

6 ЛАБОРАТОРНАЯ РАБОТА №6

«Принцип работы с формами и их методы валидации на Vue.js»

6.1 Цель работы

Изучить принцип работы с формами и их методы валидации на Vue.js.

6.3 Ход выполнения работы

6.3.1 В ходе выполнения лабораторной работы после изучения методических указаний была сделана страница с формой. Текст файла, содержащего код для формы представлен в листинге 6.1.

Листинг 6.1 – Код для страницы формы

```
<template>
  <div class="form-container">
    <form @submit.prevent="handleSubmit" class="form">
      <div class="form-group">
        <label for="name" class="form-label">Имя</label>
        <input type="text" v-model="formData.name" id="name" class="form-control">
        <div class="invalid-feedback" v-if="formErrors.name">{{
formErrors.name }}</div>
      </div>
      <div class="form-group">
        <label for="surname" class="form-label">Фамилия</label>
        <input type="text" v-model="formData.surname" id="surname"
class="form-control">
        <div class="invalid-feedback" v-if="formErrors.surname">{{
formErrors.surname }}</div>
      </div>
      <div class="form-group">
        <label for="age" class="form-label">Возраст</label>
```

```

      <input type="number" v-model="formData.age" id="age" class="form-
control">

      <div      class="invalid-feedback"      v-if="formErrors.age">{{
formErrors.age }}</div>

    </div>

    <div class="form-group">

      <label class="form-label">Пол</label>

      <div class="form-check">

        <label v-for="option in sexOptions" :key="option" class="form-
check-label">

          <input type="radio" v-model="formData.sex" :value="option"
class="form-check-input"> {{ option }}

        </label>

        <div      class="invalid-feedback"      v-if="formErrors.sex">{{
formErrors.sex }}</div>

      </div>

    </div>

    <div class="form-group">

      <label class="form-label">Фреймворки</label>

      <div class="form-check">

        <label v-for="option in frameworksOptions" :key="option"
class="form-check-label">

          <input      type="checkbox"      v-model="formData.frameworks"
:value="option" class="form-check-input"> {{ option }}

        </label>

        <div class="invalid-feedback" v-if="formErrors.frameworks">{{
formErrors.frameworks }}</div>

      </div>

    </div>

    <button type="submit" class="btn">Отправить</button>

    <div :class="['alert', classOption]">Данные отправлены!</div>

  </form>

</div>

</template>

<script>
export default {
  name: 'FormComponent',
  data() {

```

```

return {
  formData: {
    name: '',
    surname: '',
    age: null,
    sex: '',
    frameworks: []
  },
  sexOptions: ['Мужской', 'Женский'],
  frameworksOptions: ['Vue', 'React', 'Angular'],
  formErrors: {}
};
},
computed: {
  classOption() {
    return this.$store.state.classOption;
  }
},
methods: {
  handleSubmit() {
    this.formErrors = this.validateForm();
    if (Object.keys(this.formErrors).length === 0) {
      this.$store.dispatch('updateClassOption', 'visible');
    } else {
      this.$store.dispatch('updateClassOption', 'hidden');
    }
  },
  validateForm() {
    const errors = {};
    if (!this.formData.name) {
      errors.name = 'Имя обязательно';
    }
    if (!this.formData.surname) {
      errors.surname = 'Фамилия обязательна';
    }
    if (!this.formData.age || this.formData.age < 0) {

```

```

        errors.age = 'Введите корректный возраст';
    }
    if (!this.formData.sex) {
        errors.sex = 'Выберите пол';
    }
    if (this.formData.frameworks.length === 0) {
        errors.frameworks = 'Выберите хотя бы один фреймворк';
    }
    return errors;
}
},
created() {
    this.$store.dispatch('updateClassOption', 'hidden');
}
};
</script>

```

```

<style>
.hidden {
    visibility: hidden;
}

.visible {
    visibility: visible;
}

.form-container {
    display: flex;
    align-items: center;
    justify-content: center;
    height: 100vh;
}

.form {
    width: 25%;
    margin: auto;
}

```

```
}

.form-group {
  margin-bottom: 16px;
}

.form-label {
  display: block;
  font-size: 1vw;
  margin-bottom: 8px;
}

.form-control {
  display: block;
  width: 100%;
  padding: 0.375rem 0.75rem;
  font-size: 1vw;
  line-height: 1.5;
  color: #495057;
  background-color: #fff;
  background-clip: padding-box;
  border: 1px solid #ced4da;
  border-radius: 0.25rem;
  transition: border-color 0.15s ease-in-out;
}

.form-control:focus {
  color: #495057;
  background-color: #fff;
  border-color: #41B883;
  outline: 0;
  box-shadow: 0 0 0 0 #41B883;
}

.invalid-feedback {
  display: block;
```

```
width: 100%;
margin-top: 0.25rem;
font-size: 1vw;
color: #dc3545;
}

.form-check {
  margin-top: 8px;
  padding: 0;
}

.form-check-label {
  font-size: 1vw;
  margin-right: 20px;
  margin-left: 50px;
}

.form-check-input {
  margin-right: 8px;
}

.btn {
  display: inline-block;
  padding: 10px 20px;
  font-size: 1vw;
  color: #fff;
  background-color: #41B883;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  margin-top: 20px;
}

.btn:hover {
  background-color: #155724;
  color: #fff;
}
```

```
}

.alert {
  margin-top: 20px;
  font-size: 1vw;
  padding: 10px;
  border-radius: 4px;
}

.alert.hidden {
  display: none;
}

.alert.visible {
  display: block;
  background-color: #d4edda;
  color: #155724;
  border: 1px solid #c3e6cb;
}
</style>
```

На рисунке 6.1 можно увидеть готовый результат формы.

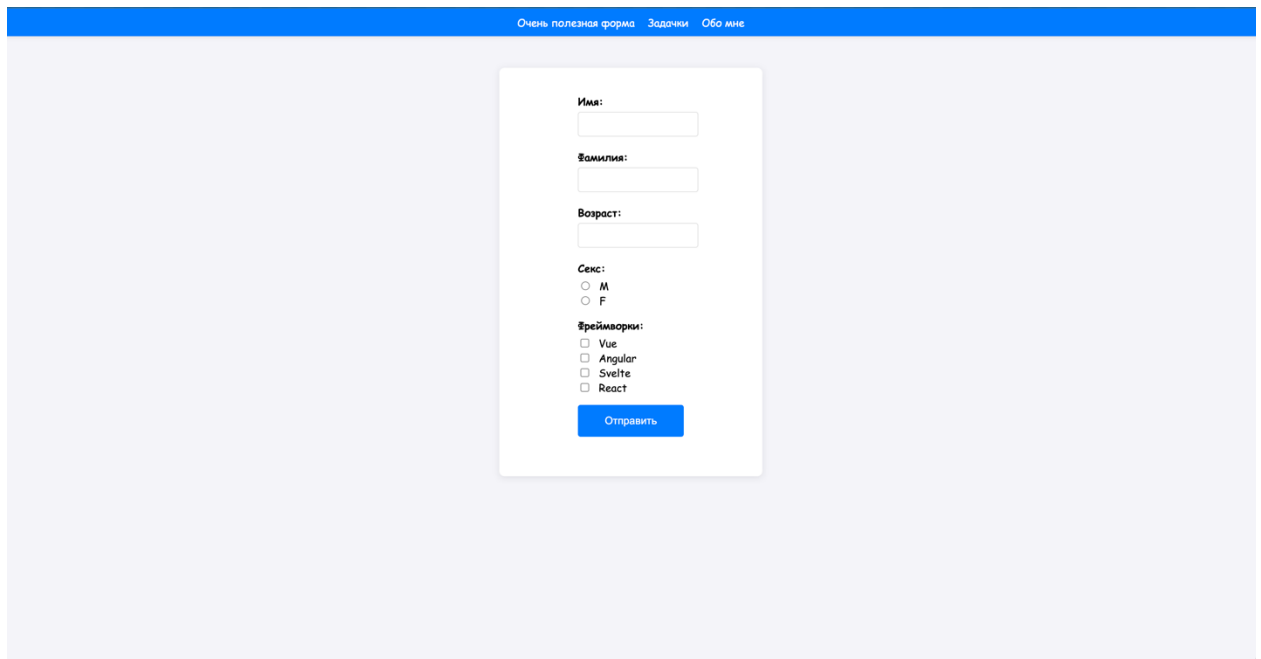


Рисунок 6.1 – Внешний вид формы

На рисунке 6.2 показана форма при неправильном вводе.

Возраст:

-3

Возраст должен быть положительным числом

Рисунок 6.2 – Внешний вид формы при неправильном вводе

Выводы

В ходе лабораторной работе был изучен принцип работы с формами и их методы валидации на Vue.js. Был разработан дизайн, создано веб-приложение, содержащее форму с указанными в ходе работы полями. Также была реализована валидация формы. При нажатии на кнопку отправить и корректно заполненной форме выводится сообщение. В конце выполнения лабораторной работы был написан отчет.