

**WEB开发技术**

**期末考核报告**

**题 目： 《小米官网登录及购买》**

**学生姓名 朱广成**

**学生学号**  **2020210883**

**专 业 计算机科学与技术**

**班 级**  **2020702**

**指导教师**   **程志梅**

**2022 年 6月 20日**

**目 录**

[1.项目来源及背景 1](#_Toc11736)

[2.需求分析 1](#_Toc9423)

[3.系统总体设计 1](#_Toc8005)

[4.详细设计与实现 1](#_Toc21827)

[5.总结与体会 1](#_Toc14549)

[参考文献 1](#_Toc23661)

# 项目来源及背景

根据自己手机的一些来源，查询小米官网时发现这个网站可以很好的练习自己学习的知识，项目主要就是模仿小米官网做一个展示界面，其中为了重点练习自己的css知识，争取做到了是与小米官网一样的布局构造，考虑项目的完整性及时间，仅仅展示小米官网100%布局第一眼的所有界面功能及布局，并且针对项目完整也添加了登录注册的界面和购物车界面。网页设置为jsp，因为要写一些java代码绑定到数据库。

**2.需求分析：**

根据要求实现展示小米官网的界面，界面按照小米官网第一眼展现出的东西完全展示出来，功能需求是主要完成了个人用户的登陆注册，以及将购买的商品添加到购物车，性能需求是完成将个人用户的信息保存到数据库中，并把购买的商品也与数据库绑定，并用Java代码从数据建库查询商品信息，并在购物车界面中显示出来，可靠性方面基本实现了需要的功能，对用户登录注册，以及添加购物车中可以实现数据库的连接访问，查询，添加等操作，保证好了数据完整性，并且没有错误，解决了业务需求，用户需求，功能需求。

# **系统总体设计**

**登录界面 绑定数据库注册表数据表**

**绑定form表单，将input标签的style标签设置为submit，提交登陆界面的账号密码，并将form表单提交至另一个jsp界面。**

**账号：**

**check-login.jsp页面 绑定数据库中的注册表**

**密码： 根据登陆界面传入的账号密码，判断表中是否有此账户，以此判断是否正确。**

**账户存在且密码正确： 小米官网主界面（Xiaomi.jsp）**

**账户不存在或密码错误： 提示账户密码错误，并在三秒后自动返回登陆界面。**

**注册界面： 绑定数据库中的注册表**

**功能：注册 --------------------返回登陆界面**

**设置两个input：账户密码 设置一个button类**

**的input，点击跳转登录界面**

**一个input为submit类型，用于**

**提交注册的账户密码，并使用on**

**click引用一个java方法（zc()）**

**添加注册表数据**

**购物车界面： 绑定数据库中的购物车表**

**与主界面的展示商品绑定，当点击商品及会添加至数据库中，购物车界面会从数据库中查询数据并展示在该页面中。**

**主界面：主要的界面展示：**

**上方，左方导航栏 商品的展示及添加购物车**

**点击上方的登录注册，以及购物车标签会跳转到指定的界面。**

**登录 主界面 注册**

**商品添加购物车**

各个页面，事件处理

**客户端 HTTP请求 数据访问存取**

数据库中的各个表

**浏览器 页面文件 返 回数据**

**前端展示**

**中间信息处理层**

小米官网主界面

**登录界面 购物车界面 注册界面**

**数据库设计: 数据库**

**结构设计：**

**登录 关联：账户**

**注册表**

**注册 密码判断登录**

**注册表：**

|  |  |  |
| --- | --- | --- |
| **序号** | **列名** | **类型** |
| **1** | **账户** | **nvarchar(50)** |
| **2** | **密码** | **nvarchar(50)** |

**购物车表：**

|  |  |  |
| --- | --- | --- |
| **序号** | **列名** | **类型** |
| **1** | **商品名称** | **nvarchar(50)** |
| **2** | **单价** | **nvarchar(50)** |
| **3** | **数量** | **nvarchar(50)** |
| **4** | **操作** | **nvarchar(50)** |

# **详细设计与实现**

我先根据之前学习的网课，跟着老师做了小米官网的主界面，当时之学习了css和html的知识，所以界面的总体只用到了html和css，首先在开始引入了一些样式表，和图标字体库，

*<!-- 引入重置样式表 -->*<link rel="stylesheet" href="./CSS/reset.css">  
*<!-- 引入公共样式表 -->*<link rel="stylesheet" href="./CSS/base.CSS">  
*<!-- 引入图标字体库 -->*<link rel="stylesheet" href="./fa/css/all.css">  
*<!-- 引入当前页面的样式表 -->*<link rel="stylesheet" href="./CSS/index.css">

接着设计了网站的图标，根据当时所学知识，网站图标是一般存储在网站的根录下，名字一般叫icon，使用link标签，并引用已经添加的图片，展示效果如下。



接着是创建顶部导航条的容器，创建div标签，设置css样式，在div中对导航条的各个信息采用ul标签中li标签显示，按照小米官网上的导航条一比一还原，具体实现代码如下：



这里面使用a标签使各个字可以点击，但不实现功能，将a标签的href设置为javascript。根据小米官网，在最后一个商城app有一个弹出层，这里需要特意设置，css的样式设置代码如下：

.app:hover .qrcode,  
.app:hover::after{  
 display: block;  
 height: 148px;  
}  
.app .qrcode{  
 */\* display: none; \*/* height: 0;  
 overflow: hidden;  
 position: absolute;  
 width: 124px;  
 left: -40px;  
 line-height: 1;  
 text-align: center;  
 background-color: #fff;  
 box-shadow: 0 0 10px rgba(0, 0, 0, .3);  
 transition: height 0.3s;  
 z-index: 9999;  
}  
*/\* 设置二维码中的文字 \*/*.app .qrcode span{  
 color: #000;  
 font-size: 14px;  
}  
*/\* 设置二维码 \*/*.app .qrcode img{  
 width: 90px;  
 margin: 17px 17px;  
 margin-bottom: 14px;  
}

接着设置购物车，这里对购物车字体设置为input标签，类型为button，并实现点击时跳转购物车界面，用户登录注册字体一样，并分别跳转登录和注册界面。

接下来是创建一个头部的外部容器，这里仿照小米官网，在上册有一个小米的logo，这里直接添加小米logo的图片，接下来是左边各个导航条，依次显示的各个商品分类及展示，左侧的导航条，主要代码如下图：



这里我对其中li标签的class属性都设置为一个名，主要是为了js中获取该元素，弹出具体的商品展示框，实现移动到不同的商品，可以弹出不同的界面，并实现移出后界面消失的效果，具体实现代码如下：

var o1 = document.getElementsByClassName("s1");  
var o2 = document.getElementById("goods2");  
var o3 = document.getElementById("shouji");  
var o4 = document.getElementById("dianshi");  
var arr = new Array();  
o1[0].onmouseover = function () {  
 o2.style.display = "block";  
 o3.style.display = "block";  
 o4.style.display = "none";  
 o2.onmouseover = function () {  
 o2.style.display = "block";  
 o3.style.display = "block";  
 };  
 o2.onmouseout = function () {  
 o2.style.display = "none";  
 o3.style.display = "none";  
 };  
};  
o1[1].onmouseover = function () {  
 o2.style.display = "block";  
 o4.style.display = "block";  
 o3.style.display = "none";  
 o2.onmouseover = function () {  
 o2.style.display = "block";  
 o4.style.display = "block";  
 };  
 o2.onmouseout = function () {  
 o2.style.display = "none";  
 o4.style.display = "none";  
 };  
};  
for (var i = 2; i < o1.length; i++) {  
 o1[i].onmouseover = function () {  
 o2.style.display = "none";  
 };  
}

css设置样式如下：

.left-menu{  
 padding: 20px 0;  
 width: 234px;  
 height: 420px;  
 background-color: rgba(0, 0, 0, .6);  
 position: absolute;  
 z-index: 999;  
 left: -250px;  
 line-height: 1;  
}  
.left-menu li a{  
 display: block;  
 height: 42px;  
 line-height: 42px;  
 color: white;  
 margin-right: 0;  
 padding: 0 30px;  
 font-size: 14px;  
}  
.left-menu li a:hover{  
   
 background-color: #FF6700;  
 color: #fff;  
}  
  
.left-menu a i{  
 float: right;  
 line-height: 42px;  
}  
  
.nav{  
 */\* width: 850px; \*/* height: 100px;  
 padding-left:183px; ;  
   
 line-height: 60px;  
}  
.nav-wrapper li a:hover{  
 color: #FF6700;  
}  
*/\* 设置导航条中的li \*/*.nav-wrapper li a{  
 display: block;  
 font-size: 16px;  
 margin-right: 20px;  
}  
  
.nav .show-goods:hover ~.goods-info,  
.goods-info:hover{  
 height: 228px;  
 border-top: 1px solid rgb(224, 224, 224);  
 box-shadow: 0 5px 3px rgba(0, 0, 0, .2);  
}  
.all-goods{  
 visibility: hidden;  
}

展示效果如下：



左侧弹出层的代码如下：

#goods2:hover{  
 display: block;  
}  
  
#goods2{  
 display: none;  
 height: 460px;  
 overflow: hidden;  
 width: 1025px;  
 background-color: #fff;  
 position: absolute;  
 top: 0px;  
 left: 250px;  
 z-index: 9999;  
}  
#shouji{  
 display: none;  
}  
#dianshi{  
 display: none;  
}  
  
#goods2 a{  
 float:left;  
 line-height: 40px;  
 padding: 18px 20px;  
 position:relative;  
 width: 80px;  
 height: 60px;  
 margin-bottom: 20px;  
}

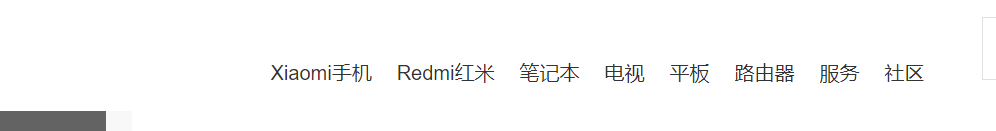
当鼠标移入到手机时展示如下：



移入到电视时：

页面展示为：



还有上方的导航条，依次显示各种商品分类，接着创建一个弹出层，实现鼠标上方的导航条时出现，与左方导航条类似，使用相关js的语句实现效果。页面展示如下：  


这里我只对前三个商品类添加了对应的各种商品，在弹出层中添加三个div盒子，里面添加了各商品，并用a标签实现可点击，这里点击时直接添转另一个界面，该界面实现将点击的商品添加到数据库，这里因为没有找到太好的方法，只能将每一个a标签传入的i不同，再在另一个界面通过判断的i数组添加指定的数据至数据库中，操作有些繁琐麻烦。具体实现代码如下：



在tj.jsp界面中的代码实现功能：

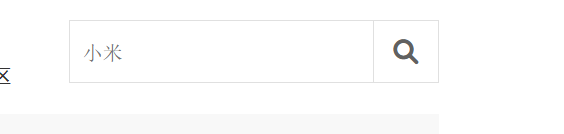
public void pd(String a,String b){  
 try {  
 Class.*forName*( "com.microsoft.sqlserver.jdbc.SQLServerDriver");  
 } catch (ClassNotFoundException e1) {  
 *// TODO Auto-generated catch block* e1.printStackTrace();  
 }  
 String connectionUrl =  
 "jdbc:sqlserver://localhost:1433;"  
 + "databaseName=web小米;"  
 + "user=sa;"  
 + "password=123456;";  
 String sql = "insert into 购物车(商品名称,单价,数量) values (?,?,?)";  
 Connection con;  
 try {  
 con = DriverManager.*getConnection*(connectionUrl);  
 PreparedStatement pst=con.prepareStatement(sql);  
 pst.setString(1,a);  
 pst.setString(2,b);  
 pst.setString(3,"1");  
 pst.executeUpdate();  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
}

pd方法为带参的方法，将商品的名称和价格传入，方法实现将数据添加至数据库中的购物车表中，默认数量为一。

弹出层的css样式设置为下：

.top-goods{  
 display: none;  
 width: 100%;  
 margin-left: 150px;  
}  
.top-goods a{  
 float: left;  
 height: 120px;  
 position: relative;  
}  
.top-goods div{  
 margin-left: 65px;  
   
}  
.top-goods a p{  
 position: absolute;  
 left: 80px;  
 top: 160px;  
 color: #FF6700;  
}  
.top-goods img{  
 width: 150px;  
 height: 110px;  
 margin-top: 30px;  
 margin-left: 30px;  
 padding-right: 40px;  
 */\* border-left: 30px; \*/* border-right: 1px solid #757575;  
}

然后设置了一个搜索框，这里搜索款仅仅是一个展示，添加了一个div盒子框，里面添加一个类型为text的input标签和button标签，button标签添加一个i图标标签，效果展示如下：



接着创建中间总体的界面。

从小米官网中下载了5个界面例图，依然使用li标签添加至div标签中，接着添加底部五个导航点，首先将导航点浮动，并设置大小，里面的导航点设置为a标签，并实现点击和移入时的切换效果。代码如下：

*/\* 设置五个导航点 \*/*.pointer{  
 position: absolute;  
 bottom: 22px;  
 right: 35px;  
}  
.pointer a{  
 float:left;  
 width: 6px;  
 height: 6px;  
 border:3px rgba(255, 255,255, .4) solid;  
 border-radius: 50%;  
 background-color: rgba(0, 0, 0, .4);  
 margin-left: 6px;   
}  
.pointer a.active,  
.pointer a:hover{  
 border-color: rgba(0, 0, 0, .4);  
 background-color:rgba(0, 0, 0, .4) ;  
}

接下来是两个方向按钮

*/\* 设置两个箭头 \*/*.prev-next a{  
 width: 41px;  
 height: 69px;  
 */\* background-color: red; \*/* background-image: url(../img/icon-slides.png);  
 position:absolute;  
 top: 0;  
 bottom: 0;  
 margin: auto 0;  
}  
.prev-next .prev:hover{  
 background-position: 0 0;  
}  
.prev-next .prev{  
 left: 229px;  
 background-position: -84px 0;;  
}  
.prev-next .next{  
 background-position: -125px 0;  
 right: 0;  
}  
.prev-next .next:hover{  
 background-position: -42px 0;  
}

接着添加右侧浮动框，依次添加四个a标签，里面添加图片标签和span标签，

<div class="back-top">  
 <a href="#">  
 <img src="https://i8.mifile.cn/b2c-mimall-media/98a23aae70f25798192693f21c4d4039.png" alt="">  
 <span>手机APP</span>  
 </a>  
 <a href="#">  
 <img src="https://i8.mifile.cn/b2c-mimall-media/55cad219421bee03a801775e7309b920.png" alt="">  
 <span>个人中心</span>  
 </a>  
 <a href="#">  
 <img src="https://cdn.cnbj1.fds.api.mi-img.com/mi-mall/12eb0965ab33dc8e05870911b90f3f13.png" alt="">  
 <span>售后服务</span>  
 </a>  
 <a href="#">  
 <img src="https://i8.mifile.cn/b2c-mimall-media/4f036ae4d45002b2a6fb6756cedebf02.png" alt="">  
 <span>人工客服</span>  
 </a>  
 <a href="#">  
 <img src="https://i8.mifile.cn/b2c-mimall-media/d7db56d1d850113f016c95e289e36efa.png" alt="">  
 <span>购物车</span>  
 </a>  
</div>

css样式设置如下：

*/\* 设置右侧浮动框的元素 \*/*.back-top{  
 position: absolute;  
 */\* display: block; \*/* width: 82px;  
 height: 100px;  
 right: 0;  
 bottom: 450px;  
 margin-top: -1px;  
 background-color: #fff;  
 border: 1px solid #f5f5f5;  
 */\* text-align: center; \*/*}  
  
.back-top a{  
 display: block;  
 width: 82px;  
 height: 90px;  
 border: 1px solid #f5f5f5;  
 background-color: #fff;  
 margin-top: -1px;  
 text-align: center;  
 color: #757575;  
 position: relative;  
}  
  
.back-top img{  
 position: absolute;  
 top: 10px;  
 left: 20px;  
 width: 30px;  
 height: 30px;  
 */\* transition: opacity .3s; \*/*}  
.back-top a{  
 text-decoration: none;  
}  
.back-top span{  
 color: #757575;  
 -webkit-transition: color .3s;  
 transition: color .3s;  
 position: absolute;  
 top: 60px;  
 left: 12px;  
 font-size: 14px;  
}

至此完成了主界面的页面展示和功能实现，这里主要是一个网页界面的展示，实现的是商品添加至购物车，以及通过点击跳转登录注册界面，更多是的css样式的设置和一些js的使用，实现动态交互界面。界面基本实现了一比一还原出了小米官网的界面，但因为一些知识能力的欠缺和时间的考虑并没有完成最初设想的全部功能和商品的全部添加。

登录界面的详细设计和实现

这个界面没有注重于页面效果的展示，仅仅是一个简单的div盒子标签，设置了一些简单的CSS样式，首先添加一个表单form实现对提交的账号密码收集，再添加一个类型为submit的input标签，点击该按钮将账号密码input框的值传入到另一个jsp界面，最后表单外添加一个button类型的标签，实现点击跳转到注册界面。

该界面的具体实现代码：

<div style="margin-left:100px;border: 3px blue solid;width: 500px;height: 350px">  
 <form method="post" action="check-login.jsp">  
 <h1 style="margin-left: 150px;color: #96fbc4">登录界面，亲</h1>  
<input type="text" value="" placeholder="邮箱/小米ID/手机号码" class="dl" name="账号1">  
<br>  
<br>  
<input type="password" placeholder="密码" class="dl" name="密码1">  
<br><br>  
<input type="radio" name="ytx" value="yes" style="margin-left: 150px;float: left"><p>已阅读并同意小米帐号 用户协议 和 隐私政策</p>  
<br>  
<input type="submit" value="登录" style="background-color:#ff6700;border:none;width: 200px;margin-left: 150px" >  
</form>  
 <br>  
<input type="button" onclick="window.location.href='register.jsp';"value="一键注册" style="background-color:#FF6700;margin-left: 150px;border: none;width: 200px">  
</div>

再判断界面中，使用<%%>编写java代码判断登陆界面传入的账号密码是否正确，首先使用requeset.getParameter得到传入的input中的值，再调用方法实现判断。当传入的账户密码正确时直接 跳转官网主界面，当不符合要求时会在页面提示账户密码错误，并在3s后返回登陆界面，具体实现代码如下：

<%!  
 String zh,mm,s1;  
 int flag=1;  
 public void judge(){  
 try {  
 Class.*forName*( "com.microsoft.sqlserver.jdbc.SQLServerDriver");  
 } catch (ClassNotFoundException e1) {  
 *// TODO Auto-generated catch block* e1.printStackTrace();  
 }  
 String connectionUrl =  
 "jdbc:sqlserver://localhost:1433;"  
 + "databaseName=web小米;"  
 + "user=sa;"  
 + "password=123456;";  
 String sql = "select *\** from 注册 where 账号='"+zh+"'";  
 Connection con;  
 try {  
 con = DriverManager.*getConnection*(connectionUrl);  
 Statement sta=con.createStatement();  
 ResultSet re=sta.executeQuery(sql);  
 while (re.next()){  
 s1=re.getString("密码");  
 }  
 if (s1.equals(mm))  
 flag=0;  
 else flag=1;  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
%>  
<%  
 zh=request.getParameter("账号1");  
 mm=request.getParameter("密码1");  
 judge();  
 if (flag==0){  
*// session.setAttribute("");* s1=null;  
 response.sendRedirect("Xiaomi.jsp");  
 }  
 else  
 {  
 out.print("<h2><div style=text-align:center;>账号或密码错误<h2></div>");  
 out.print("<h2><div style=text-align:center;>3秒后返回登录页面<h2></div>");  
 response.setHeader("refresh","3;url=http://localhost:8080/web\_war\_exploded/mi/%E7%99%BB%E5%BD%95.jsp");  
 }  
%>

注册界面：

该界面与登陆界面类似，界面的设计也比较简单，只添加了两个input框进行传入注册的账户密码，使用submit类型的input标签，实现点击运行添加的方法。以及一个button类型的input标签，实现点击跳转登陆界面，该代码是绑定注册表，将input中的值作为账户账号和密码传入到注册表中。具体实现代码如下：

function *zc*() {  
 <%  
 String zh=request.getParameter("账号");;  
 String mm=request.getParameter("密码");  
 if (zh!=null&mm!=null){  
 try {  
 Class.*forName*( "com.microsoft.sqlserver.jdbc.SQLServerDriver");  
 } catch (ClassNotFoundException e1) {  
 *// TODO Auto-generated catch block* e1.printStackTrace();  
 }  
 String connectionUrl =  
 "jdbc:sqlserver://localhost:1433;"  
 + "databaseName=web小米;"  
 + "user=sa;"  
 + "password=123456;";  
 String sql = "insert into 注册(账号,密码) values (?,?) ";  
 Connection con;  
 try {  
 con = DriverManager.*getConnection*(connectionUrl);  
 PreparedStatement pst=con.prepareStatement(sql);  
 pst.setString(1,zh);  
 pst.setString(2,mm);  
 pst.executeUpdate();  
  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 }  
 %>  
 *alert*("注册成功");  
 }  
 </script>  
</head>  
<link rel="icon" href="./img/favicon.ico">  
<body>  
<form>  
<div style="margin-left:100px;border: 3px blue solid;width: 500px;height: 350px">  
<h1 style="margin-left: 150px;color: #96fbc4">注册界面，亲</h1>  
<input type="text" placeholder="邮箱/小米ID/手机号码" class="dl" name="账号">  
<br>  
<br>  
<input type="password" placeholder="密码" class="dl" name="密码">  
<br><br>  
 <input type="submit" onclick="*zc*()"value="点击注册" style="background-color:#FF6700;margin-left: 150px;border: none;width: 200px">  
 <br><br>  
 <input type="button" onclick="window.location.href='登录.jsp';" value="返回登陆界面" style="background-color:#FF6700;margin-left: 150px;border: none;width: 200px">  
</div>

最后是购物车界面的实现，该界面主要设计了两个div标签盒子，实现上下分离，在下界面添加了table标签，用于规范商品的信息输出。商品信息是从数据库中的购物车表中查询数据，购物车的数据是在主界面时点击指定商品添加的，主界面添加商品的具体实现代码如下：



购物车界面的实现代码如下：

<%@ page import="java.sql.\*" %>  
<%@ page import="java.util.ArrayList" %>  
<%@ page contentType="text/html;charset=UTF-8" language="java" %>  
<html>  
<head>  
 <title>我的购物车-小米商城</title>  
 <style>  
 .box1 img{  
 position:absolute;  
 width: 55px;  
 height: 55px;  
 display: block;  
 background-position: center;  
 margin-left: 250px;  
 margin-top: 10px;  
 */\*transition: left 0.3s;\*/* }  
 .box1 h1{  
 position: absolute;  
 display: block;  
 background-position: center;  
 margin-left: 330px;  
 margin-top: 10px;  
 }  
 .box2{  
 background-color: #F5F5F5;  
 width: 100%;  
 height: 100%;  
 }  
 table{  
 width: 80%;  
 position: center;  
 height: 80px;  
 margin-left: 100px;  
 background-color: #FF6700;  
 font-size: 20px;  
 border: 1px solid black;  
 }  
 p{  
 float: right;  
 padding-right: 80px;  
 margin-top: 30px;  
 }  
 </style>  
 <script>  
 <%  
 String a=null,b=null,c=null;  
 try {  
 Class.*forName*( "com.microsoft.sqlserver.jdbc.SQLServerDriver");  
 } catch (ClassNotFoundException e1) {  
 *// TODO Auto-generated catch block* e1.printStackTrace();  
 }  
 String connectionUrl =  
 "jdbc:sqlserver://localhost:1433;"  
 + "databaseName=web小米;"  
 + "user=sa;"  
 + "password=123456;";  
 String sql = "select *\** from 购物车";  
 Connection con;  
 try {  
 con = DriverManager.*getConnection*(connectionUrl);  
 Statement sta=con.createStatement();  
 ResultSet re=sta.executeQuery(sql);  
 while (re.next()){  
 a=re.getString("商品名称");  
 b=re.getString("单价");  
 c=re.getString("数量");  
 %>  
  
 <%  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
  
 %>  
  
 </script>  
</head>  
<link rel="icon" href="./img/favicon.ico">  
<body>  
<div class="box1" style="border-bottom: 3px orange solid;width: 100%;height: 80px;">  
 <img src="./img/logo-mi2.png" alt="">  
 <h1>我的购物车</h1>  
 <p>个人中心</p>  
 <p>我的订单</p>  
</div>  
<div class="box2">  
 <table>  
 <tr style="border-bottom: 1px solid black;">  
 <td width="30%">商品名称</td>  
 <td width="20%">单价</td>  
 <td width="20%">数量</td>  
 <td width="20%">操作</td>  
 </tr>  
 <tr>  
 <td width="30%"><%=a%></td>  
 <td width="20%"><%=b%></td>  
 <td width="20%"><%=c%></td>  
 <td width="20%">删除</td>  
 </tr>  
 </table>  
</div>  
</body>  
</html>

至此完成了整个项目的所有功能，基本可以实现界面的展示，与小米官网中的页面一比一还原，在登录注册界面没有注重于页界面的展示，而是与数据库建立连接，实现用户的注册登录，在主界面中对一些商品通过a标签实现向数据库中的购物车表添加商品数据，购物车界面就是展示数据库中的商品信息输出出来。

# **总结与体会**

这次web大作业是我第一次完整的设计并完成一个web项目，我从选题中首先选了一个注重于css样式的小米官网，当时设计时还没学到js，jsp和与数据库连接等操作，所以着重设计了界面效果，自己在css的设计时熟练的掌握了各种标签的用法和使用，加强了各个标签的效果理解，在设计时为了达到要求，一种在不停的变换一些元素的大小，位置，效果展示，也在做的过程中学会了引用外部图标标签，简化了在设计图片方面的挑选和设计。在布局方面大多数采用的是div标签，通过对div的css的样式设计实现需要的样式设计，对元素浮动，位置，内边框，外边距的使用有了更深一步的掌握，在学习js，jsp和与数据库的连接等操作后，又完善了自己的项目，添加了登陆注册界面，与数据库的表绑定，存入和查询数据，以及通过jsp的相关技术，完成界面间的交互和跳转。添加购物车界面，实现购物功能，使用js技术，实现界面的交互功能，当鼠标移入，点击，移出相关元素时，页面发生指定的改变。通过这些天的练习自己对web相关的技术有了更深的兴趣，也在作业中不断体会到解决困难的喜悦，自己掌握了更多的知识，对自己一个学期来的学习有了一个总结性，系统性的认识。

# **参考文献**

1. **《Web程序设计》—————————中国工信出版集团**
2. **CSDN社区—————————————百度**
3. **尚硅谷——————————————百度**