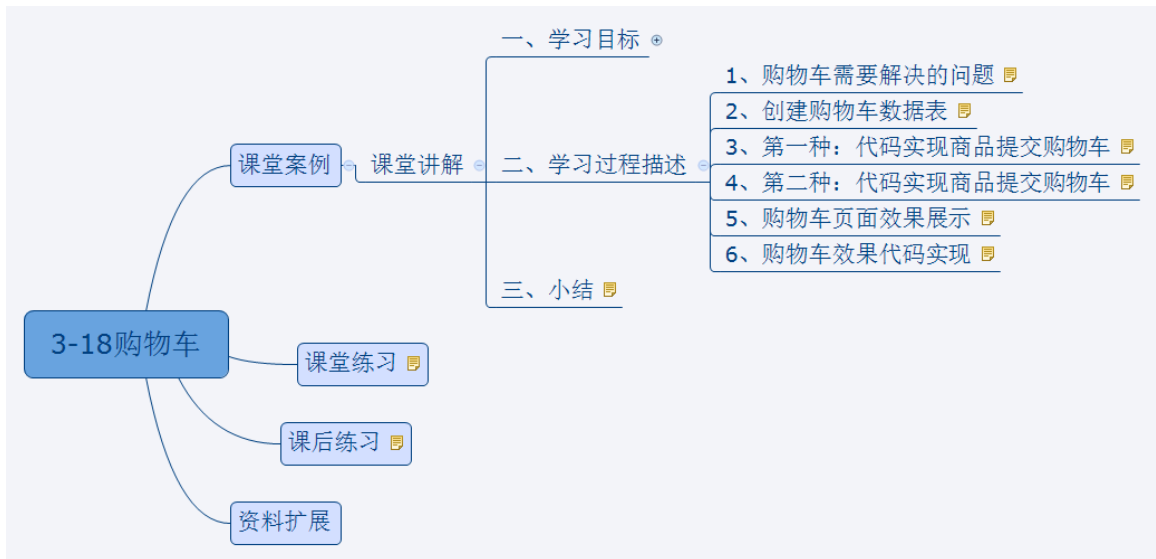


3-18购物车

3-18购物车	1
1. 课堂案例	2
课堂讲解	2
一、学习目标	2
商品购物车的实现	2
二、学习过程描述	2
1、购物车需要解决的问题	2
2、创建购物车数据表	3
3、第一种：代码实现商品提交购物车	4
4、第二种：代码实现商品提交购物车	5
5、购物车页面效果展示	7
6、购物车效果代码实现	9
三、小结	15
2. 课堂练习	15
3. 课后练习	15
4. 资料扩展	15



1. 课堂案例

课堂讲解

一、学习目标

商品购物车的实现

二、学习过程描述

1、购物车需要解决的问题

商城的购物车, 就是类似进入一家超市那个购物小车车一毛一样, 还木有到付款, 随便挑选商品暂存商品的地方

经历的事情也一样

- 1、可以把之前挑选但是现在不要的商品放回货架
- 2、感觉特别喜欢想多拿几个
- 3、算出整个购物车车商品: 数量 \times 单价 = 小计
- 4、显示整个购物车合计价格

网上商城购物车实现一般有两种思路去完成, 一般根据你的业务需要去选择最优:

第一种:

- 1、用户前面不强制用户登陆账号, 比如不需要用户登陆自己账号可以去访问首页、商品列表页、商品详情页、点击放入购物车。

2、直到用户要进入购物车, 进入个人中心, 订单中心这些私密页面才需要要求用户登陆自己账号

针对这两点, 一般情况解决的方案

1、用户还没有登陆时, 存入session或其他方式的临时存储; 等用户登陆后判断是否产生了一些数据, 存在数据则把数据存入数据库的相应列表里

注: 这种做法稍微复杂点, 看用户实际需要决定

第二种:

1、在详情页点击商品放入购物车时提示用户登陆, 如果没有登陆则直接跳转到登陆页面让你先登陆了再来加入购物车

2、当然个人私密页面需要访问也要登陆账号的

注: 这种做法稍微简便点, 有产生数据直接存入加上你个人用户标签的数据插入数据表中
这种做法可以省很多脑力去泡妞用。。。。啦啦, 啦啦

2、创建购物车数据表

购物车就是临时存放用户还未付款的一些信息, 不可能一个人建一张数据表存放, 只能建一个整个。所以这个时候插入的数据是谁的, 那么需要建一个关联用户的id字段

然后是商品id的字段、购买数量、创建时间等信息

当用户需要删除某商品时那数据消失、当用户提交订单变成等待付款状态时, 该用户刚提交的数据在购物车的记录相应去除

字段如下图所示:

rl_cart @goshopping (localhost) - 表

文件 编辑 窗口 帮助

新建 保存 另存为 添加栏位 插入栏位 删除栏位 主键 上移

栏位 索引 外键 触发器 选项 注释 SQL 预览

名	类型	长度	小数点	允许空值 (
id	int	20	0	<input type="checkbox"/>	1
userid	int	20	0	<input type="checkbox"/>	
goodsid	int	20	0	<input type="checkbox"/>	
num	int	20	0	<input type="checkbox"/>	
ctime	int	20	0	<input type="checkbox"/>	

默认:

注释: ...

☒ 自动递增

☒ 无符号

☐ 填充零

3、第一种：代码实现商品提交购物车

就是需要付款时才需登录购物车的代码，如下

```
<script type="text/javascript">
    $('#btn_gwc').click(function() {
        // 测试点击购物车有没有反应
        // alert(12345)
        // 获取商品id、选择数量
        var goods_id = $('#goods_id').val();
        var num = $('#txt').val();
        // 测试能否取到商品id和选择数量
        // alert(goods_id+'-'+num);
        $.post("{:U('cart')}",{goods_id:goods_id,num:num},function(data){
            // 请求后能否到这里来
            // alert(123);
            if(data==1) {
                alert('添加成功');
            } else {
                alert('添加失败');
            }
        });
    });
</script>
```

当一个用户登陆成功后, 要去判断session中是否有关于购物信息要存入购物车,

如果有则操作存购物车数据表里, 存成功后删除session这方面的信息

如果没有需要存储数据那么正常跳转登陆

参考代码如下:

```
public function cart()
{
    if(IS_AJAX) {
        // 判断是否登录了
        // 登录了存数据库
        if(!empty(session('user_id'))){
            $data['userid'] = session('user_id');
            $data['goodsid'] = I('goods_id');
            $data['num'] = I('num');
            $data['ctime'] = time();
            $result = M('cart')->add($data);
            $this->ajaxReturn($result);
        }
        // 未登陆存session, 用户登陆瞬间判断session是否有商品信息, 有操作session商品信息存数据库
    } else {
        $goods[] = array(
            'goodsid'=>I('goods_id'),
            'num'=>I('num')
        );
        session('goods',$goods);
        $this->ajaxReturn(1);
    }
}

// 走账户和密码都对分支
session('user_id', $info['id']);
session('user name', $info['username']);
if(!empty(session('goods'))){
    foreach(session('goods') as $val) {
        $data['userid'] = session('user_id');
        $data['goodsid'] = I('goods_id');
        $data['num'] = I('num');
        $data['ctime'] = time();
        $result = M('cart')->add($data);
    }
    session('goods',null);
}
$this->ajaxReturn(2);
```

4、第二种：代码实现商品提交购物车

这种购物车做法就是省略了临时能存储购物信息的过程, 那么有加购物车没有登陆会提示你先登陆, 登陆了正常放购物车

那么所有的购物车信息就都在购物车数据表里, 操作这个数据表就好了, 没有冗余的操作, 简单很多, so
easy 妈妈再也不担心我的学习了.....

参考代码如下:

```
// 获取商品id、选择数量
var goods_id = $('#goods_id').val();
var num = $('#txt').val();
// 测试能否取到商品id和选择数量
// alert(goods_id+'-'+num);
$.post("{:U('cart')}", {goods_id:goods_id,num:num}, function(data) {
    // 请求后能否到这里来
    // alert(123);
    /*if(data==1) {
        alert('添加成功');
    } else {
        alert('添加失败');
    }*/
    if(data==0) {
        layer.alert('请先登陆', {end: function(data) {
            location.href="{:U('User/login')}";
        }});
    } else if(data==1) {
        layer.alert('添加成功');
    } else {
        layer.alert('添加失败');
    }
});
```

```

if(!empty(session('user_id'))){
    $cart = M('cart');
    $data['userid'] = session('user_id');
    $data['goodsid'] = I('goods_id');
    // 判断之前有没有加入购物车
    $info = $cart->where($data)->find();
    // 存在则数量增
    if(!empty($info)) {
        $res = $cart->where(array('id'=>$info['id']))->setInc('num',I('num'));
        if($res) {
            // 数量添加成功
            $this->ajaxReturn(1);
        } else {
            // 数量添加失败
            $this->ajaxReturn(2);
        }
    } else {
        $data['num'] = I('num');
        $data['ctime'] = time();
        $res = M('cart')->add($data);
        if($res) {
            // 数量添加成功
            $this->ajaxReturn(1);
        } else {
            // 数量添加失败
            $this->ajaxReturn(2);
        }
    }
} else {
    // 走先登陆分支
    $this->ajaxReturn(0);
}
}

```

5、购物车页面效果展示

购物车放入产品后, 效果如下, 那么此时开始要实现的功能就是:

- 1、点全选, 能把所有购物车商品选中, 然后把选中的商品算出商品总金额
- 2、点全选后再点全选旁边的删除也能全部删除所选商品信息
- 3、单个商品数量也照样能实现数量加、数量减
- 4、点一系列商品的右侧的删除按钮能删除该商品的信息
- 5、实现点上一步和下一步跳转功能

注:为了客户体验更好, 一般都是通过js异步请求来实现这些操作

购物车列表				
<input type="checkbox"/> 全选	商品名称	数量	小计	操作
<input type="checkbox"/>	 欧莱雅洗发水	- 1 +	¥25	
<input type="checkbox"/>	 自然乐园芦荟胶	- 1 +	¥20	

☐ 全选 | 删除

商品数量总计：件

商品金额总计（不含运费）：**¥ 0**

上一步

下一步

购物车列表				
<input type="checkbox"/> 全选	商品名称	数量	小计	操作
<input checked="" type="checkbox"/>	 欧莱雅洗发水	- 3 +	¥75	
<input type="checkbox"/>	 自然乐园芦荟胶	- 1 +	¥20	





☐ 全选 | 删除

商品数量总计：3件

商品金额总计（不含运费）：**¥ 75**

上一步

下一步

购物车列表				
<input type="checkbox"/> 全选	商品名称	数量	小计	操作
<input checked="" type="checkbox"/>	 欧莱雅洗发水	- 3 +	¥75	
<input checked="" type="checkbox"/>	 自然乐园芦荟胶	- 2 +	¥40	
<div> <div> <input checked="" type="checkbox"/> 全选 删除 </div> <div> 商品数量总计：5件 商品金额总计（不含运费）：¥ 115 </div> <div> 上一步 下一步 </div> </div>				

6、购物车效果代码实现

实现5那些页面效果, 参考代码如下:

下面这段是购物车数量加、数量减的代码段

```

<script type="text/javascript">
    $(function(){
        //数量加
        $(".num_jia").each(function(a){
            $(this).click(function(){
                var i=parseInt($(".txt_num").eq(a).val());
                var num = parseInt($(".next").val());
                if(i >= num){
                    i=num;
                    $(".txt_num").eq(a).val(i);
                } else {
                    i=i+1;
                    $(".txt_num").eq(a).val(i);
                }
                // 获取该商品价格
                var price = $(".tr").find('.goods_price').val();
                // 算出小计费用
                var count = i * price;
                // 修改页面小计费用
                $(".tr").find('.price i').html(count);
                //核算费用
                var sum = 0;
                $(".cateid:checked").each(function(){
                    sum += parseInt($(".tr").find('.price i').html());
                });
                $(".sum").html(sum);
                //核算个数
                var total = 0;
                $(".cateid:checked").each(function(){
                    total += parseInt($(".tr").find('.txt_num').val());
                });
                $(".count").html(total);
            });
        });
        //数量减
    });

```

```

//数量减
$(".num_jian").each(function(a){
    $(this).click(function(){
        var i=parseInt($(".txt_num").eq(a).val());
        if(i>1)
        {
            i=i-1;
            $(".txt_num").eq(a).val(i);
        }
        // 获取该商品价格
        var price = $(this).parents("tr").find('.goods_price').val();
        // 算出小计费用
        var count = i * price;
        // 修改页面小计费用
        $(this).parents('tr').find('.price i').html(count);
        //核算费用
        var sum = 0;
        $(".cateid:checked").each(function(){
            sum += parseInt($(this).parents('tr').find('.price i').html());
        });
        $(".sum").html(sum);
        //核算个数
        var total = 0;
        $(".cateid:checked").each(function(){
            total += parseInt($(this).parents('tr').find('.txt_num').val());
        });
        $(".count").html(total);
    });
})
})
</script>

```

下面是一行商品删除的按钮对应的js

```

<!-- 右侧单个删除 -->
<script type="text/javascript">
    $(' .btn_del').click(function() {
        // 调试删除按钮是否可用
        // alert(123456)
        var cartid = $(this).parents("tr").find(".cartid").val();
        // alert(cartid);
        $.post('del',{id:cartid},function(data) {
            if(data==0) {
                layer.alert('删除失败',{end:function() {
                    location.reload();
                }});
            } else {
                layer.alert('删除成功',{end:function() {
                    location.reload();
                }});
            }
        });
    });
</script>
<!-- 全选、反选 -->

```

下面是点下全选, 再点一下全不选的代码段:

```

<!-- 全选、反选 -->
<script type="text/javascript">
    $( '.ck_all' ).click(function() {
        if(this.checked) {
            $(".cateid").attr('checked',true);

            // 核算金额
            var sum = 0;
            $( '.cateid:checked' ).each(function(){
                sum += parseInt($(this).parents('tr').find('.price i').html());
            });
            $( '.sum' ).html(sum);

            //核算个数
            var total = 0;
            $( '.cateid:checked' ).each(function(){
                total += parseInt($(this).parents('tr').find('.txt_num').val());
            });
            $( '.count' ).html(total);
        } else {
            $(".cateid").attr('checked',false);

            // 核算金额
            var sum = 0;
            $( '.cateid:checked' ).each(function(){
                sum += parseInt($(this).parents('tr').find('.price i').html());
            });
            $( '.sum' ).html(sum);

            // 核算个数
            var total = 0;
            $( '.cateid:checked' ).each(function(){
                total += parseInt($(this).parents('tr').find('.txt_num').val());
            });
            $( '.count' ).html(total);
        }
        // $('input').attr('checked',true);
    });
</script>

```

下面是批量删除收集选中商品的购物车列表的id集合, 这样我们搜集来的id集合传到后端就可以删除那些数据表了。

为什么要选择id来收集呢, 因为一个表里的id字段是唯一的, 其他字段可能会出错。

啦啦, 啦啦, 所以我们要知其然然后知其所以然.....我们是最棒的, 么么哒.....

```

<!-- 批量删除 -->
<script type="text/javascript">
    $('#del').click(function(){
        var select = [];
        $('.cateid:checked').each(function(){
            select.push($(this).val());
        });
        if(select==''){
            layer.alert('至少勾选一个');
        } else {
            $.post('del',{id:select},function(data){
                if(data==0){
                    layer.alert('删除失败',{end:function(){
                        location.reload();
                    }});
                } else {
                    layer.alert('删除成功',{end:function(){
                        location.reload();
                    }});
                }
            });
        }
    });
</script>

```

点击单个选框的js代码实现如下:

```

<!-- 点击单个选框 -->
<script type="text/javascript">
    $('.cateid').click(function(){
        //核算费用
        var sum = 0;
        $('.cateid:checked').each(function(){
            sum += parseInt($(this).parents('tr').find('price i').html());
        });
        $('#sum').html(sum);
        //核算个数
        var total = 0;
        $('.cateid:checked').each(function(){
            total += parseInt($(this).parents('tr').find('txt_num').val());
        });
        $('#count').html(total);
    });
</script>

```

后端处理单个删除和批量删除代码

```
public function del()  
{  
    if(IS_AJAX) {  
        $where['id'] = array('in',I('id'));  
        $result = M('cart')->where($where)->delete();  
        $this->ajaxReturn($result);  
    }  
}
```

三、小结

实现商品提交到购物车

购物车页面展示

联动变化相应展示出来

2. 课堂练习

购物数据表的建立

购物车数据的提交

购物车列表的制作

3. 课后练习

从购物车数据提交到提交订单完成

4. 资料扩展