ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»  
(РУТ (МИИТ))

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

ОТЧЁТ  
ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

По дисциплине «Web-программирование»

ВАРИАНТ 4

Выполнили: ст. гр. ТКИ-541

Поткин Дмитрий

Галыба Леонид

Проверил: к.т.н., доц. Сафронов А.И.

Москва 2024

1. Цель работы

Разработать четыре простых локальных одностраничных web-приложение (LSPWA) под управлением фреймворка Vue.js на языке JavaScript в соответствии с указаниями вариантов индивидуального задания (покрываются три раздела последовательного вычислительного процесса и один раздел разветвляющегося вычислительного процесса).

1. Формулировка задачи

**Реализовать:**

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

– в файле с именем index01\_Familiya\_I\_O1\_Familiya\_I\_O2.html параметрическую вставку;

– в файле с именем index02\_Familiya\_I\_O1\_Familiya\_I\_O2.html параметрическую связку;

– в файле с именем index03\_Familiya\_I\_O1\_Familiya\_I\_O2.html работу с методами и событиями;

– в файле с именем index04\_Familiya\_I\_O1\_Familiya\_I\_O2.html условную отрисовку.

**Задания по варианту:**

* Настроить при помощи параметрической подстановки тег рамки (фрейма) *<iframe></iframe>*. Параметрически выводить во фрейм сайт-визитку, созданный в Задании 2. Параметрически убрать границы (*border*) и прокрутку (*scroll*) фрейма. Внутреннее содержимое фрейма параметрически сдвинуть на одинаковую величину в 50 пикселей. Ширину и высоту фрейма принять 300 и 400 пикселей, соответственно.
* Предусмотреть блок <*div*></*div*>, в котором по двойному щелчку кнопкой мыши появляется надпись «Мой текст». Двойной щелчок кнопкой мыши при маскировании клавишей «*Shift*» последовательно реализует смену состояний выравнивания текста в блоке <*div*></*div*> сначала по центру, затем снова по левому краю.
* Блок <*div*><*/div*> фиксированных размеров 500 на 500 пикселей по ширине и высоте центрирован в браузерном окне. Предусмотреть изменение посредством нажатия на стрелки курсора на клавиатуре («←», «↑», «↓», «→») выравнивание текста в блоке <*div*><*/div>* последовательно как по горизонтали (слева, посередине, справа), так и по вертикали (сверху, посередине, снизу). Учесть тот факт, что для выравнивания содержимого по вертикали в блоке <*div*><*/div>* отсутствуют специальные перечислимые значения и параметр управления. Реализовать соответствующее выравнивание следует средствами настройки стилевого оформления тега. Переход через границы блока <*div*><*/div>* не предусматривать. При размещении текста у границ блока – не реагировать на нажатие стрелок курсора на клавиатуре.
* Подготовить форму авторизации пользователя по логину и паролю. Предусмотреть несколько режимов входа. С параметрами «*test*» – «*test*» можно только посмотреть на заложенные в программу Фамилию, Имя, Отчество разработчика, а также название и номер учебной группы, которые по умолчанию не заложены в программу. В режиме «*admin*» – «*admin*» иметь возможность отредактировать Фамилию, Имя и Отчество автора. В режиме «*user*» – «*user*» иметь возможность ввода названия и номера учебной группы. Иметь возможность выхода к форме авторизации из каждого режима.

1. Технология локального подключения Vue.js

В работе была использована vue.js 3 версии.

К каждому файлу задания была подключена сборка vue.js, установленная с источника <https://unpkg.com/vue@3/dist/vue.global.js>.

1. Таблица соотвествия переменных и методов, используемых в web-приложении

|  |  |  |
| --- | --- | --- |
| Задание | Переменная/метод | Описание |
| 1 | frameSrc | Путь к локальному сайту-визитке |
| 1 | frameWidth | Ширина фрейма |
| 1 | frameHeight | Высота фрейма |
| 1 | frameBorder | Рамка фрейма |
| 1 | frameScrolling | Прокрутка страницы |
| 2 | displayText | Текст, который будет отображаться до нажатия ЛКМ |
| 2 | textAlign | Выравнивание текста |
| 3 | horizontalAlign | Выравнивание текста по горизонтали |
| 3 | verticalAlign | Выравнивание текста по вертикали |
| 3 | handleKeydownLeft | Обработка нажатия стрелки «влево» |
| 3 | handleKeydownRight | Обработка нажатия стрелки «вправо» |
| 3 | handleKeydownUp | Обработка нажатия стрелки «вверх» |
| 3 | handleKeydownDown | Обработка нажатия стрелки «вниз» |
| 4 | username | Имя для режима входа |
| 4 | password | Пароль для режима входа |
| 4  4 | lastName  firstName | Фамилия разработчика  Имя разработчика |
| 4 | middleName | Отчества разработчика |
| 4 | groupName | Название группы |
| 4 | groupNumber | Номер группы |
| 4 | logout | Обработка выхода из системы |

1. Содержательная часть по параметрической вставке
   1. Код web-приложения

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Vue.js IFrame Example</title>

<script src="vue.global.js"></script> <!-- Локальное подключение Vue.js -->

</head>

<body>

<div id="app">

<iframe

:style="{padding: '50px'}"

:src="frameSrc"

:width="frameWidth"

:height="frameHeight"

:frameborder="frameBorder"

:scrolling="frameScrolling">

</iframe>

</div>

<script>

const app = Vue.createApp({

data() {

return {

frameSrc: 'site\_card.html', // путь к локальному сайту-визитке

frameWidth: 300,

frameHeight: 400,

frameBorder: 0, // убираем границу

frameScrolling: 'no', // убираем прокрутку

};

},

});

app.mount('#app');

</script>

</body>

</html>

* 1. Отображение в браузере

|  |
| --- |
|  |
| 1. – Отображение в браузере |

1. Содержательная часть по параметрической связке
   1. Код web-приложения

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Vue.js Double Click Example</title>

<script src="vue.global.js"></script> <!-- Локальное подключение Vue.js -->

</head>

<body>

<div id="app">

<div

@dblclick.shift="() => this.textAlign = this.textAlign === 'left' ? 'center' : 'left'"

@dblclick="() => this.displayText = 'Мой текст'"

:style="{ textAlign: textAlign, border: '1px solid black', padding: '10px', width: '200px', height: '100px' }"

>

{{ displayText }}

</div>

</div>

<script>

const app = Vue.createApp({

data() {

return {

displayText: 'Нажмите ЛКМ дважды', // Текст, который будет отображаться вначале

textAlign: 'left', // Начальное выравнивание текста

};

},

});

app.mount('#app');

</script>

</body>

</html>

* 1. Отображение в браузере

|  |
| --- |
|  |
| 1. – Начальное состояние |

|  |
| --- |
|  |
| 1. – Состояние после нажатия ЛКМ |

|  |
| --- |
|  |
| 1. – Состояние после нажатия ЛКМ + Shift |

1. Содержательная часть по методам и событиям
   1. Код web-приложения

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Centered Div</title>

<script src="vue.global.js"></script> <!-- Подключение Vue.js -->

</head>

<body>

<div id="app">

<div :style="[{display: 'flex'}, {justifyContent: 'center'}, {alignItems: 'center'}, {height: '98vh'}]">

<div

:style="blockStyle"

tabindex="0"

@keydown.left.prevent="handleKeydownLeft"

@keydown.right.prevent="handleKeydownRight"

@keydown.up.prevent="handleKeydownUp"

@keydown.down.prevent="handleKeydownDown"

>

<p>My text</p>

</div>

</div>

</div>

<script>

//flex-start, center, flex-end

const app = Vue.createApp({

data() {

return {

// Переменные для управления выравниванием

horizontalAlign: 'center',

verticalAlign: 'center',

};

},

computed: {

// Стиль для блока

blockStyle() {

return {

width: '500px',

height: '500px',

display: 'flex',

justifyContent: this.horizontalAlign, // выравнивание по горизонтали

alignItems: this.verticalAlign, // выравнивание по вертикали

border: '1px solid black'

};

},

},

methods: {

handleKeydownLeft() {

if (this.horizontalAlign === 'center') this.horizontalAlign = 'flex-start'

else if (this.horizontalAlign === 'flex-end') this.horizontalAlign = 'center'

},

handleKeydownRight() {

if (this.horizontalAlign === 'center') {

this.horizontalAlign = 'flex-end'

}

else if (this.horizontalAlign === 'flex-start') {

this.horizontalAlign = 'center'

}

},

handleKeydownUp() {

if (this.verticalAlign === 'center') {

this.verticalAlign = 'flex-start'

}

else if (this.verticalAlign === 'flex-end') {

this.verticalAlign = 'center'

}

},

handleKeydownDown() {

if (this.verticalAlign === 'center') {

this.verticalAlign = 'flex-end'

}

else if (this.verticalAlign === 'flex-start') {

this.verticalAlign = 'center'

}

},

}

});

app.mount('#app');

</script>

</body>

</html>

* 1. Отображение в браузере

|  |
| --- |
|  |
| 1. – Начальное состояние |

|  |
| --- |
|  |
| 1. – Состояние после нажатия клавиши «вверх» |

|  |
| --- |
|  |
| 1. – Состояние после нажатия клавиши «вниз» |

|  |
| --- |
|  |
| 1. – Состояние после нажатия клавиши «влево» |
|  |
| 1. – Состояние после нажатия клавиши «вправо» |

* 1. Сеть Петри

|  |
| --- |
|  |
| 1. – Полная сеть Петри |

1. Содержательная часть по условной отрисовке
   1. Код web-приложения

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Authorization Form</title>

<script src="vue.global.js"></script> <!-- Подключение Vue.js -->

</head>

<body>

<div id="app">

<div>

<h2>Форма авторизации</h2>

<input type="text" v-model="username" placeholder="Логин" />

<input type="password" v-model="password" placeholder="Пароль" />

</div>

<div>

<div v-if="(username === 'test' && password === 'test') | (username === 'admin' && password === 'admin') | (username === 'user' && password === 'user')">

<h2>Добро пожаловать, {{ username }}</h2>

<button v-on:click="logout">Выход</button>

</div>

<!-- Режим "test" -->

<div v-if="username === 'test' && password === 'test'">

<h3>Информация о разработчике:</h3>

<p>Фамилия: {{ developerInfo.lastName }}</p>

<p>Имя: {{ developerInfo.firstName }}</p>

<p>Отчество: {{ developerInfo.middleName }}</p>

<p>Учебная группа: {{ developerInfo.groupName }}</p>

<p>Номер группы: {{ developerInfo.groupNumber }}</p>

</div>

<!-- Режим "admin" -->

<div v-else-if="username === 'admin' && password === 'admin'">

<h3>Редактирование информации о разработчике:</h3>

<input type="text" v-model="developerInfo.lastName" placeholder="Фамилия" />

<input type="text" v-model="developerInfo.firstName" placeholder="Имя" />

<input type="text" v-model="developerInfo.middleName" placeholder="Отчество" />

<button v-on:click="saveAdminInfo">Сохранить</button>

</div>

<!-- Режим "user" -->

<div v-else-if="username === 'user' && password === 'user'">

<h3>Ввод информации об учебной группе:</h3>

<input type="text" v-model="developerInfo.groupName" placeholder="Название группы" />

<input type="text" v-model="developerInfo.groupNumber" placeholder="Номер группы" />

<button v-on:click="saveUserInfo">Сохранить</button>

</div>

</div>

</div>

<script>

const app = Vue.createApp({

data() {

return {

username: '',

password: '',

developerInfo: {

lastName: 'Potkin',

firstName: 'Dmitry',

middleName: 'Alekseevich',

groupName: '',

groupNumber: ''

}

};

},

methods: {

logout() {

this.username = '';

this.password = '';

},

}

});

app.mount('#app');

</script>

</body>

</html>

* 1. Отображение в браузере

|  |
| --- |
|  |
| 1. – Начальное состояние |

|  |
| --- |
|  |
| 1. – Вход в режим ‘test’ – ‘test’ |

|  |
| --- |
|  |
| 1. – Вход в режим ‘user’ – ‘user’ |

|  |
| --- |
|  |
| 1. – Добавление информации об учебной группе в режиме ‘user’ – ‘user’ |
|  |
| 1. – Вход в режим ‘admin’ – ‘admin’ |

|  |
| --- |
|  |
| 1. – Изменение информации о разработчике в режиме ‘admin’ – ‘admin’ |

* 1. Сеть Петри

|  |
| --- |
|  |
| 1. – Полная сеть Петри |

1. Вывод

В результате проведенной работы были разработаны четыре простых локальных одностраничных web-приложение (LSPWA) под управлением фреймворка Vue.js на языке JavaScript.