:45-15:15	15:35-16:55	17:25-18:55
Monday		Physical culture Details	Programming 5404 Lecture Teachers: Krinkin Details			
Tuesday		Mathematical analysis 3308 Lecture Teachers: Kolpakov A.S. Details	Algebraic structures 3308 Lecture Teachers: Zelvenski Details	Algebraic structures 3308 Lecture Teachers: Zelvenski Details	Mathematical analysis 3427 Practice Teachers: Egorov Details	
Wednesday	Physical culture Details	Physics 3107 Lecture Teachers: Levshakov Details	Algebraic structures 4210 Practice Teachers: Medvedev Details			
Thursday						

Рис. 4. Секция просмотра

- Каждая ячейка таблицы - это пара. В ней указана такая информация, как:
- Название предмета
 - Аудитория
 - Тип занятия (лекция, практика, лабораторные занятия)
 - Преподаватели

Также в каждой ячейке есть кнопка, по нажатию которой открывается модальное окно с более подробной информацией о выбранном предмете (см. рис. 5).

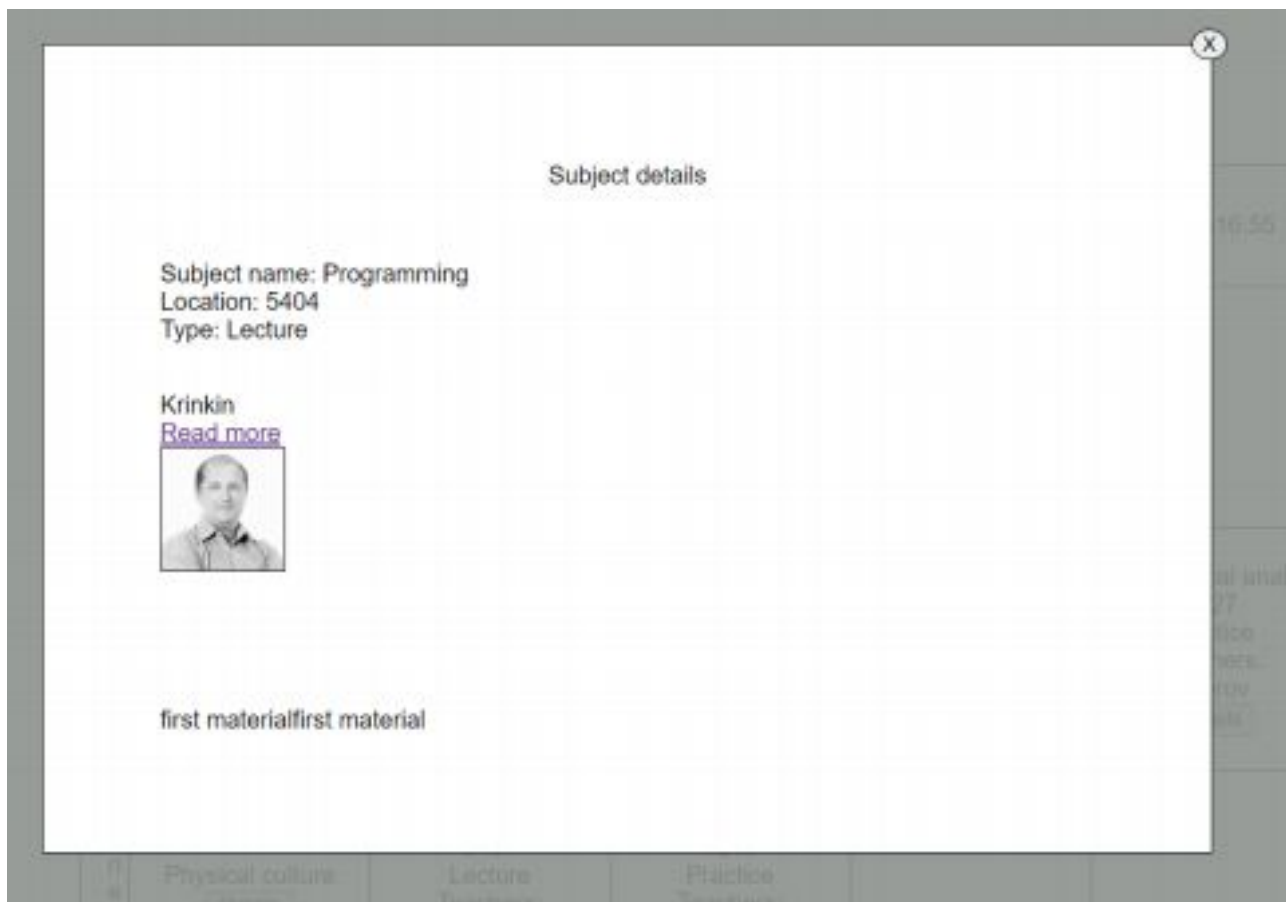


Рис. 5. Окно подробной информации

Это окно включает в себя фотографию преподавателей, которые ведут предмет с ссылкой на более подробную информацию на сайте университета. В будущем, это окно будет дополнено материалами и домашними заданиями, которые укажет преподаватель.

Примеры запроса REST указаны в таблице 1

Таблица 1. Разработанные endpoint'ы

Название endpoint'a	Адрес для обращения	Назначение
fetchUniversities	/universities	Запрос на получения списка доступных университетов
loginUser	/loginUser	Запрос для авторизации пользователя
fetchScheduleData	/fetchScheduleData	Запрос на получение списка курсов, факультов, групп и занятий для определенного университета

На данной стадии разработки модель данных не используется. В будущем будет использовать mysql. Приложение также не покрыто тестами.