

重 庆 大 学

学 生 实 验 报 告

实验课程名称 Web 开发技术

开课实验室 重庆大学 DS1501

学 院 大数据与软件学院 年级 2021 专业班 软件工程 X 班

学 生 姓 名 XXX 学 号 2021XXXX

开 课 时 间 2023 至 2024 学年第 1 学期

总 成 绩	
教师签名	XXX

重庆大学大数据与软件学院制

《Web 开发技术》实验报告

开课实验室：DS1501

2023 年 9 月 12 日

学院	大数据与软件学 院	年级、专业、班	2021 级软件工 程 X 班	姓名	XXX	成绩	
课程 名称	Web 开发技术	实验项目 名 称	Web 时钟的实现	指导教师	XXX		
教 师 评 语	<div>教师签名：XXX</div> <div>年 月 日</div>						
<div>一、实验目的（软件需求文档）</div> <div>1. 实验任务： 做一个自己的时钟页面（可用 HTML5+CSS3 或者 Web UI 库/框架），可创新设计时钟界面。提交实验报告文档。</div> <div>2. 实验目的： 本实验旨在让学生通过设计和开发一个自己的时钟页面，掌握以下关键技能和知识： ①学习使用 HTML5 和 CSS3 创建网页； ②掌握如何在网页上显示实时时间； ③理解网页布局和设计的基本原则； ④了解 Web UI 库/框架的基本用法（如果选择使用的话）； ⑤培养创新思维，设计独特和吸引人的时钟界面； 通过完成这个实验，学生将能够运用 HTML5、CSS3 和 Web 开发技术，创建一个创新性的时钟页面，并通过实验报告记录他们的学习和实践经验。</div>							

二、实验原理（软件设计文档）

1. 引言

本实验旨在创建一个自定义的时钟页面，用于显示实时的时间。时钟页面可以通过使用 HTML5 和 CSS3 手动创建，也可以选择使用现有的 Web UI 库或框架。

2. 技术栈

实验可以基于以下技术栈进行设计和开发：

- ①HTML5: 用于创建页面的基本结构和内容；
- ②CSS3: 用于样式设计，包括颜色、字体、布局等；
- ③JavaScript: 用于获取并实时更新当前时间，以及实现任何交互或动画效果；
- ④Web UI 库/框架（可选）：如果选择使用，可以加速开发过程，例如使用 React、Vue、Angular 等。

3. 页面布局

时钟页面的布局设计应考虑以下要点：

- ①页面结构：创建 HTML 文档，包括时钟的容器元素和必要的文本元素（用于显示时间）；
- ②样式设计：使用 CSS3 为页面添加样式，包括背景、字体、边框、阴影等；
- ③时钟位置：选择时钟的放置位置，可以是页面的中心、角落或其他合适的位置。

4. 实时时间更新

为了实时显示当前时间，需要使用 JavaScript 编写以下功能：

- ①获取当前时间：使用 JavaScript 的 Date 对象获取当前的小时、分钟和秒数。
- ②更新显示：将获取到的时间信息更新到页面的相应元素中，以实现实时更新。

5. 响应式设计

确保时钟页面具有响应式设计，以适应不同设备和屏幕尺寸。这可以通过 CSS3 媒体查询来实现，以适应不同的屏幕宽度和高度。

三、使用仪器、材料（软硬件开发环境）

- 1. 操作系统：Windows 11
- 2. 开发设备：Lenovo Legion R9000P2021H
- 3. 开发平台：IntelliJ IDEA 2022.1.3
- 4. 测试浏览器：Firefox 浏览器

四、实验步骤（实现的过程）

1. 初始设置

在开始实验之前，确保已经准备好所需的开发工具和环境，包括文本编辑器、浏览器以及根据选择的技术栈准备好的 HTML、CSS 和 JavaScript 文件。

2. 创建 HTML 结构

Clock.html:

```
1  <!DOCTYPE html> <!-- 声明文档类型为HTML5 -->
2  <html lang="en"> <!-- 设置文档语言为英语 -->
3  <head>
4      <meta charset="UTF-8"> <!-- 设置字符编码为UTF-8，以支持特殊字符和多语言 -->
5      <title>Web Clock</title> <!-- 设置网页标题 -->
6      <link rel="stylesheet" type="text/css" href="ClockStyle.css"> <!-- 导入外部CSS文件，用于样式设置 -->
7  </head>
8  <body>
9      <div class="clock"> <!-- 创建一个包含时钟的容器 -->
10         <div class="outer-clock-face"> <!-- 创建外部时钟圆盘容器 -->
11             <div class="marking marking-one"></div> <!-- 创建外部时钟圆盘的刻度1 -->
12             <div class="marking marking-two"></div> <!-- 创建外部时钟圆盘的刻度2 -->
13             <div class="marking marking-three"></div> <!-- 创建外部时钟圆盘的刻度3 -->
14             <div class="marking marking-four"></div> <!-- 创建外部时钟圆盘的刻度4 -->
15         </div>
16         <div class="inner-clock-face"> <!-- 创建内部时钟圆盘容器 -->
17             <div class="hand hour-hand"></div> <!-- 创建时钟的时针 -->
18             <div class="hand min-hand"></div> <!-- 创建时钟的分针 -->
19             <div class="hand second-hand"></div> <!-- 创建时钟的秒针 -->
20         </div>
21     </div>
22     <script type="text/javascript" src="ClockRun.js"></script> <!-- 导入外部JavaScript文件，用于时钟的运行和动态更新 -->
23 </body>
24 </html>
```

3. 添加样式

ClockStyle.css:

```
1  /* 设置整个HTML文档的背景颜色、文本居中和基本字体大小 */
2  html {
3      background: #f0f0f0; /* 背景颜色 */
4      text-align: center; /* 文本居中 */
5      font-size: 10px; /* 基本字体大小 */
6  }
7
8  /* 设置页面主体样式，包括背景渐变色和页面布局 */
9  body {
10     margin: 0; /* 清除页面默认的外边距 */
11     font-size: 2rem; /* 设置页面主体的字体大小 */
12     display: flex; /* 使用弹性布局 */
13     flex: 1; /* 设置主体部分充满可用空间 */
14     min-height: 100vh; /* 设置最小高度为视口高度，确保内容填满屏幕 */
15     align-items: center; /* 垂直居中内容 */
16     background: linear-gradient(to bottom, #40b0a3, #f0f0f0); /* 背景渐变色 */
17  }
18
19  /* 设置时钟外部容器的样式，包括大小、边框和阴影 */
20  .clock {
21     width: 30rem; /* 设置时钟宽度 */
22     height: 30rem; /* 设置时钟高度 */
23     position: relative; /* 定位上下文为相对定位 */
24     padding: 2rem; /* 设置内边距 */
25     border: 7px solid #f0f0f0; /* 设置边框 */
26     box-shadow: -4px -4px 10px rgba(67, 67, 67, 0.5), /* 添加阴影效果 */
27         inset 4px 4px 10px rgba(0, 0, 0, 0.5),
28         inset -4px -4px 10px rgba(67, 67, 67, 0.5),
29         4px 4px 10px rgba(0, 0, 0, 0.3);
30     border-radius: 50%; /* 将边框设置为圆形 */
31     margin: 50px auto; /* 水平居中 */
32     background: #ffffff; /* 背景颜色设置为白色 */
33  }
```

```

34
35  /* 设置外部时钟表盘的样式，包括大小和形状 */
36  .outer-clock-face {
37      position: relative; /* 相对定位 */
38      background: #ffffff; /* 背景颜色为白色 */
39      overflow: hidden; /* 隐藏超出部分 */
40      width: 100%; /* 宽度100% */
41      height: 100%; /* 高度100% */
42      border-radius: 100%; /* 边框设置为圆形 */
43  }
44
45  /* 创建时钟表盘的小时刻度，包括样式和位置 */
46  .outer-clock-face::after {
47      content: ''; /* 伪元素用于绘制刻度 */
48      position: absolute; /* 绝对定位 */
49      width: 5px; /* 刻度宽度 */
50      height: 100%; /* 刻度高度 */
51      background: #1df52f; /* 刻度颜色 */
52      z-index: 0; /* 层级设置为0 */
53      left: 49%; /* 水平位置居中 */
54      transform: rotate(90deg); /* 旋转90度，使刻度竖直显示 */
55  }
56
57  /* 创建时钟表盘的分钟刻度，包括样式和位置 */
58  .outer-clock-face::before,
59  .outer-clock-face .marking {
60      content: ''; /* 伪元素用于绘制刻度 */
61      position: absolute; /* 绝对定位 */
62      width: 5px; /* 刻度宽度 */
63      height: 100%; /* 刻度高度 */
64      background: #1df52f; /* 刻度颜色 */
65      z-index: 0; /* 层级设置为0 */
66      left: 49%; /* 水平位置居中 */
67  }
68
69  /* 设置分钟刻度的样式和位置 */
70  .outer-clock-face .marking {
71      background: #bdbdc9; /* 刻度颜色 */
72      width: 3px; /* 刻度宽度 */
73  }
74
75  .outer-clock-face .marking.marking-one {
76      transform: rotate(30deg); /* 旋转30度，表示第一分钟刻度 */
77  }
78
79  .outer-clock-face .marking.marking-two {
80      transform: rotate(60deg); /* 旋转60度，表示第二分钟刻度 */
81  }
82
83  .outer-clock-face .marking.marking-three {
84      transform: rotate(120deg); /* 旋转120度，表示第三分钟刻度 */
85  }
86
87  .outer-clock-face .marking.marking-four {
88      transform: rotate(150deg); /* 旋转150度，表示第四分钟刻度 */
89  }

```

```

90
91  /* 设置时钟内部表盘的样式，包括大小和形状 */
92  .inner-clock-face {
93      position: absolute; /* 绝对定位 */
94      top: 10%; /* 顶部距离父容器的百分比 */
95      left: 10%; /* 左侧距离父容器的百分比 */
96      width: 80%; /* 宽度80% */
97      height: 80%; /* 高度80% */
98      background: #ffffff; /* 背景颜色为白色 */
99      border-radius: 100%; /* 边框设置为圆形 */
100  }

```

```

101
102  /* 设置时钟中心点的样式和位置 */
103  .inner-clock-face::before {
104      content: ''; /* 伪元素用于绘制时钟中心点 */
105      position: absolute; /* 绝对定位 */
106      top: 50%; /* 顶部位置居中 */
107      left: 50%; /* 左侧位置居中 */
108      width: 16px; /* 宽度 */
109      height: 16px; /* 高度 */
110      border-radius: 18px; /* 边框设置为圆形 */
111      margin-left: -9px; /* 水平居中 */
112      margin-top: -6px; /* 垂直居中 */
113      background: #4d4b63; /* 背景颜色 */
114      z-index: 11; /* 层级设置为11，位于时钟指针之上 */
115  }
116
117  /* 设置时钟指针的样式和位置 */
118  .hand {
119      width: 50%; /* 宽度50% */
120      right: 50%; /* 右侧位置居中 */
121      height: 6px; /* 高度 */
122      background: #61afff; /* 指针颜色 */
123      position: absolute; /* 绝对定位 */
124      top: 50%; /* 顶部位置居中 */
125      border-radius: 6px; /* 边框设置为圆形 */
126      transform-origin: 100%; /* 旋转中心点在右侧 */
127      transform: rotate(90deg); /* 初始旋转角度为90度 */
128      transition-timing-function: cubic-bezier(0.1, 2.7, 0.58, 1); /* 过渡动画函数 */
129  }
130
131  /* 设置时钟的时针样式和层级 */
132  .hand.hour-hand {
133      width: 30%; /* 时针宽度为整个指针的30% */
134      z-index: 3; /* 层级设置为3，位于分钟和秒针之上 */
135  }
136
137  /* 设置时钟的分针样式和层级 */
138  .hand.min-hand {
139      height: 3px; /* 分针高度为3px */
140      z-index: 10; /* 层级设置为10，位于秒针之上 */
141      width: 40%; /* 分针宽度为整个指针的40% */
142  }
143
144  /* 设置时钟的秒针样式 */
145  .hand.second-hand {
146      background: #ee791a; /* 秒针颜色为橙色 */
147      width: 45%; /* 秒针宽度为整个指针的45% */
148      height: 2px; /* 秒针高度为2px */
149  }

```

4. JavaScript编程

ClockRun.js:

```
1 document.addEventListener( type: "DOMContentLoaded", listener: function () {
2     // 选择HTML文档中具有以下类名的元素，分别代表时钟的秒、分和时针
3     const secondHand = document.querySelector( selectors: '.second-hand' );
4     const minsHand = document.querySelector( selectors: '.min-hand' );
5     const hourHand = document.querySelector( selectors: '.hour-hand' );
6
7     // 定义一个用于更新时钟指针位置的函数
8     function setDate() {
9         // 创建一个表示当前时间的Date对象
10        const now = new Date();
11
12        // 获取当前秒数
13        const seconds = now.getSeconds();
14        // 计算秒针的旋转角度，每秒钟旋转6度，加上90度是为了让指针从12点钟方向开始
15        const secondsDegrees = ((seconds / 60) * 360) + 90;
16        // 将计算出的角度应用到秒针的CSS样式中，使其旋转到正确的位置
17        secondHand.style.transform = `rotate(${secondsDegrees}deg)`;
18
19        // 获取当前分钟数
20        const mins = now.getMinutes();
21        // 计算分针的旋转角度，每分钟旋转6度，再根据秒数调整位置，加上90度是为了让指针从12点钟方向开始
22        const minsDegrees = ((mins / 60) * 360) + ((seconds / 60) * 6) + 90;
23        // 将计算出的角度应用到分针的CSS样式中，使其旋转到正确的位置
24        minsHand.style.transform = `rotate(${minsDegrees}deg)`;
25
26        // 获取当前小时数
27        const hour = now.getHours();
28        // 计算时针的旋转角度，每小时旋转30度，再根据分钟数调整位置，加上90度是为了让指针从12点钟方向开始
29        const hourDegrees = ((hour / 12) * 360) + ((mins / 60) * 30) + 90;
30        // 将计算出的角度应用到时针的CSS样式中，使其旋转到正确的位置
31        hourHand.style.transform = `rotate(${hourDegrees}deg)`;
32    }
33
34    // 每隔1秒钟调用一次setDate函数，以更新时钟指针的位置，实现动画效果
35    setInterval(setDate, timeout: 1000);
36
37    // 初始化页面加载时钟的位置，避免刷新页面时出现指针跳动
38    setDate();
39 } });
```

5. 总体结构

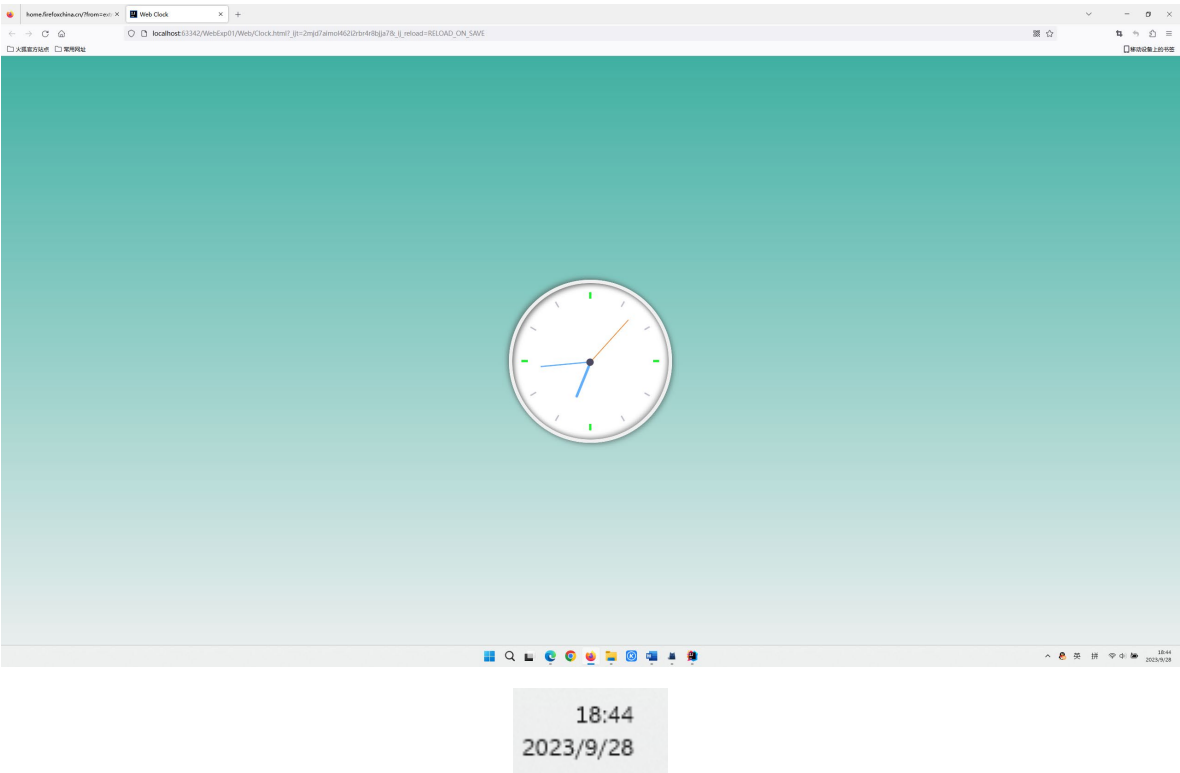


6. 测试和调试

- ①在Firefox浏览器中打开HTML文件，检查时钟页面是否正确显示当前时间；
- ②进行测试，确保时钟在不同屏幕尺寸和设备上具有良好的响应性。

五、实验结果及分析（实现的效果，包括屏幕截图、系统总体运行情况和测试情况等）

1. 实现效果：



程序正常运行，在一台 24 寸 240hz 浏览器和一台 27 寸 170hz 浏览器上均正常显示，且始终的结果与系统时间相符合。

2.实验分析：

（1）时钟页面设计

时钟页面的设计采用了简洁、现代和吸引人的外观。以下是一些主要设计元素：

- ①背景和颜色方案：页面背景采用浅色调，突出显示时钟内容。时针分针秒针颜色分别为深蓝色、蓝色、橙色，增强了可读性；
- ②布局：时钟页面的布局经过精心设计，使其在不同屏幕尺寸上都能够良好地呈现；
- ③动画效果：添加了平滑的秒针动画，使时钟更生动。

（2）实时时间显示

时钟页面成功地实现了实时时间的显示功能。当前的小时、分钟和秒数以钟表形式显示在页面上，并随时间的推移而更新。

（3）响应式设计

时钟页面经过响应式设计，能够适应不同设备和屏幕尺寸，在不同的显示器上均能正常显示，确保了用户体验的一致性。

（4）总结和展望

通过这个实验，成功地创建了一个自定义的时钟页面，实现了实时时间的显示，并添加了创新设计元素。学到了如何使用 HTML5、CSS3 和 JavaScript 来创建动态网页，并如何实现响应式设计。这些技能对于作为 Web 开发技术初学者的我来说是非常有价值的。

未来，可以进一步改进时钟页面，添加更多的创新设计元素，或将其作为一个更大项目的一部分，以进一步提高用户体验。此外，还可以考虑将该页面与其他功能集成，例如提醒功能或时区转换。

实验报告打印格式说明

1. 标题：三号加粗黑体
2. 开课实验室：5 号加粗宋体
3. 表中内容：
 - (1) 标题：5 号黑体
 - (2) 正文：5 号宋体
4. 纸张：16 开 (20cm×26.5cm)
5. 版芯
 - 上距：2cm
 - 下距：2cm
 - 左距：2.8cm
 - 右距：2.8cm

说明：1、“年级专业班”可填写为“00 电子 1 班”，表示 2000 级电子工程专业第 1 班。

2、实验成绩可按五级记分制（即优、良、中、及格、不及格），或者百分制记载，若需要将实验成绩加入对应课程总成绩的，则五级记分应转换为百分制。