**ЛАБОРАТОРНАЯ РАБОТА №6**

**«ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ ФРЕЙМВОРКА VUE.JS»**

**Цель работы**

Изучить принцип работы с формами и их методы валидации.

**Постановка задачи**

Разработать дизайн, создать веб-приложение, содержащее форму с необходимыми полями. Реализовать валидацию формы. При нажатии на кнопку отправить и выводить сообщение.

Номер зачетной книжки: 399228.

**Ход работы**

Для данной работы было разработано одностраничное приложение при помощи фреймворка Vue.js (рисунки 1-2).

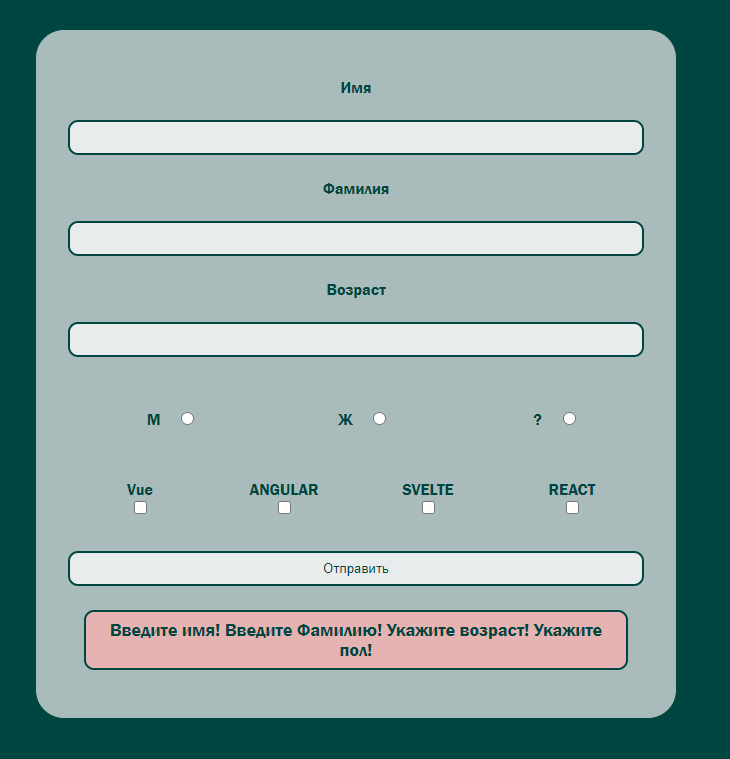


Рисунок 1 – Незаполненная форма

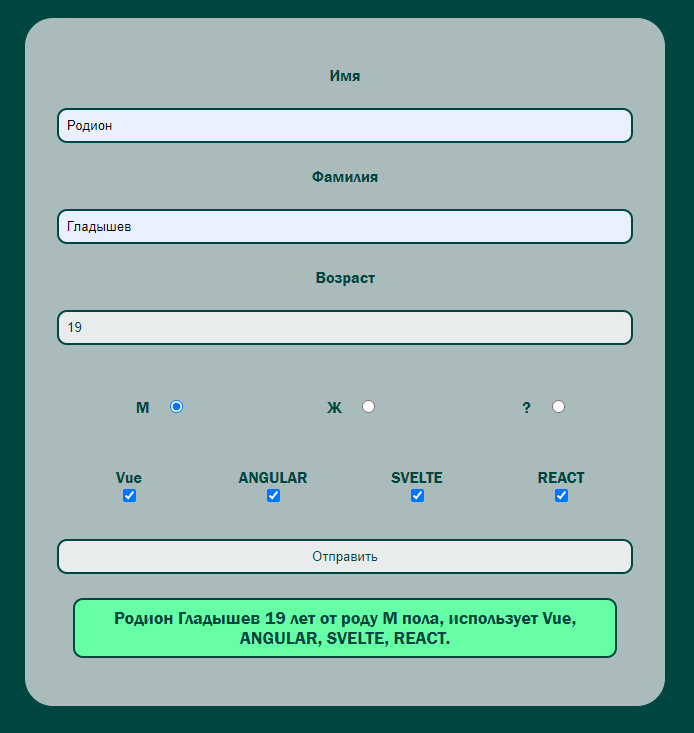


Рисунок 2 – Форма с заполненными значениями

При отправке формы происходит переадресация на сервис электронной почты и генерируется сообщение, содержащее введённые в форму данные (рисунок 3).

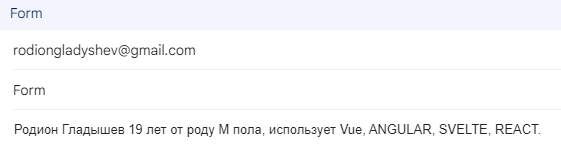


Рисунок 3 – Письмо, сгенерированное при отправке формы

Весь исходный код для создания сайта представлен в приложении А.

**Выводы**

В ходе выполнения лабораторной работы был изучены принципы работы с формами и их методы валидации на примере создания веб-приложения с помощью фреймворка Vue.js.

**ПРИЛОЖЕНИЕ А**

**Код страницы**

Листинг 1 – Текст исходного файла App.vue

<script>

    export default {

        data: () => ({

            formData: {

                name: "",

                surname: "",

                age: null,

                sex: "",

                frameworks:[]

            },

            sexOptions: Object.freeze({

                M: "М",

                F: "Ж",

                S: "?"

            }),

            frameWorksOptions: Object.freeze({

                VUE: "Vue",

                ANGULAR: "ANGULAR",

                SVELTE: "SVELTE",

                REACT: "REACT"

            }),

        }),

        computed: {

            errors() {

                let errors = [];

                if (!this.formData.name) errors.push("Введите имя! ");

                else errors.filter((error) => error.field !== "name");

                if (!this.formData.surname) errors.push("Введите Фамилию! ");

                else errors.filter((error) => error.field !== "surname");

                if ((!this.formData.age && typeof this.formData.age !== "number") || (this.formData.age < 1)) errors.push("Укажите возраст! " );

                else errors.filter((error) => error.field !== "age");

                if (!this.formData.sex) errors.push("Укажите пол! ");

                else errors.filter((error) => error.field !== "sex");

                let result = Array.from(errors.values()).join(" ");

                return result;

            },

            result() {

                return ""

                  + this.formData.name + " "

                  + this.formData.surname + " "

                  + this.formData.age + " лет от роду "

                  + this.formData.sex + " пола"

                  + ((this.formData.frameworks != "") ? ", использует " + this.formData.frameworks.join(", ") : "")

                  + ".";

            }

        },

        methods: {

            checkAndSubmitForm: function(event) {

                if (this.errors != "") event.preventDefault();

                else {

                  window.location.href = "mailto:rodiongladyshev@gmail.com?subject=Form&body=" + this.result;

                }

            }

        }

    }

</script>

<template>

    <form novalidate="true">

        <label for="firstname"> Имя </label>

        <input type="text" v-model="formData.name" id="firstname"/>

        <label for="surname"> Фамилия </label>

        <input type="text" v-model="formData.surname" id="surname"/>

        <label for="age"> Возраст </label>

        <input type="number" v-model="formData.age" id="age"/>

        <br><br>

        <div style="display: flex; text-align: center">

            <div v-for="option in sexOptions" :key='option' style="flex: 1">

                <label :for="option">{{option}}</label>

                <input type="radio" :value="option" v-model="formData.sex" id="sex">

            </div>

        </div>

        <br><br>

        <div style="display: flex; text-align: center">

            <div v-for="option in frameWorksOptions" :key='option' style="flex: 1">

                <label :for="option">{{option}}</label>

                <br>

                <input type="checkbox" :value="option" v-model="formData.frameworks" id="frameworks">

            </div>

        </div>

        <br>

        <input type="button" value="Отправить" id="submitButton" @click="checkAndSubmitForm">

        <p v-if="errors != ''" id="resultError"> {{errors}} </p>

        <p v-else id="resultOk"> {{result}} </p>

    </form>

</template>

Листинг 2 – Текст исходного файла main.scss

$backgroundColor: #a9bcbb;

$secondaryColor: #e8ecec;

$primaryColor: #004640;

$colorAccept: #66FFA5;

$colorDeny: #E6B3B3;

$font-face-gothic: 'Franklin Gothic', 'Arial Narrow', Arial, sans-serif;

body {

    font-family: $font-face-gothic;

    background-color: $primaryColor !important;

    margin: 0;

    overflow: hidden;

}

#app {

    position: fixed;

    top: 0;

    left: 0;

    bottom: 0;

    right: 0;

}

form {

    display: flex;

    flex-direction: column;

    gap: 0.5rem;

    min-width: 30%;

    max-width: 30%;

    max-height: 90%;

    text-align: center;

    padding: 2rem;

    margin: auto;

    margin-top: 2%;

    border-radius: 29px;

    border: 1px solid $primaryColor;

    background-color: $backgroundColor;

}

input {

    padding: 0.5rem;

    border-radius: 10px;

    border: 2px solid $primaryColor;

    background-color: $secondaryColor;

    color: $primaryColor;

}

label, p {

    margin: 1rem;

    color: $primaryColor;

    font-weight: bold;

}

%result {

    padding: 0.5rem;

    word-wrap: normal;

    font-size: large;

    color: $primaryColor;

    border-radius: 10px;

    border: 2px solid $primaryColor;

}

#resultError {

    @extend %result;

    background-color: $colorDeny;

}

#resultOk {

    @extend %result;

    background-color: $colorAccept;

}

Листинг 3 – Текст исходного файла main.js

import { createApp } from 'vue'

import App from './App.vue'

import './main.css'

createApp(App).mount('#app')

Листинг 4 – Текст исходного файла index.html

<!DOCTYPE html>

<html lang="ru">

  <head>

    <meta charset="UTF-8">

    <link rel="icon" href="/favicon.ico">

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

    <title>Vite App</title>

  </head>

  <body>

    <div id="app"></div>

    <script type="module" src="/src/main.js"></script>

  </body>

</html>