

Министерство науки и высшего образования Российской Федерации  
**Муромский институт**  
федерального государственного бюджетного образовательного учреждения  
высшего образования  
**«Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»  
(МИВлГУ)**

Факультет \_\_\_\_\_ ИТР

Кафедра \_\_\_\_\_ ИС

# ***ПРАКТИЧЕСКАЯ РАБОТА №4-7***

по \_\_\_\_\_ Распределенные ИС

Тема Организация живого поиска на сайте с помощью фреймворка  
Vue.js

Руководитель

Варламов А.Д.  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись) (дата)

Студент ИСм-121  
(группа)

Минеев Р. Р.  
(фамилия, инициалы)

\_\_\_\_\_  
(подпись) (дата)

Муром 2022

## Практическая работа №4-7.

*Тема:* Организация живого поиска на сайте с помощью фреймворка Vue.js.

*Цель работы:* приобрести навыки организации живого поиска на сайте с помощью технологии Vue.

Для прехода от технологии Ajax на фреймворк Vue.js был изменён код html:

```
<div id="app" class="page">
  <form>
    <div class="searchWindow">
      <div class="searchField">
        <input id="input" v-model="input" placeholder="Search your word..."
class="textHolder" type="text" autocomplete="off" v-on:input="findSimilar">
      </div>
      <div v-if="items.length > 0" id="contentBox" class="contentSearched">
        <div class="line"></div>
        <ul id="list" class="items">
          <li @click="setWord" v-for="item in items"
class="item">{{item.title}}</li>
        </ul>
      </div>
    </div>
  </form>
</div>
```

Объявляем пространство проекта id="app" и к input добавляем обработчик события ввода текста v-on. Добавляем условие к отображению элементов поиска div v-if, которое будет добавлять div на страницу, когда будет, что показывать из вариантов поиска. Список поиска инициализируется циклом li v-for.

Создадим обработчики событий для поиска и установки вариантов поиска:

```
var app = new Vue({
  el: '#app',
  data: {
    input: "",
    items: [],
    placeholder: "Search your word..."
  },
  methods: {
    setItems: function(data) {
      this.items = []
      console.log(data);
      console.log(data['data'][0]['word'])
      data['data'].forEach(obj => {
        this.items.push({title: obj['word']})
      });
    }
  }
});
```

					МИВУ 09.04.02-04-7.001			
Изм	Лист	№ докум.	Подп.	Дата	Практическая работа №4-7 Организация живого поиска на сайте с помощью фреймворка Vue.js	Литера	Лист	Листов
Студент	Минеев Р. Р.			03.03.		У	2	3
Руков.	Варламов А. Д.					МИ ВлГУ ИСм-121		
Конс								
Н.контр.								
Утв.								

```

findSimilar: function(event){
    if (event.target.value.length > 2) {
        const requestOptions = {
            method: "POST",
            mode: 'cors',
            headers: { "Content-Type": "application/json" },
            body: JSON.stringify({ "word": event.target.value })
        };
        fetch("http://localhost/php/hello.php", requestOptions)
            .then(response => response.json())
            .then(data => this.setItems(data));

    } else {
        this.items = []
    }
},

```

findSimilar отправляет слово из строки поиска в виде json файла на localhost на выполнение файлу hello.php, а setItems добавляет на страницу всё, что вернул метод findSimilar.

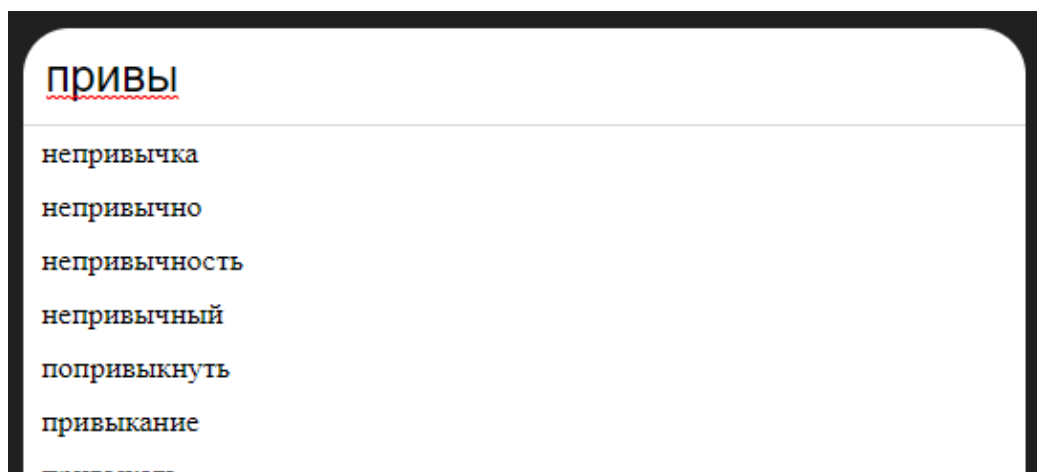
```

<?php
$data = json_decode(file_get_contents('php://input'), true);
header('Content-Type: application/json; charset=utf-8');
$word = $data['word'];
$mysqli = new mysqli("localhost", "root", null, "words");
$mysqli -> query("SET NAMES 'utf8'") or die (json_encode('Database connection error'));
$db_words = $mysqli -> query("SELECT * from WORDS WHERE word LIKE '%$word%' LIMIT 13");
$data = [];
while ($row = $db_words -> fetch_assoc()) {
    $data[] = $row;
}
$response = [];
$response["data"] = $data;

echo json_encode($response, JSON_PRETTY_PRINT);
?>

```

Данный код найдёт 13 (или меньше) похожих слов и вернёт их обратно в скрипт.



Вывод: В данной практической работе были приобретены навыки работы с Vue.js и реализован механизм живого поиска слов в базе данных.