**МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»**

**(МИИТ)**

|  |
| --- |
| **ИНСТИТУТ ТРАНСПОРТНОЙ ТЕХНИКИ И СИСТЕМ УПРАВЛЕНИЯ**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Кафедра «Управление и защита информации»

Лабораторная работа №4

**«Веб-приложения (LSPWA) под управлением фреймворка Vue.js на языке JavaScript»**

по дисциплине

**«Web программирование»**

|  |  |
| --- | --- |
|  | Выполнил: студент группы ТКИ-541  Куминов В. П.  Проверил: к.т.н., доц.  Сафронов А. И. |

**Москва – 2023 г.**

**1. Цель:**

Построить два одностраничных веб-приложения (LSPWA) под управлением фреймворка Vue.js на языке JavaScript.

**2. Постановка задачи:**

Разработать два простых локальных одностраничных веб-приложение (LSPWA) под управлением фреймворка Vue.js на языке JavaScript в соответствии с указаниями вариантов индивидуального задания (покрываются два раздела последовательного вычислительного процесса).

Базовая структура отчёта по работе:

1. Цель работы.

2. Формулировка задачи.

3. Технология локального подключения фреймворка Vue.js.

4. Таблица соответствия переменных и методов, используемых в веб-приложениях.

5. Содержательная часть по параметрической вставке (код веб-приложения + отображение в браузере (с учётом изменяющихся состояний) + Сеть Петри).

6. Содержательная часть по методам и событиям (код веб-приложения + отображение в браузере (с учётом изменяющихся состояний) + Сеть Петри).

7. Вывод.

Не использовать: jQuerry, «чистый» JavaScript, каскадные таблицы стилей (CSS), директиву v-for, директивы v-if, v-else, v-else-if, v-model, теги <template>.

Реализовать:

– локально (LSPWA), не прибегая к инструментарию Node.js и npm (Node Package Manager);

– в файле с именем index\_Familiya\_I\_O.html

Настроить при помощи параметрической подстановки тег <font></font>. Семейство, размер и цвет шрифта задать параметрически, установив «Courier New», 3 единицы и тёмно-синий цвет (из числа перечислимых цветовых констант), соответственно.

Предусмотреть элемент <input></input>, который без маски проявляет стандартное поведение. При маскировании клавишей «Alt» выдаёт значение транслитерации введённого кириллического символа. На остальные символы при маскировании <input></input> не реагирует.

**3. Ход работы**

Построим сеть Петри первого приложения:



Рисунок . Сеть Петри приложения 1

Построим сеть Петри второго приложения:

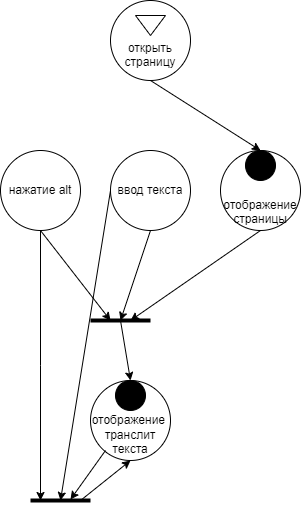


Рисунок Сеть Петри второго приложения

**Код программы приложения 1:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Vue.js LSPWA - Первое приложение</title>

<script src="https://cdn.jsdelivr.net/npm/vue@2.6.14/dist/vue.js"></script>

</head>

<body>

<div id="app1">

<div :style="{ fontFamily: fontFamily, fontSize: fontSize + 'rem', color: fontColor }">

Привет, {{ name }}!

</div>

</div>

<script>

const app1 = new Vue({

el: '#app1',

data: {

name: 'Мир',

fontFamily: 'Courier New',

fontSize: 3,

fontColor: 'darkblue'

}

});

</script>

</body>

</html>

Результат работы кода можно увидеть на рисунке 3

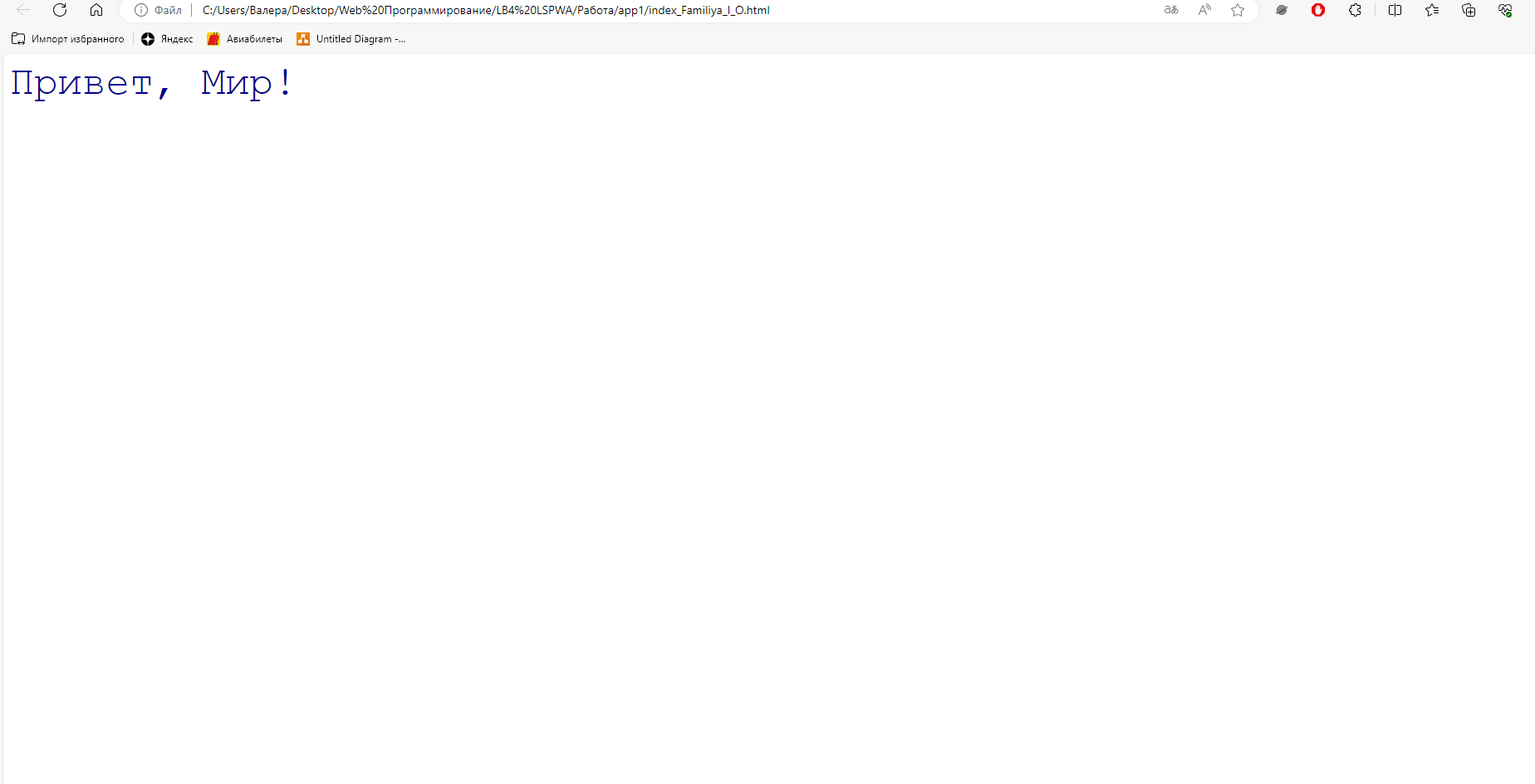


Рисунок . Отображение приложения 1.

**Код программы приложения 2:**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Vue.js LSPWA - Второе приложение</title>

<script src="https://cdn.jsdelivr.net/npm/vue@2.6.14/dist/vue.js"></script>

</head>

<body>

<div id="app2">

<input @keyup.alt="transliterate($event)">

<div>

Транслитерированный текст: {{ transliteratedText }}

</div>

</div>

<script>

const app2 = new Vue({

el: '#app2',

data: {

transliteratedText: ''

},

methods: {

transliterate(event) {

const cyrillicChar = event.key;

const transliterationMap = {

'а': 'a', 'б': 'b', 'в': 'v', 'г': 'g', 'д': 'd',

'е': 'e', 'ё': 'yo', 'ж': 'zh', 'з': 'z', 'и': 'i',

'й': 'y', 'к': 'k', 'л': 'l', 'м': 'm', 'н': 'n',

'о': 'o', 'п': 'p', 'р': 'r', 'с': 's', 'т': 't',

'у': 'u', 'ф': 'f', 'х': 'kh', 'ц': 'ts', 'ч': 'ch',

'ш': 'sh', 'щ': 'shch', 'ъ': '', 'ы': 'y', 'ь': '',

'э': 'e', 'ю': 'yu', 'я': 'ya'

};

if (transliterationMap[cyrillicChar.toLowerCase()]) {

this.transliteratedText += transliterationMap[cyrillicChar.toLowerCase()];

}

}

}

});

</script>

</body>

</html>

Результат работы кода можно увидеть на рисунках 4-5

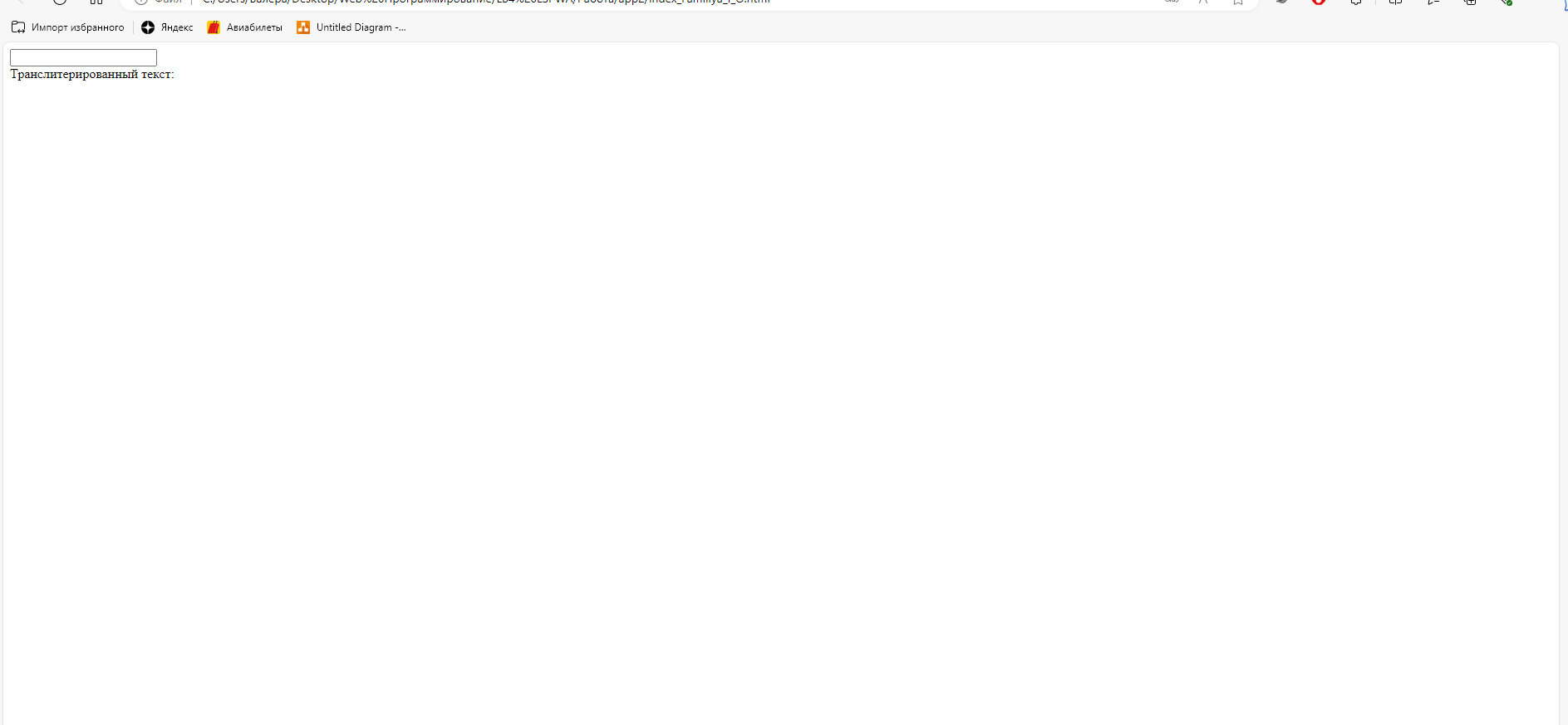


Рисунок . Открытие приложения.



Рисунок . Результат при вводе текста в поле при нажатом Alt

**4. Вывод**

В ходе выполнения задачи, было разработано два одностраничных веб-приложения (LSPWA) под управлением фреймворка Vue.js на языке JavaScript.