**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 №5

**«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-приложения (LSPWA) под управлением фреймворка Vue.js на языке JavaScript в соответствии с указаниями вариантов индивидуального задания.

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 связке (код веб-приложения + отображение в браузере (с учётом изменяющихся состояний) + Сеть Петри).

6. Содержательная часть по условной отрисовке (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, Vue.component при условной отрисовке.

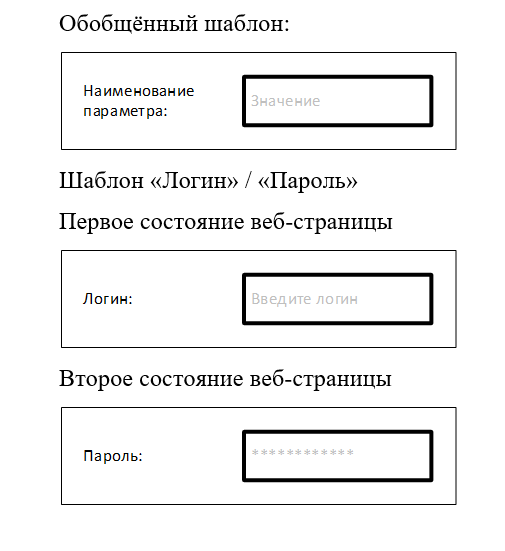
Реализовать:

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

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

v-model (Параметрическая связка) 9. В обрамлённой точками области масштабируемой векторной графики размером 420 на 576 пикселей по ширине и высоте, соответственно, в псевдослучайно сгенерированной позиции наносится текстовая надпись. Стрелками курсора на клавиатуре менять с шагом в 5 пикселей положение надписи посредством пробелов («Spacebar») в горизонтальном направлении, посредством «Enter» в вертикальном направлении. Противоположные направления по горизонтали и вертикали задаются маской по клавише «Shift». Не позволять надписи подступать к границам ближе, чем на 5 пикселей даже при инициализации начального положения. Текст надписи задаётся через элемент, расположенный под <svg></svg>..

v-if( Условная отрисовка/шаблоны) 9. Предусмотреть «убегающую» при наведении на неё курсором мыши экранную кнопку. Предусмотреть вторую экранную кнопку, обеспечивающую отключение режима «бегства» первой. Первая экранная кнопка отвечает за переключение режимов структуры одностраничного веб-приложения, представленного в Приложении(прикрепленное фото). Вводятся «Login» / «First N



**3. Ход работы**

Построим сеть Петри первого приложения:

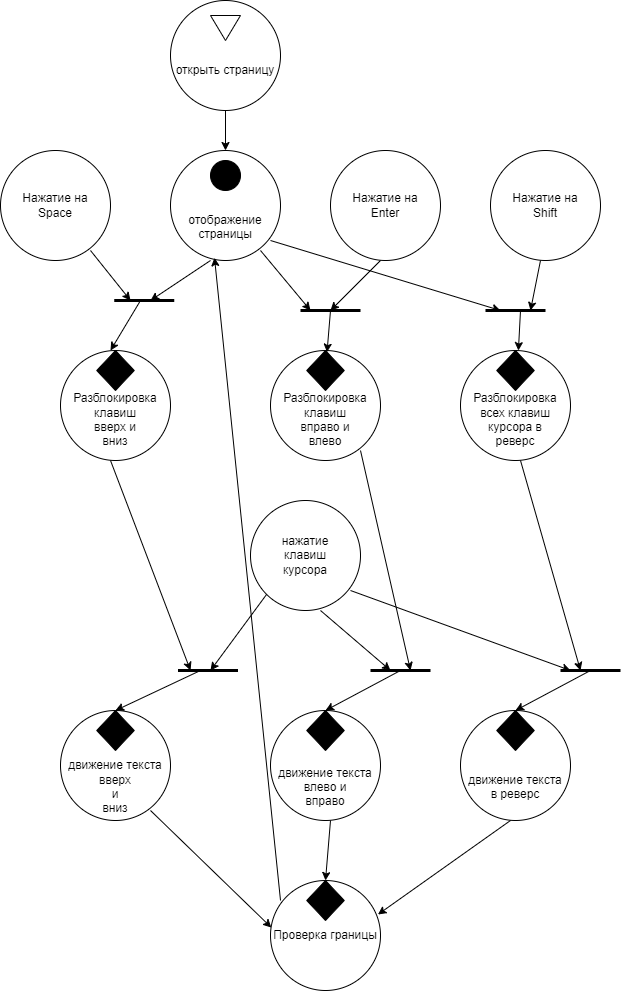


Рисунок . Сеть Петри приложения 1

Построим сеть Петри второго приложения:

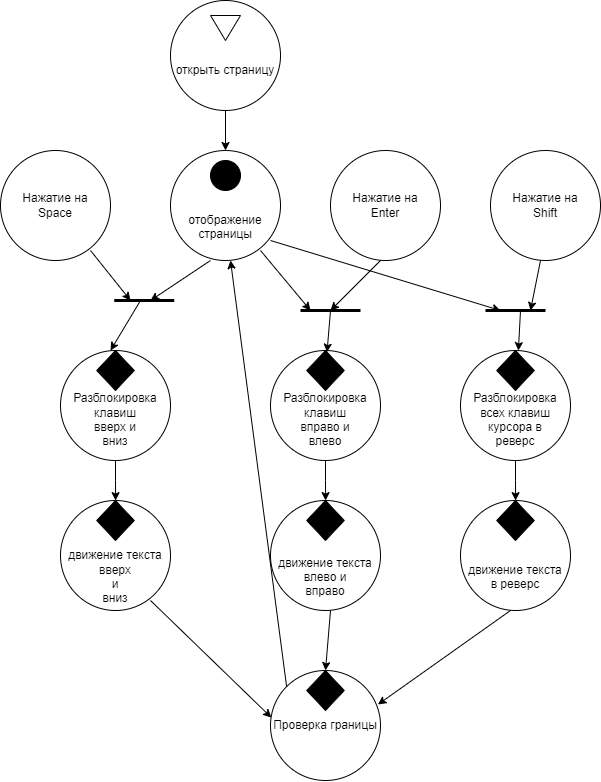


Рисунок Сеть Петри второго приложения

**Код программы приложения 1:**

<!DOCTYPE html>

<html>

<head>

<title>Приложение 1</title>

<script src="https://cdn.jsdelivr.net/npm/vue@2"></script>

<style>

**#app** {

position: relative;

width: 420px;

height: 576px;

border: 1px solid black;

}

**.text** {

position: absolute;

user-select: none;

}

</style>

</head>

<body>

<div id="app">

<div class="text" :style="{ left: `${x}px`, top: `${y}px` }">{{ message }}</div>

</div>

<script>

new Vue({

el: '#app',

data: {

x: Math.random() \* (420 - 50 - 5) + 5,

y: Math.random() \* (576 - 50 - 5) + 5,

message: 'Hello Vue.js!',

spacePressed: false,

enterPressed: false

},

mounted() {

window.addEventListener('keydown', this.handleKeydown);

window.addEventListener('keyup', this.handleKeyup);

},

beforeDestroy() {

window.removeEventListener('keydown', this.handleKeydown);

window.removeEventListener('keyup', this.handleKeyup);

},

methods: {

handleKeydown(event) {

const step = 5;

if (event.code === 'Space') this.spacePressed = true;

if (event.code === 'Enter') this.enterPressed = true;

switch (event.key) {

case 'ArrowUp':

if (this.spacePressed) this.y = Math.max(this.y - step, 5);

else if (event.shiftKey) this.y = Math.min(this.y + step, 576 - 50 - 5);

break;

case 'ArrowDown':

if (this.spacePressed) this.y = Math.min(this.y + step, 576 - 50 - 5);

else if (event.shiftKey) this.y = Math.max(this.y - step, 5);

break;

case 'ArrowLeft':

if (this.enterPressed) this.x = Math.max(this.x - step, 5);

else if (event.shiftKey) this.x = Math.min(this.x + step, 420 - 50 - 5);

break;

case 'ArrowRight':

if (this.enterPressed) this.x = Math.min(this.x + step, 420 - 50 - 5);

else if (event.shiftKey) this.x = Math.max(this.x - step, 5);

break;

}

},

handleKeyup(event) {

if (event.code === 'Space') this.spacePressed = false;

if (event.code === 'Enter') this.enterPressed = false;

}

}

});

</script>

</body>

</html>

Результат работы кода можно увидеть на рисунке 3

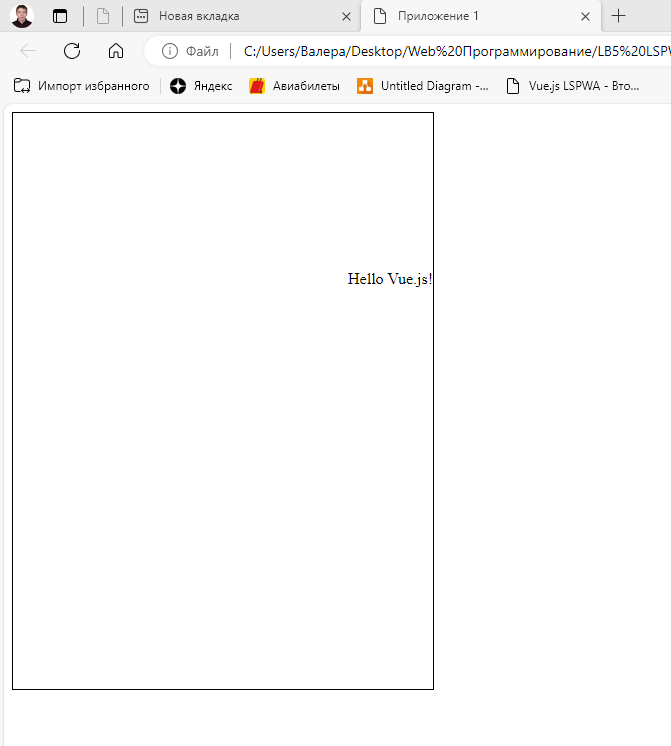


Рисунок . Отображение приложения 1.

**Код программы приложения 2:**

<!DOCTYPE html>

<html>

<head>

<title>Приложение 2</title>

<script src="https://cdn.jsdelivr.net/npm/vue@2"></script>

<style>

**.button** {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

transition: .5s ease;

}

**.button:hover** {

cursor: pointer;

transform: translate(100px, 100px);

}

**.button.stop:hover** {

transform: none;

}

</style>

</head>

<body>

<div id="app">

<button class="button" :class="{ stop: isStop }" @mouseover="moveButton" @click="isFormVisible = !isFormVisible">Переключить режим</button>

<button @click="isStop = !isStop">Отключить режим бегства</button>

<div v-if="isFormVisible">

<input type="text" v-model="login" placeholder="Login">

<input type="text" v-model="firstName" placeholder="First Name">

<input type="text" v-model="lastName" placeholder="Last Name">

<input type="number" v-model="age" placeholder="Age">

<input type="email" v-model="email" placeholder="E-mail">

<select v-if="isAllFieldsFilled" v-model="selectedOption">

<option disabled value="">Выберите опцию</option>

<option>Скрыть</option>

<option>Отобразить</option>

</select>

<table v-if="selectedOption === 'Отобразить'">

<tr><td>Login:</td><td>{{ login }}</td></tr>

<tr><td>First Name:</td><td>{{ firstName }}</td></tr>

<tr><td>Last Name:</td><td>{{ lastName }}</td></tr>

<tr><td>Age:</td><td>{{ age }}</td></tr>

<tr><td>E-mail:</td><td>{{ email }}</td></tr>

</table>

</div>

</div>

<script>

new Vue({

el: '#app',

data: {

isStop: false,

isFormVisible: false,

login: '',

firstName: '',

lastName: '',

age: null,

email: '',

selectedOption: ''

},

methods: {

moveButton(event) {

if (!this.isStop) {

event.target.style.transform = `translate(${Math.random() \* 200}px, ${Math.random() \* 200}px)`;

}

}

},

computed: {

isAllFieldsFilled() {

return this.login && this.firstName && this.lastName && this.age && this.email;

}

},

watch: {

selectedOption(value) {

if (value === 'Скрыть') {

this.isFormVisible = false;

} else if (value === 'Отобразить') {

this.isFormVisible = true;

}

}

}

});

</script>

</body>

</html>

Результат работы кода можно увидеть на рисунках 4-5

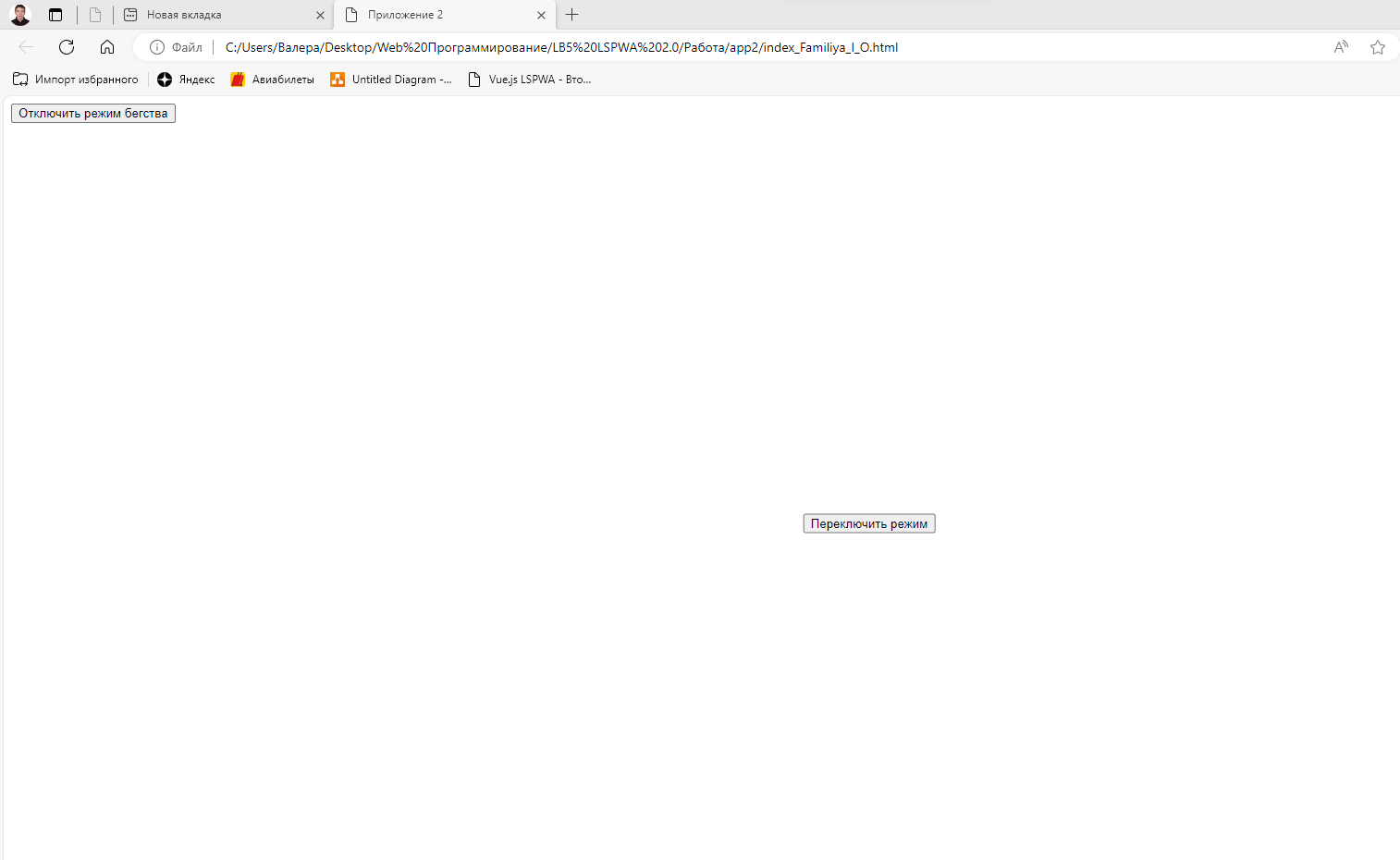


Рисунок . Открытие приложения.

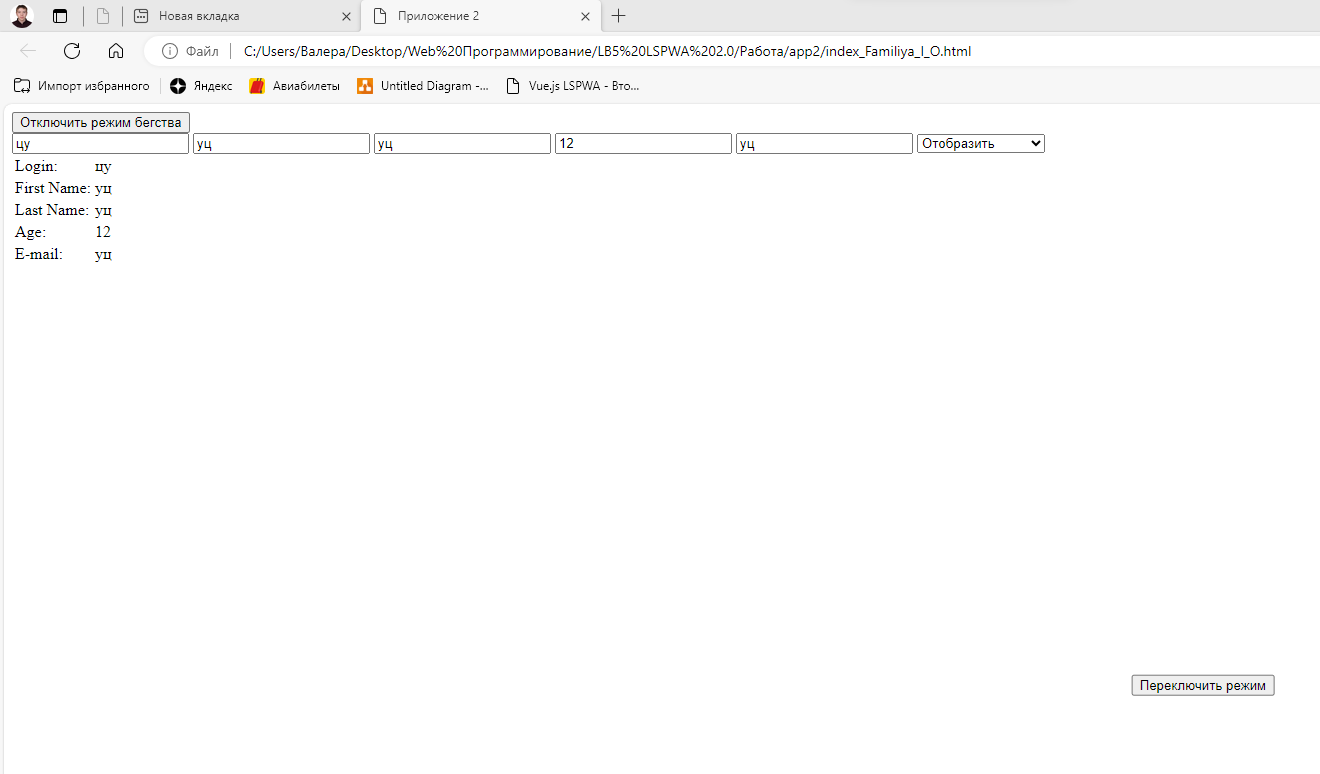


Рисунок . Результат при нажатии клавиши переключить режим и ввода данных в поля

**4. Вывод**

В ходе выполнения данного задания были разработаны два одностраничных веб-приложения на фреймворке Vue.js. В процессе работы были использованы такие возможности Vue.js, как параметрическая связка (v-model) и условная отрисовка (v-if).

В первом приложении реализована возможность перемещения текстовой надписи в области векторной графики с помощью клавиатуры. Во втором приложении создана “убегающая” кнопка, которая меняет свое положение при наведении курсора мыши.

В целом, работа над заданием позволила углубить знания и навыки работы с фреймворком Vue.js и его возможностями. Это был интересный и полезный опыт.