关键特性报告

姓名:

课程:

目录

本报告完成了对 web 页面开发的代码提供注释和页面显示和控制最重要的 6 个特性。

一、CV 展示网页的六个主要设计特点及其实施

1. LocalStorage

localStorage 用于持久化的本地存储，除非手动删除数据，否则数据永远不会过期。 使用 localStorage 只是把数据存储在客户端，不会发送到服务器上，而且对于某一个域来说，localStorage 是共享的（多个窗口共享一个“数据库”）。

代码如下：

```
window.addEventListener("storage", function(e) {
    if (e.key == "time" && e.newValue) {
        var dataHtml = "",
            data = "";
        for (var i = localStorage.length - 1; i >= 0; i--) //效率更高的循环方法
        {
            data = localStorage.getItem(localStorage.key(i)).split("|");
            //dataHtml += "<p><span class=\"msg\">" + data[0] + "</span><span
class=\"datetime\">" + data[1] + "</span><span>" + data[2]+"</span></p>";
            dataHtml += "<span style=> 公司名称： " + data[1] + "<span class=\"thetime\"> 访问
时间： " + data[2] + "</span><p><span class=\"phone\">联系方式：" + data[0] + "</span></p>";
        }
        document.getElementById("comment").innerHTML = dataHtml;
        // document.getElementById("time").innerHTML = e.newValue;
```

```
        // 调用 removeItem 方法再次 set 的时候,保证每次 set 事件都可以监听到,同时可以释放资源
        localStorage.removeItem("time");
    }});
```

2. 正则表达式

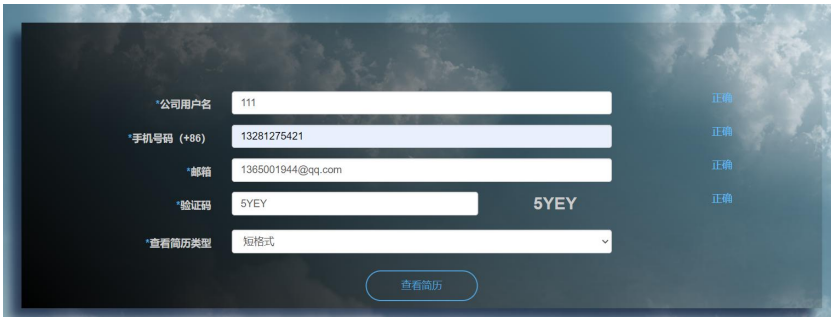
运行了正则表达式进行了简单用户的验证，对邮箱、电话号码进行了校验，并进行了验证码来避免机器输入。

代码如下：

```
function validate() { //判断验证码是否正确
    var validate = document.getElementById("validate");
    var valimg = document.getElementById("valimg");
    validate.onblur = function() {
        if (validate.value == "") {
            Cvalidate.innerHTML = "验证码不能为空";
            Cvalidate.style.color = "red";
            Cvalidate.style.fontSize = "15px"
        }
    }
    validate.onchange = function() {
        var va_text = valimg.firstChild.nodeValue;
        if (validate.value == va_text) {
            Cvalidate.innerHTML = "正确";
            Cvalidate.style.color = "#61BFFF";
```

```
            Cvalidate.style.fontSize = "15px"
        } else {
            Cvalidate.innerHTML = "验证码错误";
            Cvalidate.style.color = "red";
            Cvalidate.style.fontSize = "15px"
        }
    }}
```

运行如下：

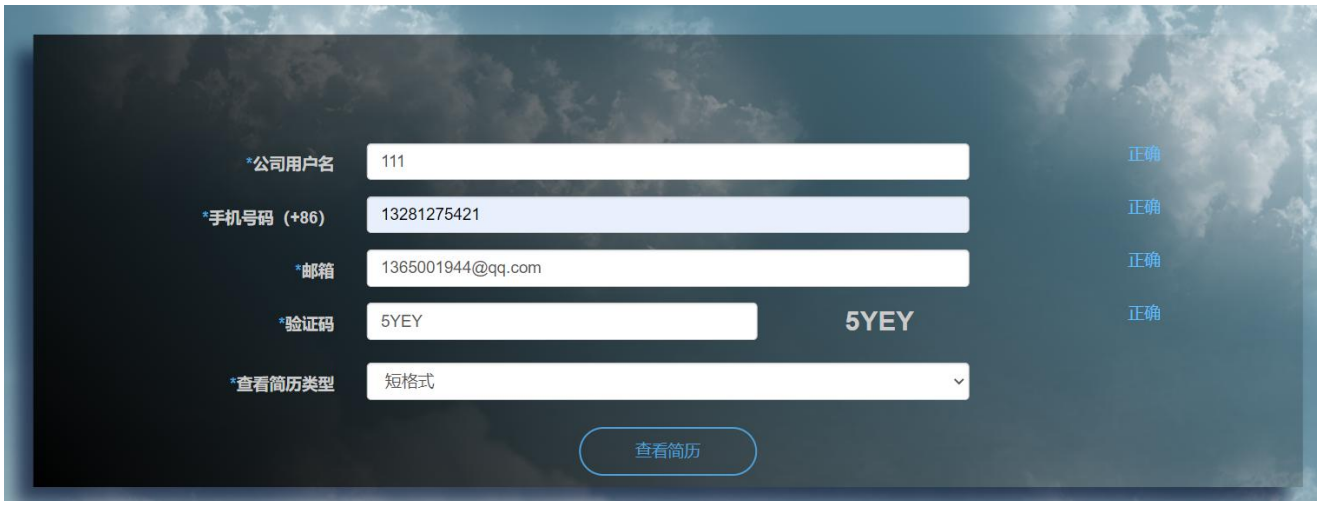


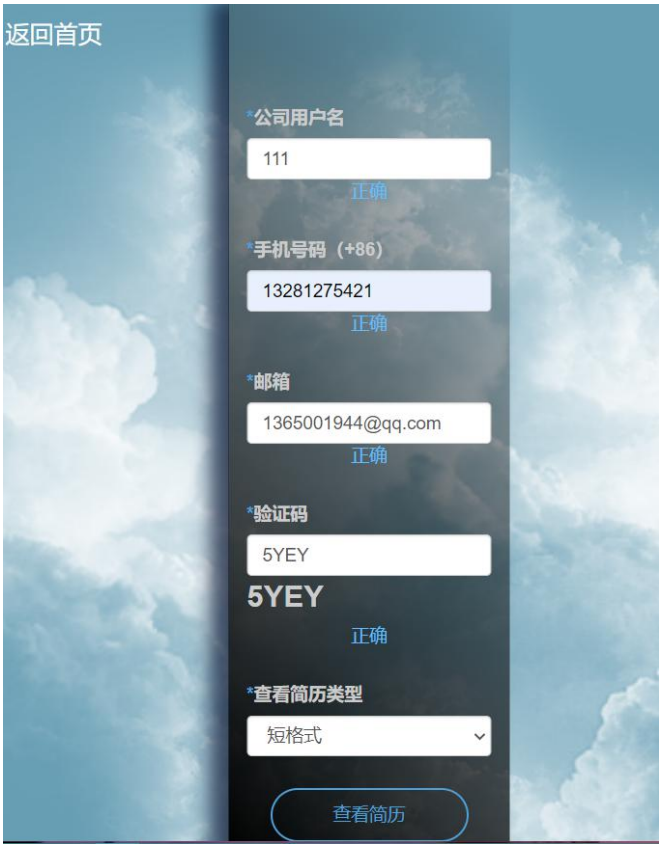
如果在浏览器中运行，并且输入错误的信息，会看到如下：

3. 响应式

根据用户使用的设备，进行相应的调整，相同的网站和页面在手机、平板电脑、台式机、手提电脑上可以进行自动适配，根据不同的屏幕尺寸，显示出不同的界面。

运行如下：





4. 兼容性

根据用户打开的浏览器不同，进行相应的调整。可以支持 360、FireFox、Chrome、Safari、Opera、傲游、搜狗、世界之窗等浏览器. 对 IE8、IE9 进行了兼容。

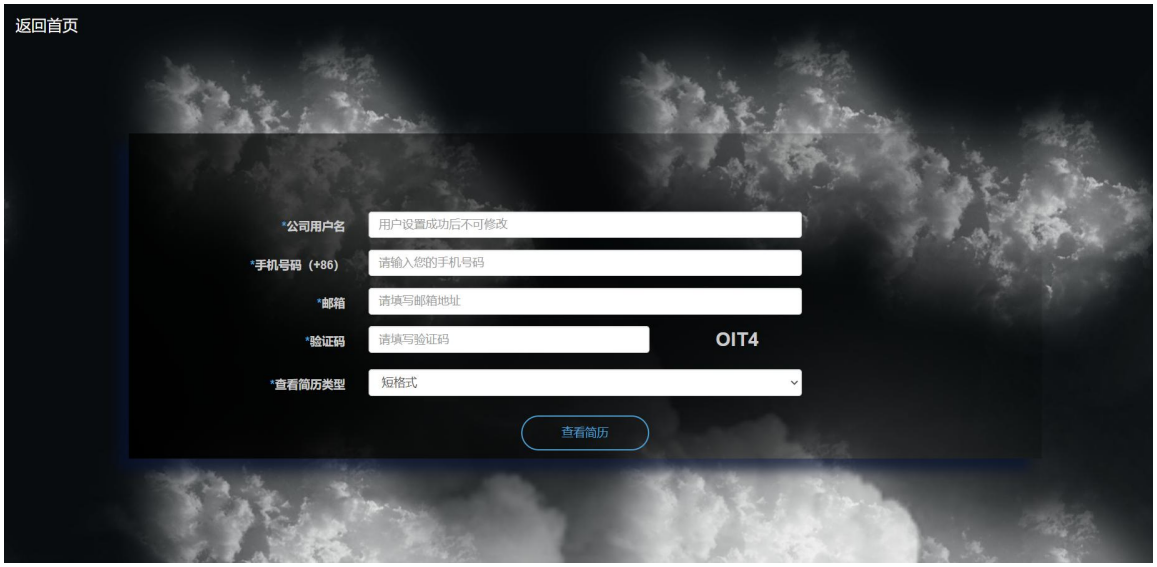
代码如下：

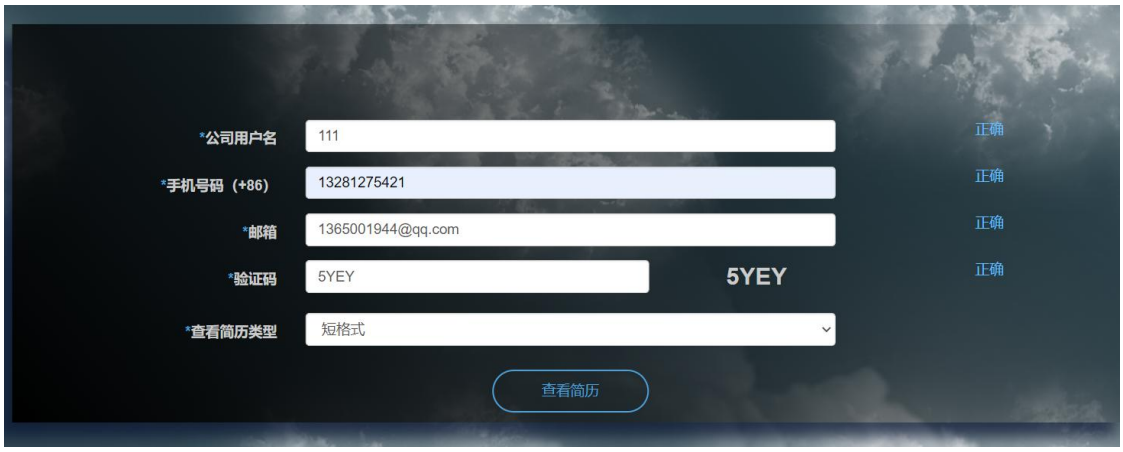
```
<!--[if lte IE 8]><script src="assets/JS/ie/html5shiv.js"></script><![endif]-->
<link rel="stylesheet" href="assets/CSS/image_map.css" />
<link rel="stylesheet" href="assets/CSS/font-awesome.min.css">
<!--[if lte IE 9]><link rel="stylesheet" href="assets/CSS/ie9.css" /><![endif]-->
<!--[if lte IE 8]><link rel="stylesheet" href="assets/CSS/ie8.css" /><![endif]-->
```

5. 用户体验好

界面采用 caverns 进行动画设计，对用户来说用来良好的视觉效果，对于错误提示等都有良好的反馈信息，用户体验极好。

运行如下：





6. 高质量

公司提交的数据可以永久保存（除非手动删除），各个页面跳转流畅，基本

实现了全部功能。

二、参考文献

IEEE 格式

三、附录

1. HTML

```
<form action="" method="" class="form-horizontal" role="form">
    <!-- 公司用户名 -->
    <div class="form-group">
        <label for="username" class="col-sm-3 control-label"> <span>*</span>公司用户名</label>
        <div class="col-sm-6">
            <input type="text" class="form-control" id="username" placeholder="用户设置成功后不可修改">
        </div>
        <div class=" col-sm-3">
            <span><p id="un"></p></span>
        </div>
    </div>
    <!-- 电话号码 -->
    <div class="form-group ">
        <label for="phone1 " class="col-sm-3 control-label "> <span>*</span>手机号码（+86）</label>
        <div class="col-sm-6 ">
            <input type="text" class="form-control " id="phone" placeholder="请输入您的手机号码 ">
```

```
                </div>
                <div class=" col-sm-3">
                    <span><p id="ph"></p></span>
                </div>
            </div>
            <!-- 邮箱 -->
            <div class="form-group ">
                <label for="email " class="col-sm-3 control-label "> <span>*</span>邮箱
</label>
                <div class="col-sm-6 ">
                    <input type="text" class="form-control " id="mail" placeholder="请填写邮箱地
址 ">
                </div>
                <div class=" col-sm-3">
                    <span><p id="ma"> </p></span>
                </div>
            </div>
            <!-- 验证码 -->
            <div class="form-group ">
                <label for="validate" class="col-sm-3 control-label "> <span>*</span>验证码
</label>
                <div class="col-sm-4">
```

```
                    <input type="text" class="form-control " id="validate" placeholder="请填写验
证码 ">
                </div>
                <label id="valimg" class="col-sm-2"> </label>
                <div class="col-sm-3">
                    <span><p id="vd"> </p></span>
                </div>
            </div>
            <!-- 简历类型 -->
            <div class="form-group ">
                <label for="trander" class="col-sm-3 control-label"><span>*</span>查看简历类
型</label>
                <div class="col-sm-6">
                    <select class="form-control" id="resume">
                <option >短格式</option>
                <option >长格式</option>
              </select>
                </div>
                <div class=" col-sm-3">

                </div>
            </div>
```

```
            <div class='login_fields__submit'>
                <input id="submit" type='button' value='查看简历'>
            </div>
        </form>
```

2. CSS

```
@keyframes one {
    from {
        margin-left: 0;
    }
    to {
        margin-left: -100%;
    }}
@keyframes two {
    from {
        margin-left: 0;
    }
    to {
        margin-left: -200%;
    }}
@keyframes three {
    from {
```

```
        margin-left: 0;
    }
    to {
        margin-left: -300%;
    }}
```

3. JavaScript

```
const transforms = {
                    x(m, v) {
                            m[12] += m[0] * v;
                            m[13] += m[1] * v;
                            m[14] += m[2] * v;
                            m[15] += m[3] * v;
                    },
                    y(m, v) {
                            m[12] += m[4] * v;
                            m[13] += m[5] * v;
                            m[14] += m[6] * v;
                            m[15] += m[7] * v;
                    },
                    z(m, v) {
                            m[12] += m[8] * v;
```

```
            m[13] += m[9] * v;
            m[14] += m[10] * v;
            m[15] += m[11] * v;
        },
        s(m, v) {
            const a = Array.isArray(v);
            const x = a ? v[0] : v;
            const y = a ? v[1] : x;
            const z = a ? v[2] : x;
            m[0] *= x;
            m[1] *= x;
            m[2] *= x;
            m[3] *= x;
            m[4] *= y;
            m[5] *= y;
            m[6] *= y;
            m[7] *= y;
            m[8] *= z;
            m[9] *= z;
            m[10] *= z;
            m[11] *= z;
        },
```

```
rx(m, v) {
        const rad = Math.PI * (v / 180);
        const s = Math.sin(rad);
        const c = Math.cos(rad);
        const a10 = m[4];
        const a11 = m[5];
        const a12 = m[6];
        const a13 = m[7];
        const a20 = m[8];
        const a21 = m[9];
        const a22 = m[10];
        const a23 = m[11];
        m[4] = a10 * c + a20 * s;
        m[5] = a11 * c + a21 * s;
        m[6] = a12 * c + a22 * s;
        m[7] = a13 * c + a23 * s;
        m[8] = a10 * -s + a20 * c;
        m[9] = a11 * -s + a21 * c;
        m[10] = a12 * -s + a22 * c;
        m[11] = a13 * -s + a23 * c;
},
ry(m, v) {
```

```
            const rad = Math.PI * (v / 180);
            const s = Math.sin(rad);
            const c = Math.cos(rad);
            const a00 = m[0];
            const a01 = m[1];
            const a02 = m[2];
            const a03 = m[3];
            const a20 = m[8];
            const a21 = m[9];
            const a22 = m[10];
            const a23 = m[11];
            m[0] = a00 * c + a20 * -s;
            m[1] = a01 * c + a21 * -s;
            m[2] = a02 * c + a22 * -s;
            m[3] = a03 * c + a23 * -s;
            m[8] = a00 * s + a20 * c;
            m[9] = a01 * s + a21 * c;
            m[10] = a02 * s + a22 * c;
            m[11] = a03 * s + a23 * c;
    },
    rz(m, v) {
            const rad = Math.PI * (v / 180);
```

```
            const s = Math.sin(rad);
            const c = Math.cos(rad);
            const a00 = m[0];
            const a01 = m[1];
            const a02 = m[2];
            const a03 = m[3];
            const a10 = m[4];
            const a11 = m[5];
            const a12 = m[6];
            const a13 = m[7];
            m[0] = a00 * c + a10 * s;
            m[1] = a01 * c + a11 * s;
            m[2] = a02 * c + a12 * s;
            m[3] = a03 * c + a13 * s;
            m[4] = a00 * -s + a10 * c;
            m[5] = a01 * -s + a11 * c;
            m[6] = a02 * -s + a12 * c;
            m[7] = a03 * -s + a13 * c;
    },
    b(m, v) {
            if (v > 0) {
                m[16] += v * (1 - m[16]);
```

```
                } else {

                    m[16] += v * m[16];

                }

            }

        };
```