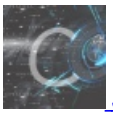
Java web实验购物网站（IDEA开发环境，JavaScript，JSP，Servlet，jQuery，Ajax，MySql等）——实现购物车

原创

weixin_44354613 于 2020-07-15 23:51:02 发布 5058 收藏 140

分类专栏： Java web 文章标签： java jsp web jdbc 数据库

本文链接：https://blog.csdn.net/weixin_44354613/article/details/107372138


Java 同时被 2 个专栏收录

10 篇文章 0 订阅

订阅专栏


web

3 篇文章 0 订阅

订阅专栏

Java web实验购物网站

实验要求

- 掌握静态HTML, CSS，JavaScript,，JSP，Servlet，jQuery，Ajax等技术；
- 本实验为综合实验，学会灵活运用本课程所学的知识解决实际问题；
- 自选主题设计并建立一个购物网站(如花店网站，书店网站等）；
- 不建议使用框架，用基础知识去完成项目；
- 网站入口为:index.jsp。

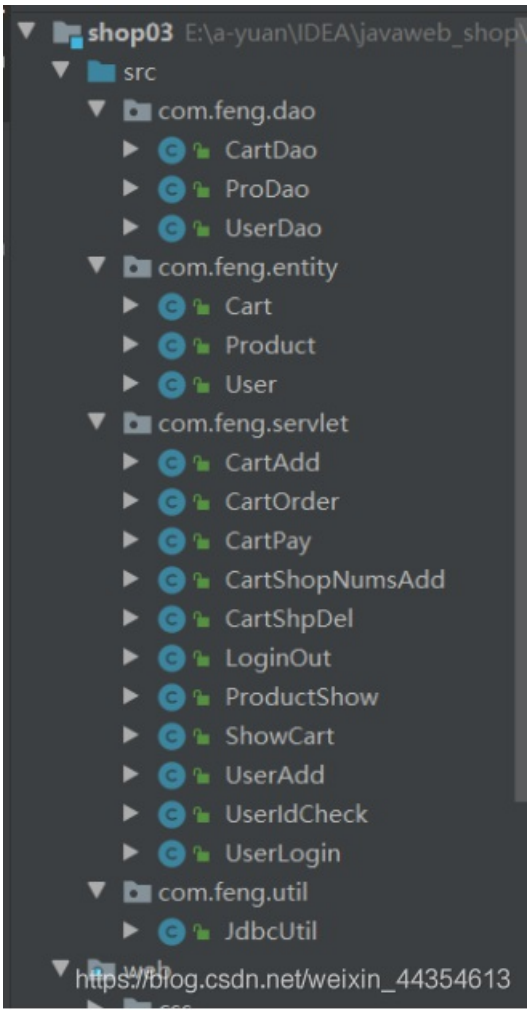
实验开发工具及使用技术

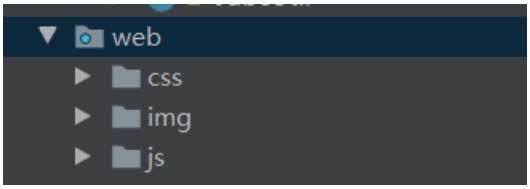
- win10系统，浏览器
- MySql数据库，SQLyog数据库图形化工具
- IDEA集成开发环境（IDEA真的是神器，学生党从ecipse转过来不久，就深深爱上了它）
- Apache Tomcat服务器
- 基础知识HTML, CSS，JavaScript，JSP，Servlet，jQuery，Ajax，JSON，JDBC等等

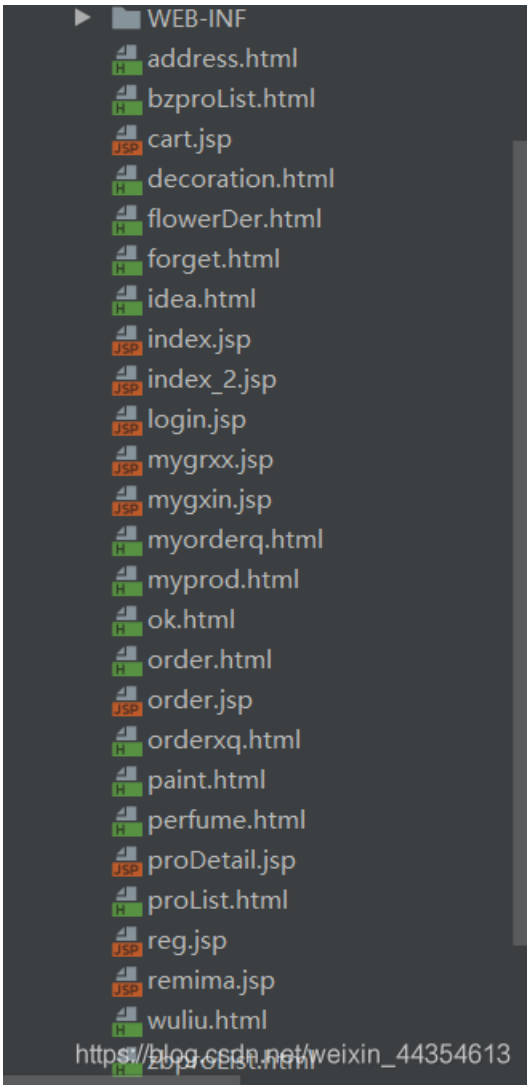
准备工作

- 思考：不用框架，用基础知识来实现。这样一个人完成工作量就很大，所以我决定从网上下载前端静态模版，然后做简单修改来使用。
- 前端准备，基本页面，页面展示都完成后。
- 搭建好开发环境

完整项目目录结构

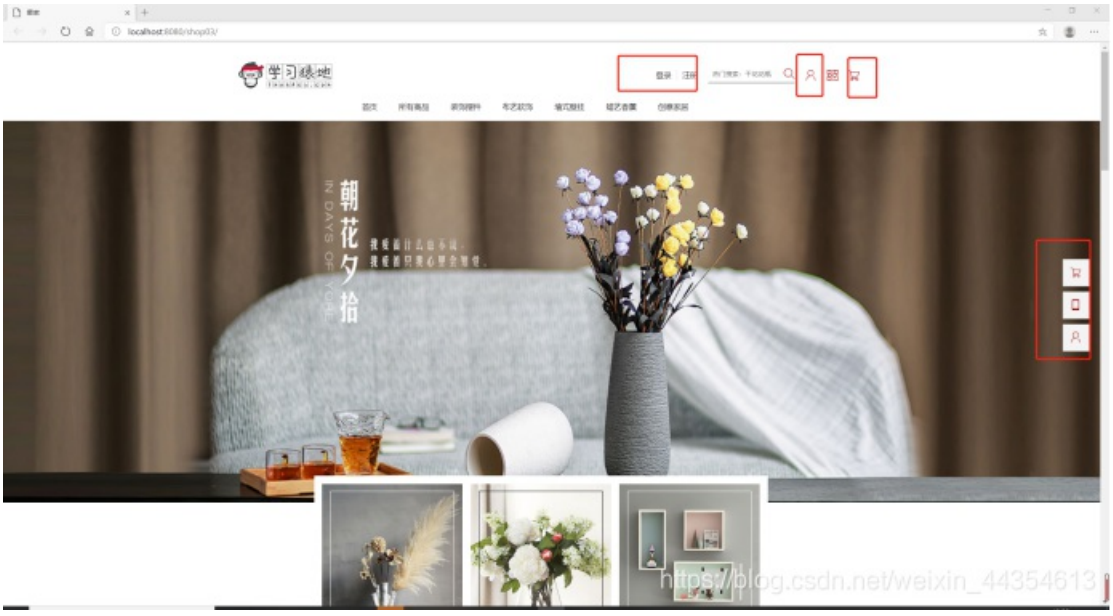






实验结果展示

主页面

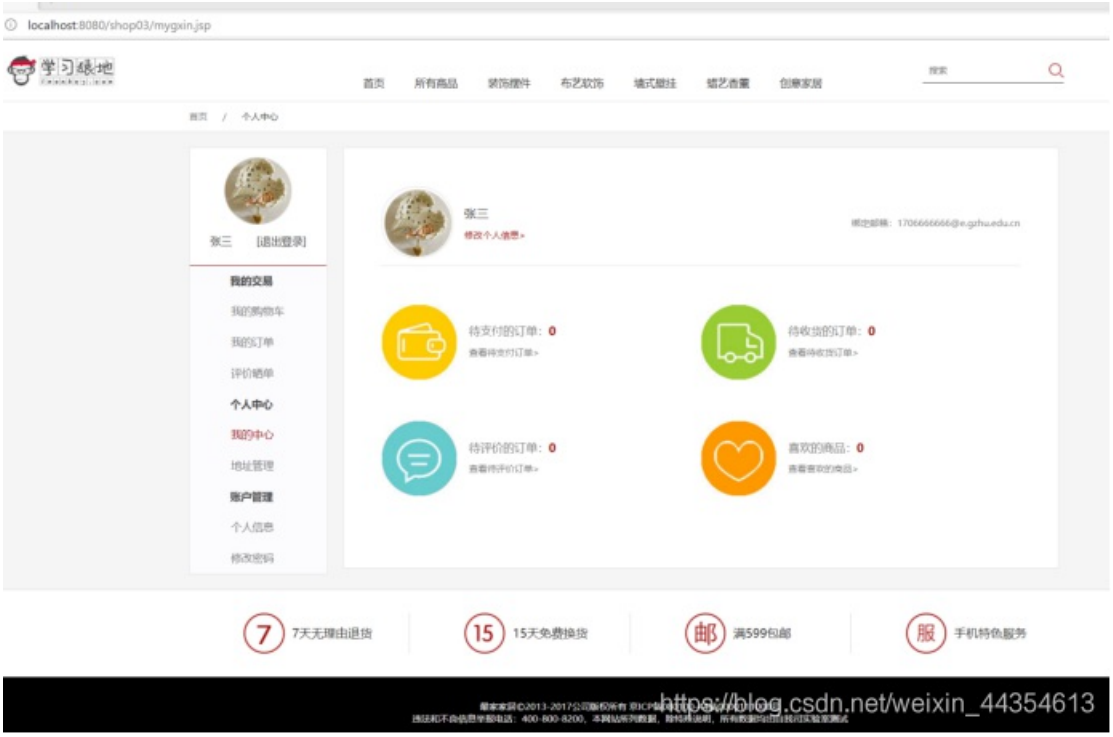


登录页面

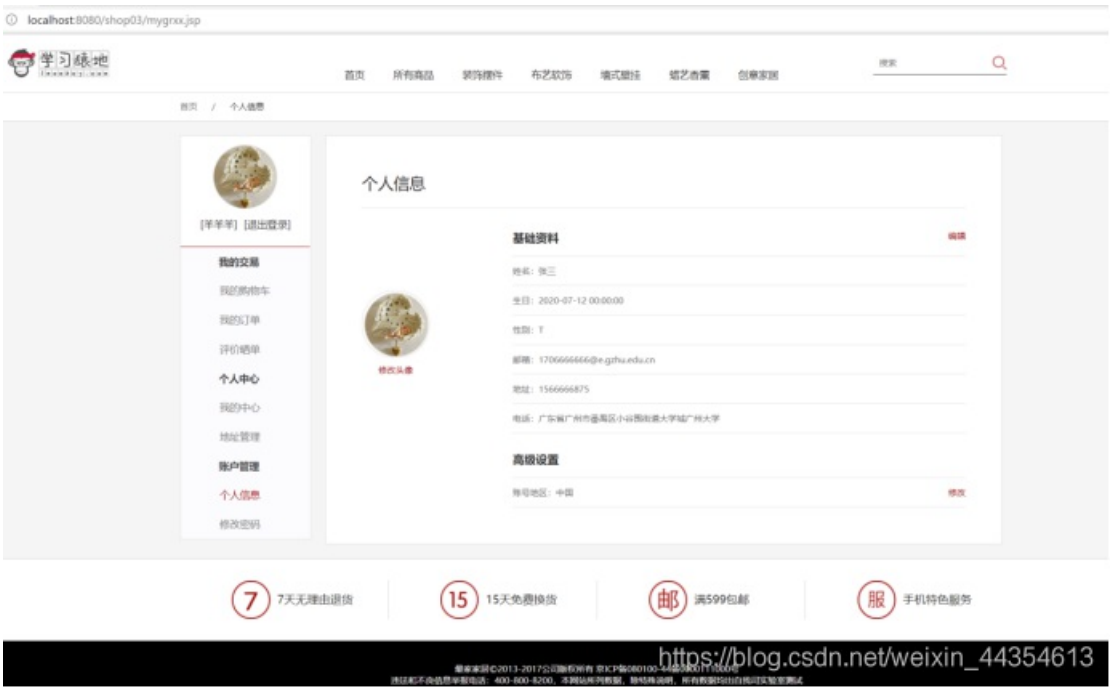

学习猿地
公共场所不建议自动登录，以防账号丢失
用户ID
密码
登 录
免费注册
忘记密码?

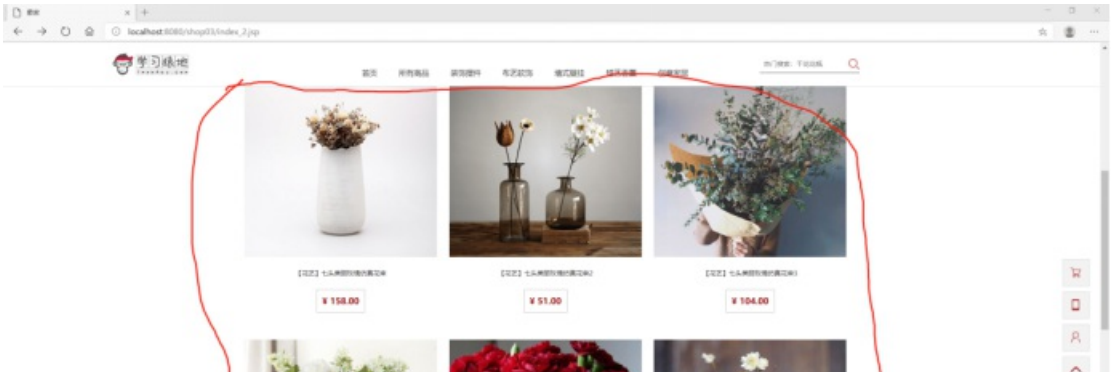
注册页面

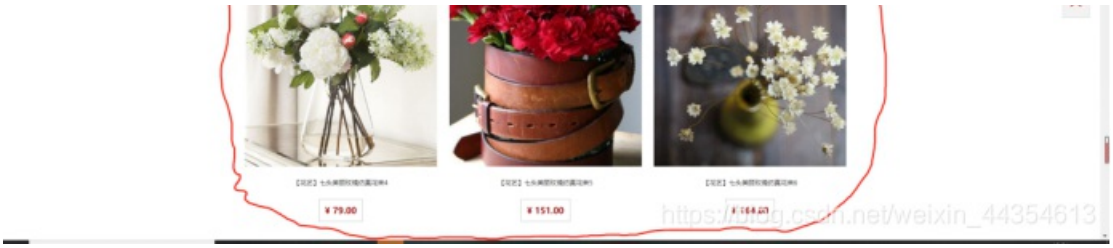

学习猿地
用户注册
请输入用户id
用户名不能为空
请输入姓名
姓名不能为空
请输入密码
密码不能为空
请输确认密码
确认密码不能为空
男 女
yyyy/mm/日
请输入邮箱
请输入手机号码
请输入送货地址
验证码
看不清，换一张
注册
完成此注册，即表明您同意了我们的 <使用条款和隐私策略>
已有账号登录


localhost:8080/shop03/mygxin.jsp
学习基地
首页
所有商品
装饰摆件
布艺软饰
墙式壁挂
蜡艺香薰
创意家居
首页 / 个人中心
张三 [退出登录]
我的交易
我的购物车
我的订单
评价晒单
个人中心
我的中心
地址管理
账户管理
个人信息
修改密码
张三
修改个人信息>
待支付的订单: 0
查看待支付订单>
待收货的订单: 0
查看待收货订单>
待评价的订单: 0
查看待评价订单>
喜欢的商品: 0
查看喜欢的商品>
7天无理由退货
15天免费换货
满599包邮
手机特色服务
https://blog.csdn.net/weixin_44354613

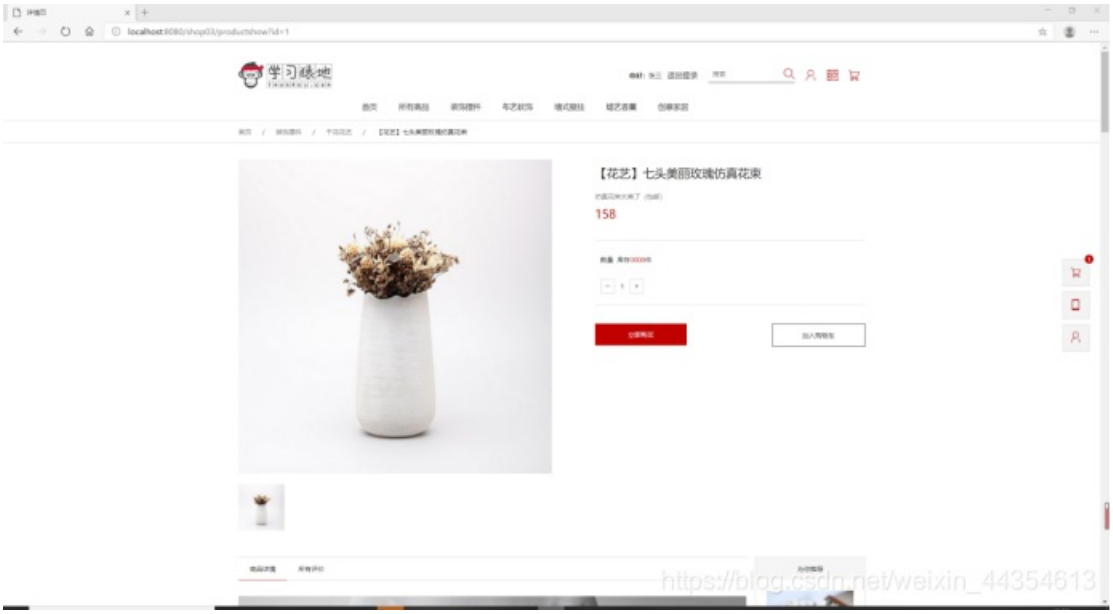
个人信息页面


localhost:8080/shop03/mygrxx.jsp
学习基地
首页
所有商品
装饰摆件
布艺软饰
墙式壁挂
蜡艺香薰
创意家居
首页 / 个人信息
[半羊羊] [退出登录]
我的交易
我的购物车
我的订单
评价晒单
个人中心
我的中心
地址管理
账户管理
个人信息
修改密码
个人信息
基础资料
编辑
姓名：张三
生日：2020-07-12 00:00:00
性别：T
邮箱：1706666666@e.gzhu.edu.cn
地址：1566666875
电话：广东省广州市番禺区小谷围街道大学城广州大学
高级设置
账号地区：中国
修改
修改头像
7天无理由退货
15天免费换货
满599包邮
手机特色服务
https://blog.csdn.net/weixin_44354613

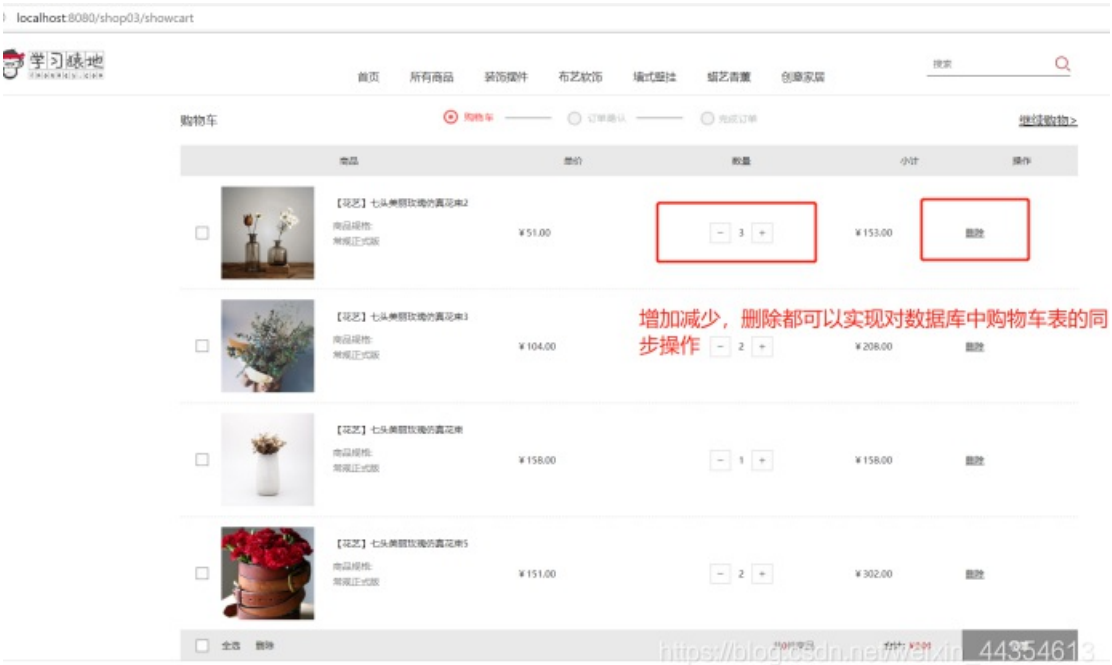

¥ 158.00
¥ 51.00
¥ 104.00



商品详细信息页面



购物车页面



结算页面

实验步骤

- 第一：根据项目需求设计数据库表结构，并且生成对应实体类
  - 用户信息表user，对应实体类User

```sql
/*Table structure for table `user` */

DROP TABLE IF EXISTS `user`;

CREATE TABLE `user` (
  `id` varchar(32) NOT NULL,
  `name` varchar(20) NOT NULL,
  `password` varchar(20) NOT NULL,
  `sex` varchar(1) NOT NULL,
  `birthday` datetime DEFAULT NULL,
  `idenity_code` varchar(60) DEFAULT NULL,
  `email` varchar(60) DEFAULT NULL,
  `mobile` varchar(11) DEFAULT NULL,
  `address` varchar(200) NOT NULL,
  `status` decimal(6,0) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

/*Data for the table `user` */
```

```java
package com.feng.entity;

/**
 * 用户信息类
 */
public class User {
    private String id;//用户id
    private String name;//用户名
    private String password;//用户密码
    private String sex;//用户性别
    private String birthday;//生日
    private String idenity_code;//扩展字段
```

```
    private String email;//电子邮箱
    private String mobile;//联系电话
    private String address;//地址
    private int status;//
```


- 商品信息表product，对应实体类Product

```
/*Table structure for table `product` */

DROP TABLE IF EXISTS `product`;

CREATE TABLE `product` (
  `PRODUCT_ID` int NOT NULL AUTO_INCREMENT,
  `PRODUCT_NAME` varchar(128) NOT NULL,
  `PRODUCT_DESCRIPTION` varchar(512) DEFAULT NULL,
  `PRODUCT_PRICE` decimal(10,2) NOT NULL,
  `PRODUCT_STOCK` decimal(10,0) DEFAULT NULL,
  `PRODUCT_FILENAME` varchar(200) DEFAULT NULL,
  PRIMARY KEY (`PRODUCT_ID`)
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8;

/*Data for the table `product` */
```



```
package com.feng.entity;
/*
定义商品实体类
 */
public class Product {
    private  int PRODUCT_ID;//商品id
    private String PRODUCT_NAME;//商品名称
    private String PRODUCT_DESCRIPTION;//商品描述
    private  int PRODUCT_PRICE;//商品价格
    private  int PRODUCT_STOCK;//商品库存
    private String PRODUCT_FILENAME;//商品图片地址
```


- 购物车信息表cart，对应实体类Cart

```
/*Table structure for table `cart` */

DROP TABLE IF EXISTS `cart`;

CREATE TABLE `cart` (
  `cart_id` int NOT NULL AUTO_INCREMENT,
  `cart_p_filename` varchar(128) DEFAULT NULL,
  `cart_p_name` varchar(64) DEFAULT NULL,
  `cart_p_price` decimal(10,2) DEFAULT NULL,
  `cart_nums` int DEFAULT NULL,
  `cart_p_stock` int DEFAULT NULL,
```

```
  `cart_p_id` int DEFAULT NULL,
  `cart_u_id` varchar(64) DEFAULT NULL,
  `cart_valid` int DEFAULT NULL,
  PRIMARY KEY (`cart_id`)
) ENGINE=InnoDB AUTO_INCREMENT=18 DEFAULT CHARSET=utf8;

/*Data for the table `cart` */
```



```java
package com.feng.entity;

/**
 * 定义一个购物车实体类
 */
public class Cart {
    private int cart_id;//购物车id
    private String cart_p_filename;//购物车中的商品图片地址
    private String cart_p_name;//购物车中的商品名称
    private int cart_p_price;//购物车商品价格
    private int cart_nums;//购物车商品数量
    private int cart_p_stock;//购物车商品库存
    private int cart_p_id;//购物车商品id
    private String cart_u_id;//购物车用户id
    private int cart_valid;//购物车状态判断
```


- 设计用户注册登录页面（以及修改个人信息）
  - 不建议使用过多的JSP来编写（如servlet，JDBC连接数据库等），尽管老师要求使用。
  - 把准备好的静态前端页面.html修改为.JSP动态，我仅仅是用来动态交互信息，==servlet和数据库操作都用java来写，虽然代码比JSP多了点，但是有IDEA根本不用害怕。==个人觉得用servlet来写比JSP思路清晰很多。
  - 言归正传，注册页面，用JS脚本，正则表达式来判断用户注册信息合法性，用ajax局部刷新来判断注册用户ID是否已经被注册过
  - 主要代码如下：注册页面function.js，后端表单发送数据servlet，数据库操作就不看了吧

```
function  FocusItem(obj) {
    var focus=$(obj).next('span');
    if (focus.attr('class')=='error'){
        obj.value="";
    }
    focus.html("");
    focus.removeClass("error");
}
function BlueItem(obj) {
    var msgBox=$(obj).next('span');
    switch ($(obj).attr('name')) {
        case "userName":
            if (obj.value==""){
                msgBox.addClass('error');
                msgBox.html('用户名不能为空');
            }else{
                var url="userIdCheck?useName="+encodeURI($(obj).val())+"&"+new Date().getTime();
                $.get(url,function (data) {
                    if (data>0){
                        msgBox.addClass('error');
                        msgBox.html("用户名已存在,请登录");
                    }
                })
            }
            break;
        case "name":
            if (obj.value==""){
                msgBox.addClass('error');
                msgBox.html('姓名不能为空');
            }
            break;
        case "passWord":
            if (obj.value==""){
                msgBox.addClass('error');
                msgBox.html('密码不能为空');
            }
            break;
        case "rePassWord":
            if (obj.value==""){
                msgBox.addClass('error');
                msgBox.html('确认密码不能为空');
            }else if (obj.value!=$("input[name='passWord']").val()){
                msgBox.addClass('error');
                msgBox.html('两次输入的密码不一致');
            }
            break;
    }
}
function CheckFrom(obj) {
    var ress=$("input[name='userName']").next('span').attr("class");
    var res=$("input[name='rePassWord']").next('span').attr("class");
    if (res=="error"||ress=="error"){
        return false;
    }
    return true;
}
```

- JDBC封装类，后面一堆操作数据库，这里吧JDBC操作简单封装一下可以简化代码很多
  代码如下：

```
package com.feng.util;

import java.sql.*;

/**
 * 封装
 * Jdbc工具类，简化Jdbc编程
 */
public class JdbcUtil {
    final String URL="jdbc:mysql://localhost:3306/shop2?useSSL=true&serverTimezone=GMT";
    final String USERNAME="root";
    final String PASSWORD="333";
    Connection conn=null;
    PreparedStatement ps=null;

    //将jar包中的driver实现类加载到JVM中
    //静态代码块在类加载时执行，并且只执行一次
    static {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }
    /**
     * 获取数据库连接对象
     * @return 连接对象Conn
     */
    private  Connection getConn(){
        try {
            conn= DriverManager.getConnection(URL,USERNAME,PASSWORD);
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
        return conn;
    }
    /**
     * 获取数据库操作对象
     * @param
     * @return ps
     */
    public  PreparedStatement getPs(String sql){
        try {
            ps=getConn().prepareStatement(sql);
        } catch (SQLException throwables) {
            throwables.printStackTrace();
        }
        return ps;
    }
    /**
     * 关闭资源
     */
    public  void close(){
        if (ps != null) {
            try {
                ps.close();
```

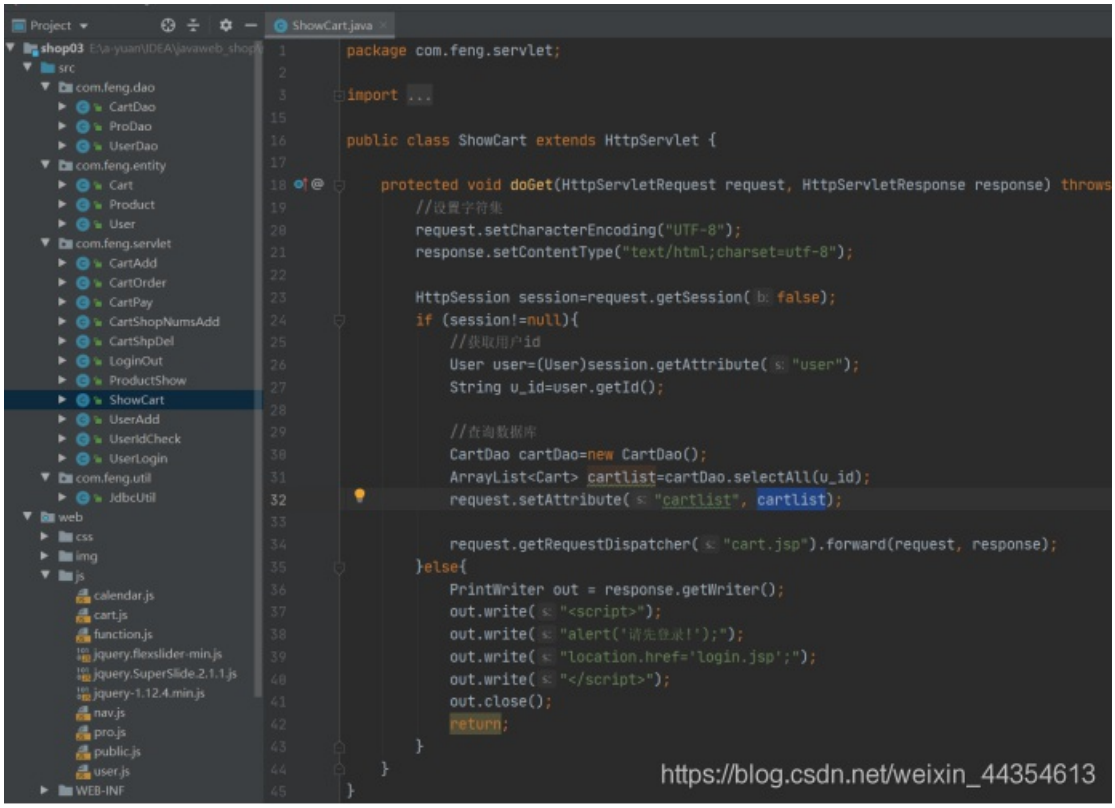
```
            } catch (SQLException throwables) {
                throwables.printStackTrace();
            }
        }
        if (conn != null) {
            try {
                conn.close();
            } catch (SQLException throwables) {
                throwables.printStackTrace();
            }
        }
    }
    public void close(PreparedStatement ps) {
        if (ps != null) {
            try {
                ps.close();
            } catch (SQLException throwables) {
                throwables.printStackTrace();
            }
        }
    }
    public void close(PreparedStatement ps, ResultSet rs){
        if (ps != null) {
            try {
                ps.close();
            } catch (SQLException throwables) {
                throwables.printStackTrace();
            }
        }
        if (rs != null) {
            try {
                rs.close();
            } catch (SQLException throwables) {
                throwables.printStackTrace();
            }
        }
    }
}
```

- 商品详情页面设计（那么多商品总不能每个商品设计一个商品详情页面吧）
- 所以前面设计商品信息类在这里就可以等到很好的使用，只需要给同一个商品详情页附带商品id，从数据库拿出各自商品详细信息展示到页面就可以

```
<div class="news">
    <div class="wrapper"><h2></h2>
        <h2><img src="img/ih2.jpg"/></h2>
        <div class="flower clearfix tran"><a href="productshow?id=1" class="clearfix">
            <dl>
                <dt><span class="abl"></span><img src="img/flo1.jpg"/><span class="abr"></span></dt>
                <dd>【花艺】七头美丽玫瑰仿真花束</dd>
                <dd><span>¥ 158.00</span></dd>
            </dl>
        </a><a href="productshow?id=2">
            <dl>
                <dt><span class="abl"></span><img src="img/flo2.jpg"/><span class="abr"></span></dt>
                <dd>【花艺】七头美丽玫瑰仿真花束2</dd>
                <dd><span>¥ 51.00</span></dd>
            </dl>
        </a><a href="productshow?id=3">
            <dl>
                <dt><span class="abl"></span><img src="img/flo3.jpg"/><span class="abr"></span></dt>
                <dd>【花艺】七头美丽玫瑰仿真花束3</dd>
```

```
            <dd><span>¥ 104.00</span></dd>
        </dl>
    </a></div>
    <div class="flower clearfix tran"><a href="productshow?id=4">
        <dl>
            <dt><span class="abl"></span><img src="img/flo4.jpg"/><span class="abr"></span></dt>
            <dd>【花艺】七头美丽玫瑰仿真花束4</dd>
            <dd><span>¥ 79.00</span></dd>
        </dl>
    </a><a href="productshow?id=5">
        <dl>
            <dt><span class="abl"></span><img src="img/flo5.jpg"/><span class="abr"></span></dt>
            <dd>【花艺】七头美丽玫瑰仿真花束5</dd>
            <dd><span>¥ 151.00</span></dd>
        </dl>
    </a><a href="productshow?id=6">
        <dl>
            <dt><span class="abl"></span><img src="img/flo6.jpg"/><span class="abr"></span></dt>
            <dd>【花艺】七头美丽玫瑰仿真花束6</dd>
            <dd><span>¥ 164.00</span></dd>
        </dl>
    </a></div>
</div>
```


- 购物车设计
- 从购物车数据库中读取用户所有购物车信息存放在ArrayList cartlist



- 然后是使用JSP标准标签库<c:forEach>，来遍历传回来的cartlist数组，并展示成购物车页面

```
cart.jsp
    <a href="index_2.jsp." class="fr">继续购物></a></p></div><!----------------table----------------
<div class="table wrapper">
    <div class="tr">
        <div>商品</div>
        <div>单价</div>
        <div>数量</div>
        <div>小计</div>
        <div>操作</div>
    </div>
    <c:forEach var="rs" items="${requestScope.cartlist }">
        <div class="th">
```

- 还有购物车商品删除，增加、减少数量，商品结算，模拟支付，支付的同事展示用户地址，电话等等，就不贴不来了，完整代码另外打包压缩。

小结

首先，这个项目一个人来做工作量很大（要考虑很多东西），因为我最开做登录注册在一部分慢慢精挑细作，仔细到每个输入框都判断一次，然后写完这一块我发现我已经用了很多很多时间。导致后面购物车没有多少时间来完成。

工作量大，但是实际写代码还是比较简单的，比如很多servlet用到的都是差不多的代码，数据库查询也是这样。总结就是技术不难，不过对整个网页要考虑周全，却很难，也很费时间。（比如我在大屏幕显示器注册页面可以展示，但是换到我的笔记本，小屏幕下注册页面就显示不全，界面不够友好。）等等很多类似问题，时间花在这些细小修改上的才是最多的。

一个小体会、小总结：整体项目主要功能代码书写并不难，难的是不断修改页面，修改代码这一块，也是最耗时间的。

ajxa的局部刷新很好用，例如我在写注册信息判断用户注册id是否合法的时候。代码书写的时候很多地方我是用JQuery来写的，代码因此简洁了很多。

这项目还有一个难点就是前期准备工作，对数据库表的创建和规划。我在用户表和购物车表的基础上，用了一个商品信息表使总体思路清晰了很多。

时间有限，还有很多可以优化的地方都没有时间写，简单记录一下目前我想到的可以优化的地方：

- 1、功能不全，缺少商家后台管理系统（这一步我在写用户表最后一列就是表示用户状态管理员或者普通用户，但是没有时间来实现这一块）
- 2、用户支付后的订单跟踪没有做（比如发货中，包裹签收，评价等）
- 3、没有写过滤器（应该用过滤器来保护一些web中的资源不被随便访问，就是大部分资源需要用户登录后才可以访问）我在实用中用session判断了几个重要资源，但还是不够安全（比如购物车页面需要用户登录后才可以访问）
- 4、没有设计数据库连接池，（如果设计了数据库连接池应该可以在时间上优化很多）进而给用户带来好的体验。

项目完整代码及数据库.SQL文件

链接: gitee仓库链接.

自学网站

> https://www.bilibili.com/video/BV1zE411Y7Mg
> https://www.bilibili.com/read/cv6395168?spm_id_from=333.788.b_636f6d6d656e74.12

2021.05.25补充

> 这个项目写了很久，当时是我刚学了前端和Java后做的一个实验。现在看来我就写了这里面几个接口的后端代码，而且其中的业务逻辑写的很乱也有很多不太合理的逻辑。
> 使用的技术都是原生的，没有用框架，所有整体代码很乱，也很多冗余代码。但是也用了一些包装类。
> 最后，这个项目纯粹就是练练手，将学习的技术应用到项目中，体验一下业务接口的开发，能跑就ok了。真的存在很多问题。
> 总结：虽然存在很多问题，但是这个实验也是给了我很多学习的动力，和对Java的兴趣（好玩）。然后就是用原生技术来写真的可以锻炼你对底层技术的掌握和理解，不像现在大家都用框架来开发，很多底层都已经封装好了。当时完成这个项目对我这个Java初学者帮助也挺大的。大家加油！！！