

1

6 Лабораторная работа №6 Формы в VueJS

6.1. Цель работы

Изучить принцип работы с формами и их методы валидации.

6.2. Постановка задачи

Разработать дизайн, создать веб-приложение, содержащее форму с указанными в ходе работы полями. Реализовать валидацию формы. При нажатии на кнопку отправить и корректно заполненной форме выводить сообщение.

6.3. Ход выполнения работы

В ходе выполнения лабораторной работы был написан код, представленный в листинге 1. Также в существующую страницу были внесены дополнения, представленные в листинге 2. При запросе готовой страницы у веб-сервера была получена форма, представленная на рисунке 1.

Листинг 1 — Код скрипта horrible-form.js

```
import { createApp } from "../vue.esm-browser.js";
import { HorriblePersonData } from "../HorriblePersonData.esm.js";

const app = createApp({
  data() {
    {
      return {
        form: {
          date: {
            id: "dateField",
            name: "date",
            text: "Дата",
            data: ""
          },
          name: {
            id: "nameField",
            name: "name",
            text: "ФИО",
```

```

    data: ""
  },

  sex:
  {
    name: "sex",
    data: ""
  },

  age:
  {
    id: "ageField",
    name: "age",
    text: "Возраст",
    data: 18
  },

  telephoneNumber:
  {
    id: "telNumField",
    name: "telnum",
    text: "Тел.",
    data: ""
  },

  email:
  {
    id: "emailField",
    name: "email",
    text: "Email",
    data: ""
  },
},

sexOptions: Object.freeze({
  m: {
    id: "maleSexField",
    text: "Мужской",
    value: "m"
  },
  f: {
    id: "femaleSexField",
    text: "Женский",
    value: "f"
  },
  other: {
    id: "otherSexField",
    text: "Песец",
    value: "o"
  }
})
});
},

computed:
{
  errors()
  {
    const errContainer = [];

    if (!new HorriblePersonData(this.form.name.data).isFIO())
    {
      errContainer.push({ text: "Неверные ФИО." });
    }

    if (!new HorriblePersonData(this.form.age.data).isAge())
    {
      errContainer.push({ text: "Неверный возраст." });
    }

    if (!new HorriblePersonData(this.form.telephoneNumber.data).isTelephoneNumber())
    {
      errContainer.push({ text: "Неверный номер телефона." });
    }

    if (!new HorriblePersonData(this.form.email.data).isEmail())

```

```

    {
      errContainer.push({ text: "Неверный email." });
    }

    return errContainer;
  }
}
});

app.component("radio-field", {
  props: ["id", "name", "vm", "val", "text"],
  template: `
    <label :for="id">{{ text }}</label>
    <input :id="id" type="radio" :name="name" :value="val" v-model="vm" />
  `
});

app.component("error-message", {
  props: ["header", "text"],
  template: `
    <div class="message e-m">
      <div class="message-header">{{ header }}</div>
      <div class="message-body">{{ text }}</div>
    </div>
  `
});

app.mount("#contactsForm");

```

Листинг 2 — Дополнительный код для about/index.html

```

<form id="contactsForm" action="mailto:activity.alex69985@gmail.com" method="post">
<label :for="form.date.id">{{ form.date.text }}</label>
<input type="text" v-model="form.date.data" :name="form.date.name" :id="form.date.id" />
<br />
<label :for="form.name.id">{{ form.name.text }}</label>
<input type="text" v-model="form.name.data" :name="form.name.name" :id="form.name.id" />
<br />
<radio-field v-for="option in sexOptions" :key="option.id" :id="option.id" :name="form.sex.name"
:vm="form.sex.data" :val="option.value" :text="option.text"></radio-field>
<br />
<label :for="form.age.id">{{ form.age.text }}</label>
<input :id="form.age.id" type="number" v-model="form.age.data" :name="form.age.name" min="16" max="96" />
<br />
<label :for="form.telephoneNumber.id">{{ form.telephoneNumber.text }}</label>
<input :id="form.telephoneNumber.id" type="text" v-model="form.telephoneNumber.data"
:name="form.telephoneNumber.name" />
<br />
<label :for="form.email.id">{{ form.email.text }}</label>
<input :id="form.email.id" type="text" v-model="form.email.data" :name="form.email.name"
placeholder="email" />
<div v-if="errors.length > 0">
<hr />
<error-message v-for="(error, index) in errors" :key="index" header="Ошибка" :text="error.text"></error-message>
</div>
<hr />
<input type="reset" value="Очистить форму" />
<input id="submitButton" type="button" value="Отправить" />
</form>

```

Контакт

Дата

ФИО

Мужской ☐ Женский ☐ Песец ☒

Возраст

Тел.

Email

Ошибка

Неверный номер телефона.

Ошибка

Неверный email.

Рисунок 1 — Итоговый вид формы

Вывод

При выполнении данной лабораторной работы были получены навыки построения форм и использования архитектуры MVVM при создании клиентских веб-приложений с использованием VueJS.