

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 10 月 29 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	利用 ajax 实现省市联动	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：<div>每个省份下面有很多城市，当用户选择不同的省份后，相应城市列表的名称利用 Ajax 方法马上从数据库中取出，并迅速地进行异步更新。</div></div> <div>2. 实验目的：<div>本实验旨在让学生通过设计和开发一个自己的省市联动页面，实现动态的、实时的省市联动，让用户能够方便地根据所选省份获取相应城市的信息，提高用户体验和数据准确性，在此过程中学生需要掌握以下关键技能和知识：<div>①创建一个动态的省市联动功能，通过网页提供省份的选择框和城市的选择框；</div><div>②当用户选择不同的省份时，通过 Ajax 方法发起异步请求，将用户选择的省份信息传递给后端服务器；</div><div>③后端服务器接收用户选择的省份信息，并查询数据库，以获取该省份下的城市列表数据；</div><div>④后端服务器将查询到的城市列表数据以 JSON 格式返回给前端；</div><div>⑤前端使用 Ajax 方法接收后端返回的城市数据，并实时更新城市的选择框，以显示与所选省份相关的城市选项。</div><div>这个实验有助于学生学习如何使用 Ajax 技术，从数据库中获取数据并将其动态展示在网页上，从而提升前端开发和数据交互的技能。</div></div></div>							

二、实验原理（软件设计文档）

1. 引言

本实验旨在开发一个省市区三级联动页面，实现动态的、实时的省市区联动，让用户能够方便地根据所选省份获取相应城市的信息，提高用户体验和数据准确性。

2. 技术栈

实验可以基于以下技术栈进行设计和开发：

①HTML 和 CSS：基本的网页标记语言和样式表用于构建和样式化用户界面，包括省份和城市选择框的布局和样式；

②JavaScript：JavaScript 编程语言用于实现前端逻辑，包括处理用户选择省份后的异步数据获取和更新城市列表；

③Ajax（Asynchronous JavaScript and XML）：Ajax 是一种前端技术，用于实现异步数据交换。在这个实验中，Ajax 方法被用来异步获取城市数据，而不需要重新加载整个页面；

④JSON（JavaScript Object Notation）：JSON 是一种数据交换格式，通常用于在前端和后端之间传递数据。在这个实验中，城市数据通常以 JSON 格式返回给前端，然后解析并更新城市选择框；

⑤后端开发语言和数据库：实验中需要一个后端服务器，通常使用后端开发语言（如 PHP、Python、Node.js 等）来处理 Ajax 请求并从数据库中检索城市数据。数据库用于存储省份和城市的信息；

⑥网络请求和响应：实验中需要前端与后端之间的网络通信。前端通过 Ajax 请求将用户选择的省份发送到后端，后端处理请求并返回城市数据的响应；

⑦前端框架和库：在实际应用中，可以使用前端框架或库来简化和加速开发。一些流行的选择包括 jQuery、Vue.js、React 等，它们提供了更高级的工具和模板来管理 Ajax 请求和动态 UI 更新。

3. 页面布局

①页面的背景颜色为淡灰色；

②页面使用弹性布局，使内容垂直居中；

③标题段落具有大字号、加粗字体、底部外边距和深灰色文本颜色；

④下拉选择框具有一定宽度、内外边距、边框、圆角、外边距、字体大小和白色背景颜色；

⑤选择框选项的字体大小设置为 16 像素；

⑥特定 ID 选择框#districtSelect 具有底部外边距；

⑦脚本标签具有顶部外边距；

⑧定制滚动条样式（仅适用于 webkit 浏览器），设置滚动条宽度和滑块样式。

4. 省市区三级联动选择框

为实现省市区三级联动选择框，以下功能被实现：

- ①初始化选择框：页面加载后，通过 `setupSelects()`函数初始化省、市、区三个下拉选择框，以及它们之间的关联；
- ②填充省份数据：将省份数据填充到省份选择框(`provinceSelect`)中；
- ③城市选择：当用户选择省份后，城市选择框(`citySelect`)根据所选省份的编码填充对应的城市数据，并清空区选择框；
- ④区选择：当用户选择城市后，区选择框(`districtSelect`)根据所选城市的编码填充对应的区数据；
- ⑤三级联动：用户可以从省、市、区三个选择框中选择地区，实现省市区三级联动选择功能。

5. 响应式设计

为了确保省市区三级联动页面具有响应式设计，能够适应不同设备和屏幕尺寸，可以采用以下方法：

- ① 使用媒体查询：在 CSS 样式表中使用媒体查询（Media Queries），根据不同的屏幕宽度和高度应用不同的样式规则。例如，定义不同的布局、字体大小、间距等属性，以适应小屏幕设备和大屏幕设备；
- ② 弹性布局：使用弹性盒子（Flexbox）和网格布局（Grid Layout）等 CSS 布局技术，以确保页面元素能够自适应屏幕尺寸的变化，实现灵活的排列和布局；
- ④ 移动优先设计：采用移动优先设计原则，首先考虑小屏幕设备，然后逐渐增加样式和功能以适应大屏幕设备，以提供更好的用户体验；
- ⑤ 测试和调整：在不同设备和浏览器中进行测试，并根据用户反馈和性能优化结果，对响应式设计进行不断调整和改进，以确保页面在各种情况下都能良好运行。

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

Linkage.html:

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <!-- 设置文档的兼容性声明，以支持IE浏览器 -->
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <!-- 设置视口的宽度和初始缩放比例 -->
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <!-- 引入外部CSS样式表文件，LinkageCSS.css -->
9      <link rel="stylesheet" type="text/css" href="LinkageCSS.css">
10     <!-- 设置页面标题 -->
11     <title>Web-Linkage</title>
12     <!-- 引入外部JavaScript文件，PandC-Information.js，包含地区信息数据 -->
13     <script src="PandC-Information.js"></script>
14     <!-- 引入外部JavaScript文件，LinkageRun.js，包含省市区联动的逻辑代码 -->
15     <script src="LinkageRun.js"></script>
16 </head>
17 <body onload="setupSelects()">
18     <!-- 页面主体部分开始，当页面加载完成后会执行setupSelects()函数 -->
19     <p>全国省市区三级联动</p>
20     <!-- 三个下拉选择框，用于选择省、市、区 -->
21     <select name="" id="provinceSelect">
22         <!-- 默认选项，提示用户进行选择 -->
23         <option value="">---请选择---</option>
24     </select>
25     <select name="" id="citySelect">
26         <!-- 默认选项，提示用户进行选择 -->
27         <option value="">---请选择---</option>
28     </select>
29     <select name="" id="districtSelect">
30         <!-- 默认选项，提示用户进行选择 -->
31         <option value="">---请选择---</option>
32     </select>
33 </body>
34 </html>
```

3.添加样式

LinkageCSS.css:

```
1      /* 通用重置：将所有元素的内外边距和盒模型设为边框模型 */
2      * {
3          margin: 0;
4          padding: 0;
5          box-sizing: border-box;
6      }
7
8      /* 设置整个页面的样式 */
9      body {
10         /* 使用Arial字体族，如果不可用，则使用系统默认无衬线字体 */
11         font-family: Arial, sans-serif;
12         /* 设置页面背景颜色 */
13         background-color: #f5f5f5;
14         /* 使用弹性布局，垂直居中显示内容 */
15         display: flex;
16         flex-direction: column;
17         align-items: center;
18         justify-content: center;
19         /* 设置页面高度为视口高度的100% */
20         height: 100vh;
21     }
22
23     /* 标题段落的样式 */
24     p {
25         /* 设置字体大小为24像素，加粗字体 */
26         font-size: 24px;
27         font-weight: bold;
28         /* 底部外边距为20像素 */
29         margin-bottom: 20px;
30         /* 文本颜色为深灰色 */
31         color: #333;
32     }
```

```
34  /* 下拉选择框的样式 */
35  select {
36      /* 设置宽度为250像素 */
37      width: 250px;
38      /* 设置内边距为10像素 */
39      padding: 10px;
40      /* 设置边框为1像素实线，颜色为灰色 */
41      border: 1px solid #ccc;
42      /* 设置边框的圆角为5像素 */
43      border-radius: 5px;
44      /* 设置外边距为10像素 */
45      margin: 10px;
46      /* 设置字体大小为16像素 */
47      font-size: 16px;
48      /* 设置背景颜色为白色 */
49      background-color: #fff;
50  }
51
52  /* 选择框选项的样式 */
53  option {
54      /* 设置字体大小为16像素 */
55      font-size: 16px;
56  }
57
58  /* 特定ID选择框的样式 */
59  #districtSelect {
60      /* 底部外边距为20像素 */
61      margin-bottom: 20px;
62  }
63
64  /* 脚本标签的样式 */
65  script {
66      /* 顶部外边距为20像素 */
67      margin-top: 20px;
68  }
69
70  /* 定制滚动条样式（仅适用于webkit浏览器） */
71  select::-webkit-scrollbar {
72      /* 设置滚动条宽度为10像素 */
73      width: 10px;
74  }
```

```
75
76     /* 定制滚动条滑块样式（仅适用于webkit浏览器） */
77     select::-webkit-scrollbar-thumb {
78         /* 设置滑块背景颜色为灰色 */
79         background-color: #ccc;
80         /* 设置滑块的圆角为5像素 */
81         border-radius: 5px;
82     }
```

4.JavaScript编程

LinkageRun. js:


```

43
44 // 创建文档片段，用于临时存储城市选项
45 var cityOptionFragment = document.createDocumentFragment();
46
47 // 遍历城市数据，创建城市选项元素并添加到文档片段中
48 for (var j = 0; j < citiesArray.length; j++) {
49     var cityOptionElement = document.createElement( tagName: "option");
50     cityOptionElement.innerHTML = citiesArray[j].name;
51     cityOptionElement.value = citiesArray[j].code;
52     cityOptionFragment.appendChild(cityOptionElement);
53 }
54 // 将城市选项元素添加到城市选择框
55 citySelect.appendChild(cityOptionFragment);
56 }
57
58 // 定义一个数组，用于存储区数据
59 var districtsArray = [];
60
61 // 当城市选择框内容发生改变时，执行以下代码
62 citySelect.onchange = function () {
63     // 清空区选择框的内容
64     districtSelect.innerHTML = "<option value=\"\">---请选择---</option>";
65
66     // 遍历城市数据，查找与所选城市编码匹配的区数据
67     for (var i = 0; i < citiesArray.length; i++) {
68         if (citiesArray[i].code == this.value) {
69             districtsArray = citiesArray[i].children;
70             break;
71         }
72     }
73 }

```

```

74 // 创建文档片段，用于临时存储区选项
75 var districtOptionFragment = document.createDocumentFragment();
76
77 // 遍历区数据，创建区选项元素并添加到文档片段中
78 for (var j = 0; j < districtsArray.length; j++) {
79     var districtOptionElement = document.createElement( tagName: "option");
80     districtOptionElement.innerHTML = districtsArray[j].name;
81     districtOptionElement.value = districtsArray[j].code;
82     districtOptionFragment.appendChild(districtOptionElement);
83 }
84 // 将区选项元素添加到区选择框
85 districtSelect.appendChild(districtOptionFragment);
86 }
87 }

```

5. 省市区信息存储

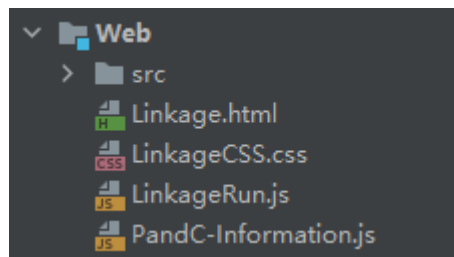
PandC-Information.js (因篇幅过多, 此处仅展示一部分, 详情请见附件)

```
1  var data = [  
2  {  
3      "code" : "110000",  
4      "name" : "北京市",  
5      "children" : [{  
6          "code" : "110100",  
7          "name" : "市辖区",  
8          "children" : [{  
9              "code" : "110101",  
10             "name" : "东城区"  
11         }, {  
12             "code" : "110102",  
13             "name" : "西城区"  
14         }, {  
15             "code" : "110105",  
16             "name" : "朝阳区"  
17         }, {  
18             "code" : "110106",  
19             "name" : "丰台区"  
20         }, {  
21             "code" : "110107",  
22             "name" : "石景山区"  
23         }, {  
24             "code" : "110108",  
25             "name" : "海淀区"  
26         }, {  
27             "code" : "110109",  
28             "name" : "门头沟区"  
29         }, {  
30             "code" : "110111",  
31             "name" : "房山区"  
32         }, {  
33             "code" : "110112",  
34             "name" : "通州区"  
35         }, {  
36             "code" : "110113",  
37             "name" : "顺义区"  
38         }, {  
39             "code" : "110114",  
40             "name" : "昌平区"  
41         }, {  
42             "code" : "110115",
```

```
43         "name" : "大兴区"
44     }, {
45         "code" : "110116",
46         "name" : "怀柔区"
47     }, {
48         "code" : "110117",
49         "name" : "平谷区"
50     }, {
51         "code" : "110118",
52         "name" : "密云区"
53     }, {
54         "code" : "110119",
55         "name" : "延庆区"
56     }
57 ]
58 }
59 ]
60 },
61 {
62     "code" : "120000",
63     "name" : "天津市",
64     "children" : [{
65         "code" : "120100",
66         "name" : "市辖区",
67         "children" : [{
68             "code" : "120101",
69             "name" : "和平区"
70         }, {
71             "code" : "120102",
72             "name" : "河东区"
73         }, {
74             "code" : "120103",
75             "name" : "河西区"
76         }, {
77             "code" : "120104",
78             "name" : "南开区"
```

```
79      }, {
80          "code" : "120105",
81          "name" : "河北区"
82      }, {
83          "code" : "120106",
84          "name" : "红桥区"
85      }, {
86          "code" : "120110",
87          "name" : "东丽区"
88      }, {
89          "code" : "120111",
90          "name" : "西青区"
91      }, {
92          "code" : "120112",
93          "name" : "津南区"
94      }, {
95          "code" : "120113",
96          "name" : "北辰区"
97      }, {
98          "code" : "120114",
99          "name" : "武清区"
100     }, {
101         "code" : "120115",
102         "name" : "宝坻区"
```

6. 总体结构



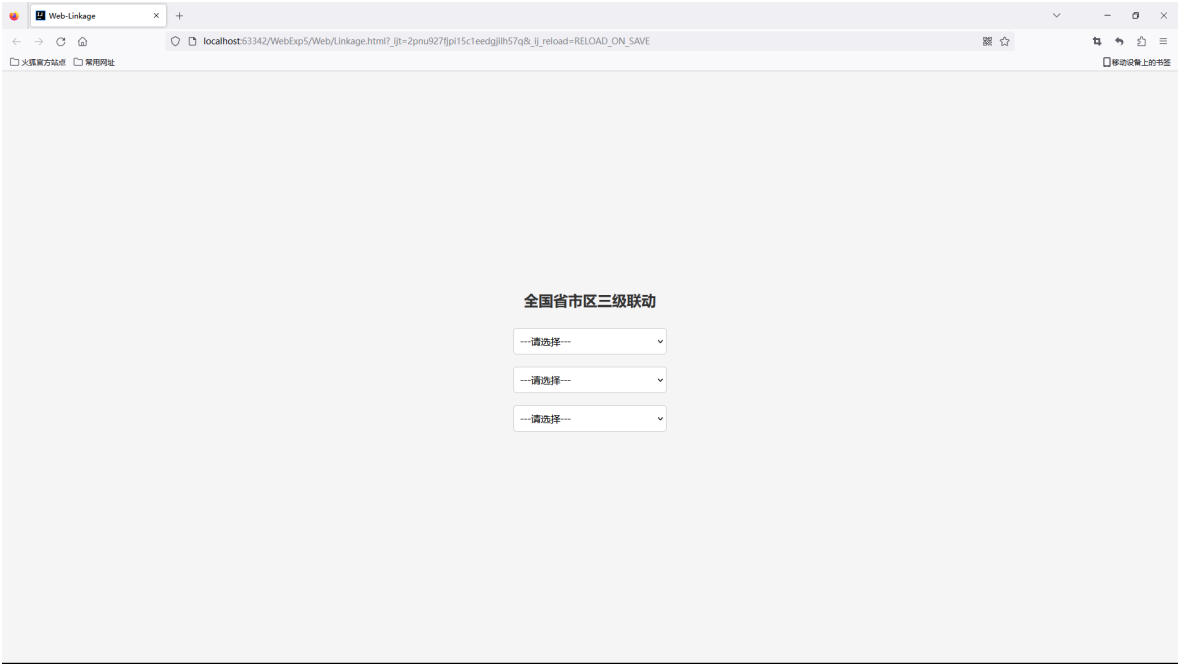
7. 测试和调试

- ①在Firefox浏览器中打开HTML文件，检查省市区三级联动页面是否正常工作；
- ②进行测试，确保省市区三级联动页面在不同屏幕尺寸和设备上具有良好的响应性。

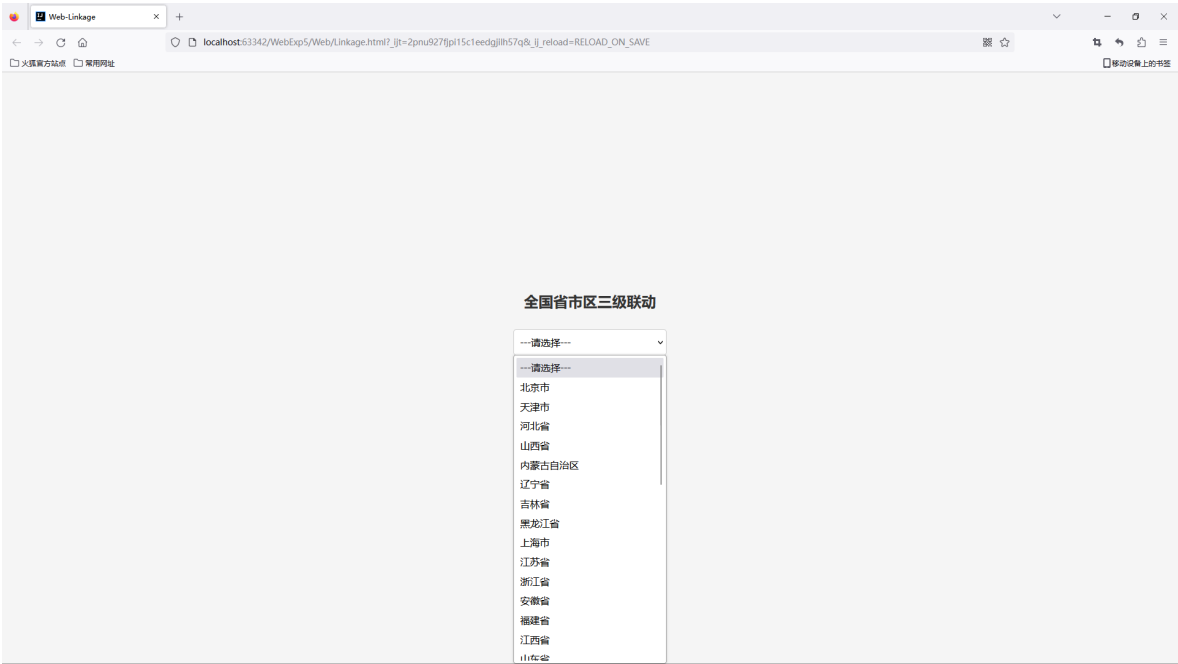
五、实验结果及分析（实现的效果，包括屏幕截图、系统总体运行情况和测试情况等）

1. 实现效果：

①主界面

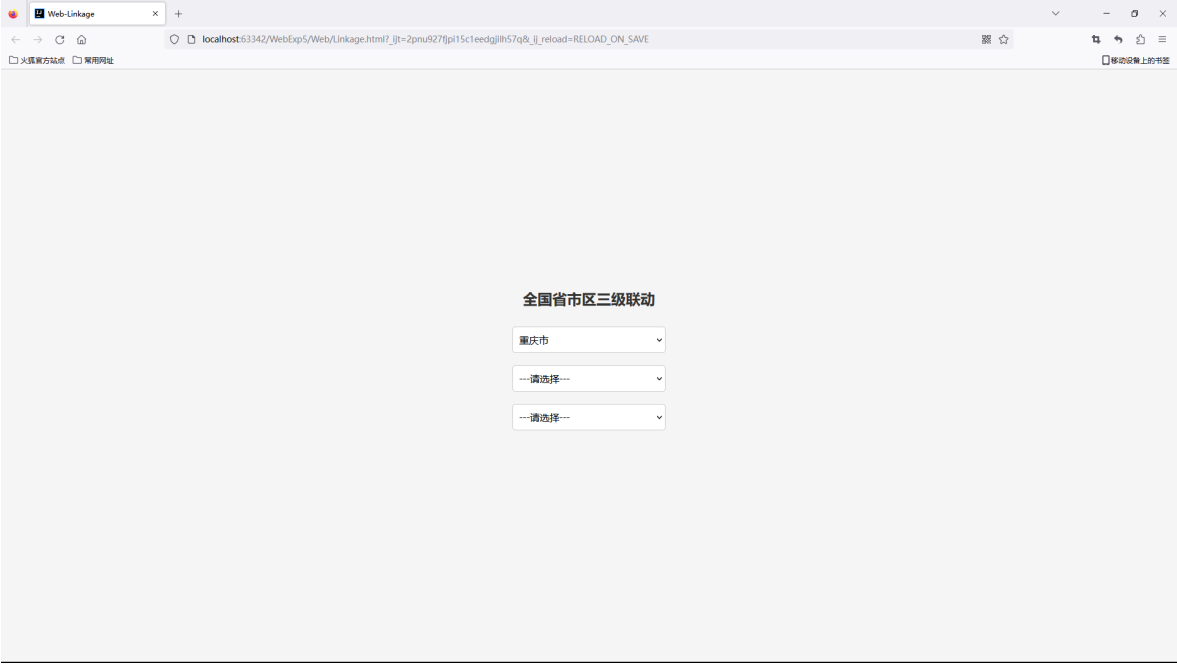
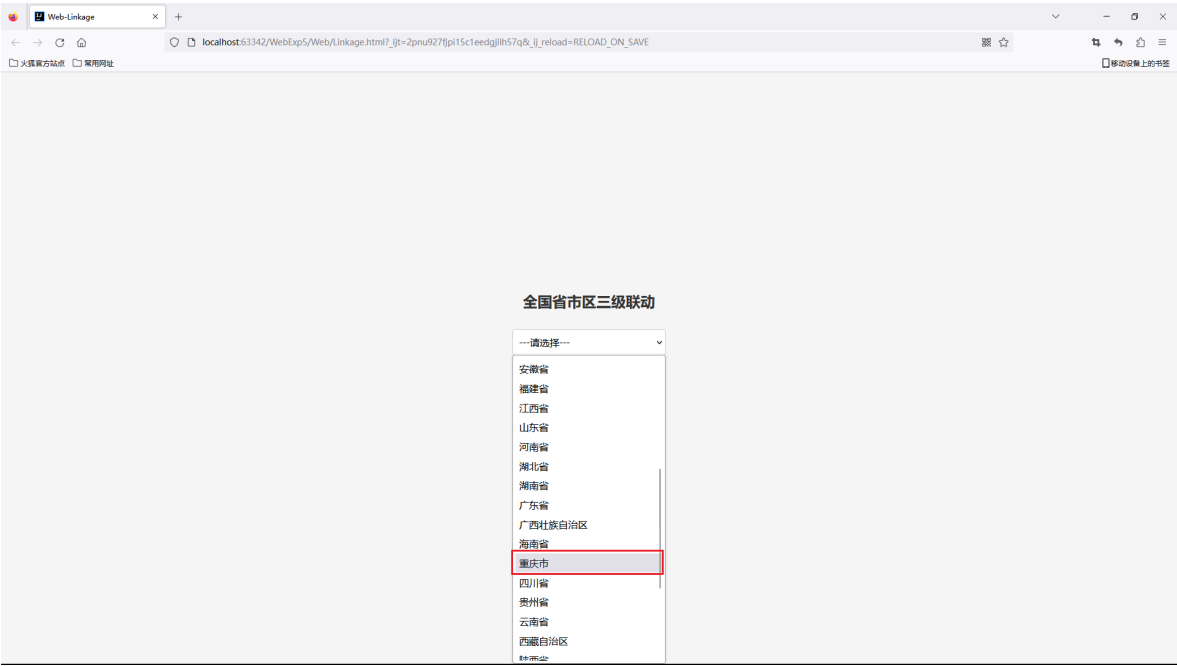


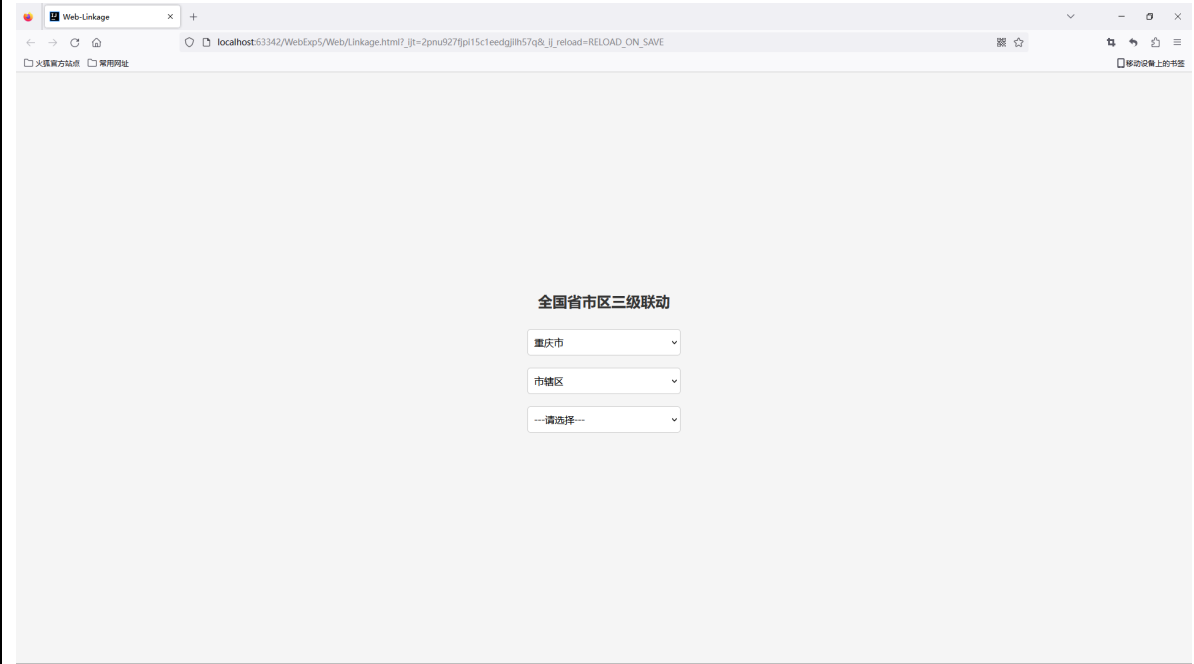
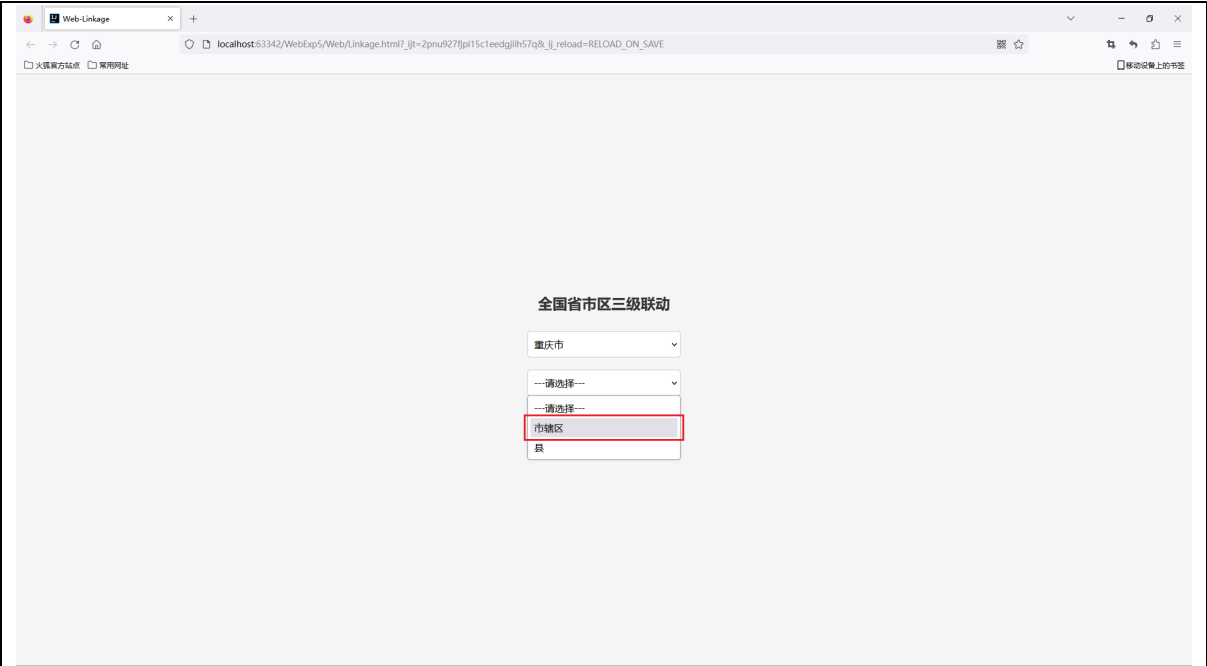
从上到下可以依次选择省级行政区域、市级行政区域、区级行政区域：

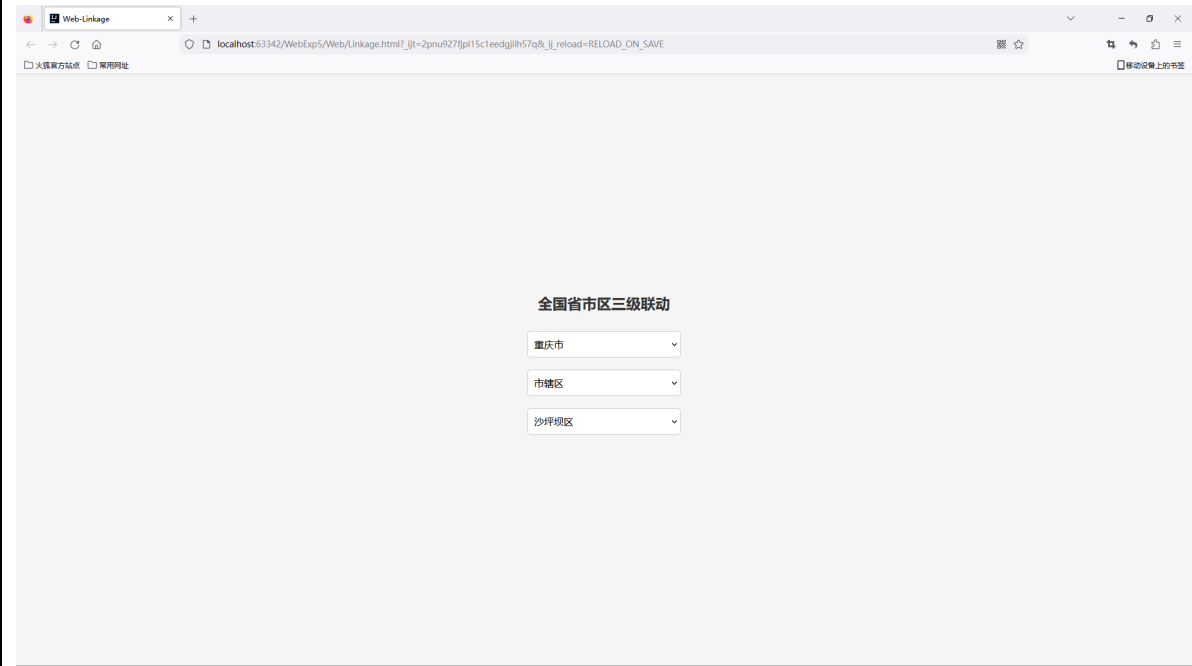
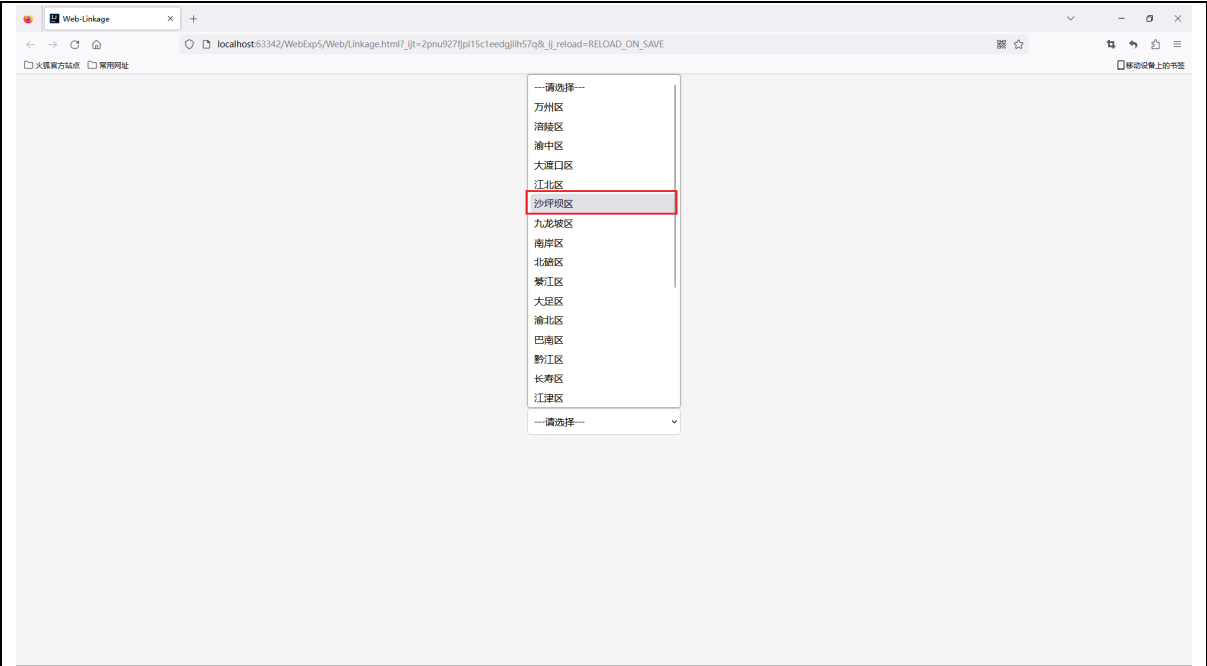


②省市联动功能

a.示例 1 重庆市沙坪坝区

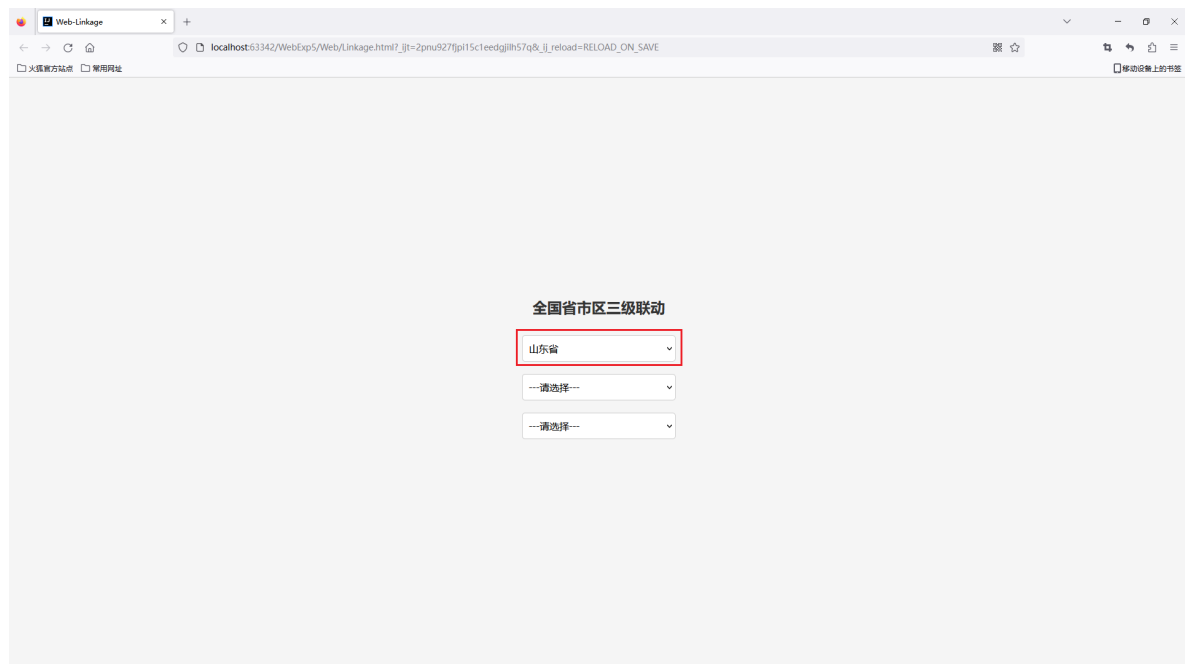
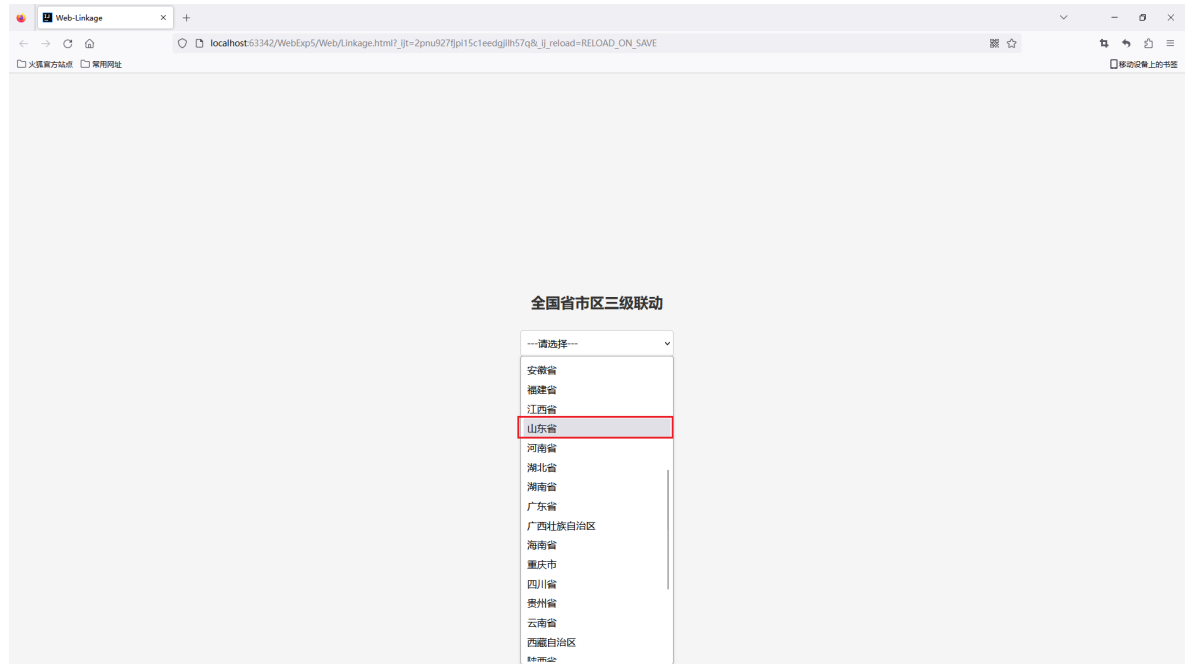


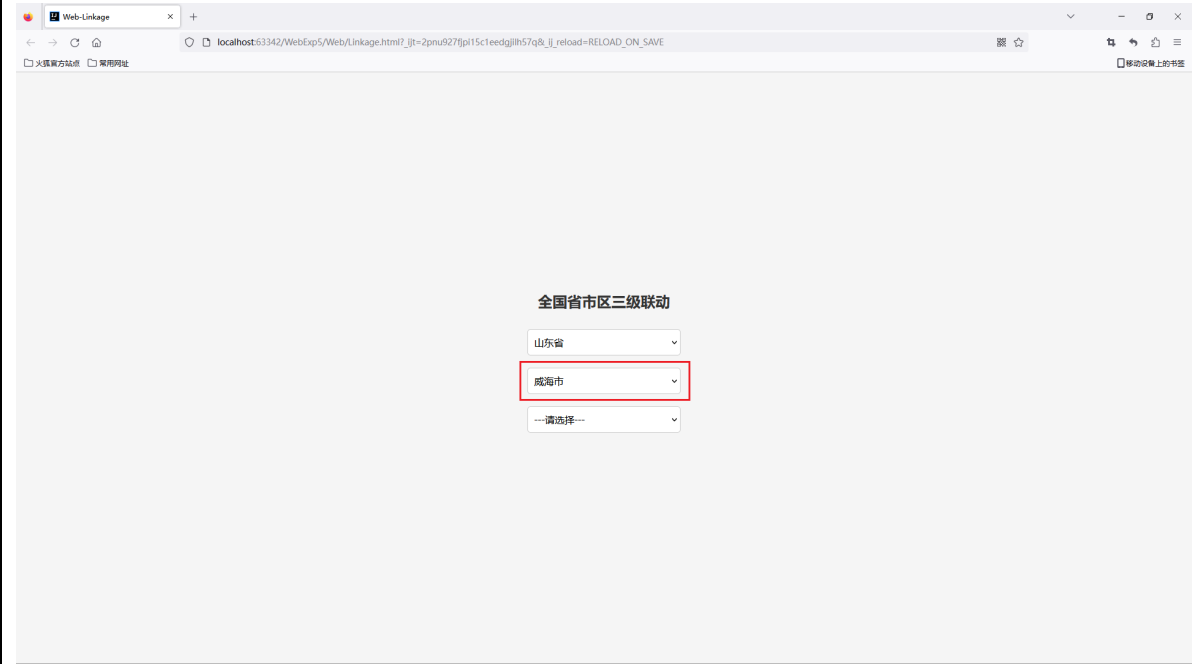
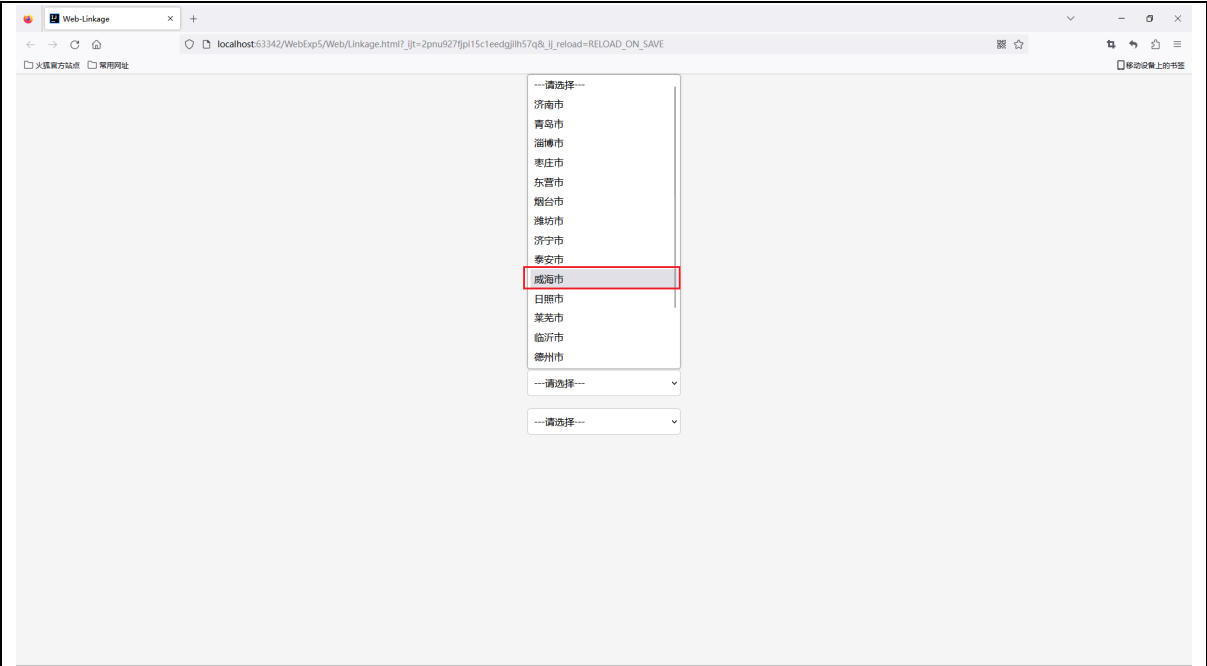


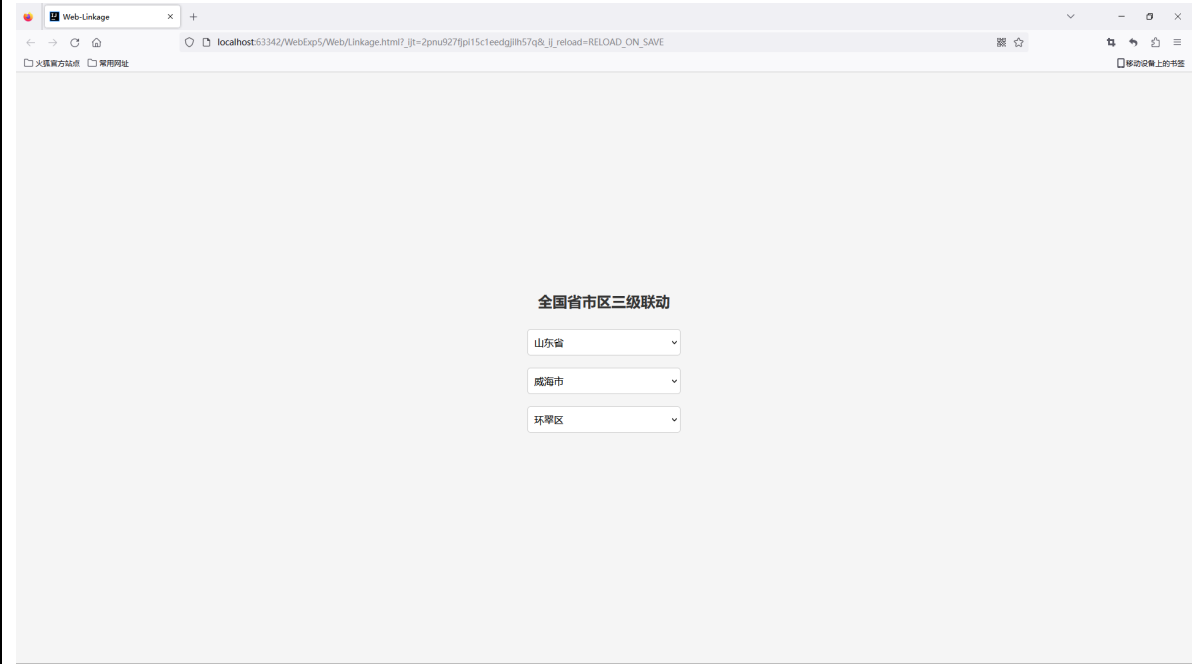
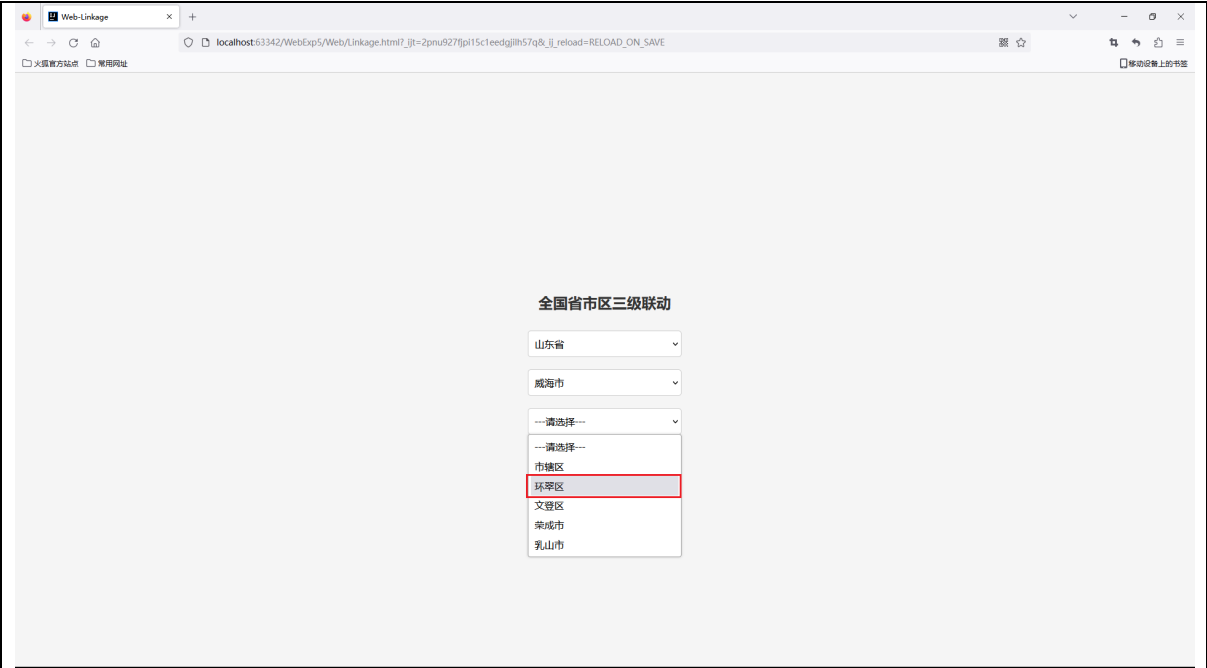


利用 Ajax 技术成功实现动态的、实时的省市区三级联动功能，不需要每次进行刷新。

b.示例 2 山东省威海市环翠区







利用 Ajax 技术成功实现动态的、实时的省市区三级联动功能，不需要每次进行刷新。

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，且能正常实现省市区三级联动功能。

2. 实验分析：

（1）三级联动省市选择页面设计

该三级联动省市选择页面的设计充分考虑了用户体验和可用性，采用了现代和简洁的设计元素，以使其易于操作和吸引人。以下是一些主要设计元素：

①背景和颜色方案： 页面背景采用淡色调，使选项内容清晰可见。选择框的背景色为白色，突出显示内容。字体颜色采用深灰色，确保良好的可读性；

②布局设计： 页面的布局经过精心设计，采用弹性布局，使其在不同屏幕尺寸上都能够良好地呈现。这有助于确保用户在不同设备上都能够轻松访问和使用；

③下拉选择框样式： 下拉选择框具有适度的宽度，内边距和边框，使其在视觉上更吸引人且易于点击。这有助于用户明确了解如何与页面交互；

④选项样式： 选项的字体大小适中，确保清晰可读。此外，选项的下拉滚动条采用了自定义的样式，增加了页面的视觉吸引力。

（2）三级联动功能

成功实现了省、市、区三级联动选择功能。当用户选择一个省份后，城市和区域选择框会根据所选省份的不同而动态加载对应数据，使用户能够方便快速地选择目标地区。

（3）实时数据更新

利用 Ajax 技术，实现了实时数据的异步加载和更新，确保了页面的动态性。用户的选择会立即触发相关数据的加载和更新。

（4）响应式设计

三级联动省市选择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，这有助于更好地适应用户的需求，无论用户使用的是桌面计算机、平板还是手机，确保了用户体验的一致性。

（5）总结和展望

通过这个实验，我成功地创建了一个三级联动的省市选择页面。实验中我利用了 HTML、CSS、JavaScript 和 Ajax 等技术，使用户可以动态选择目标地区，同时也为页面添加了响应式设计，确保在不同屏幕和设备上都能够良好展示。

未来，可以进一步改进三级联动的省市选择页面，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，还可以进一步改进和扩展这个实验项目，以提供更多功能和改进用户体验，比如：

①搜索功能：添加搜索功能，让用户能够直接搜索并选择目标地区，提高了用户体验；

②数据优化：优化地区数据的存储和加载方式，以减少页面加载时间，特别是在大数据量情况下；

③多语言支持：提供多语言支持，以适应国际用户，允许用户在不同语言之间切换；

④自定义主题：允许用户选择自定义的主题或颜色方案，以满足个性化需求；

⑤数据更新机制：建立一个自动数据更新机制，确保地区数据始终保持最新；

⑥地图集成：考虑将地区选择与地图集成，以提供更直观的地区选择方式。

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。