

# 基于 SpringBoot 的 B2C 网上商城

## ——第十组数据库课程项目设计

小组成员姓名	学号	分工	工作量占比
蒋光启	2020141450013	项目架构设计, 负责用户管理、订单两大功能模块的后端设计, 参与商品购物车功能的实现和前端页面设计调试	40%
范勇奇	2020141450109	前端页面设计, UI 设计, 参与 API 设计, 负责数据库设计, 参与商品和购物车管理模块的功能开发	30%
李沛钊	2020141470307	参与订单、商品购物车处理功能的实现, 负责设计表单, 设计前端页面, 负责收货地址模块功能的实现	30%

完成日期: 2022 年 6 月 15 日

# 一、选题背景

## 1.网上商城业务背景

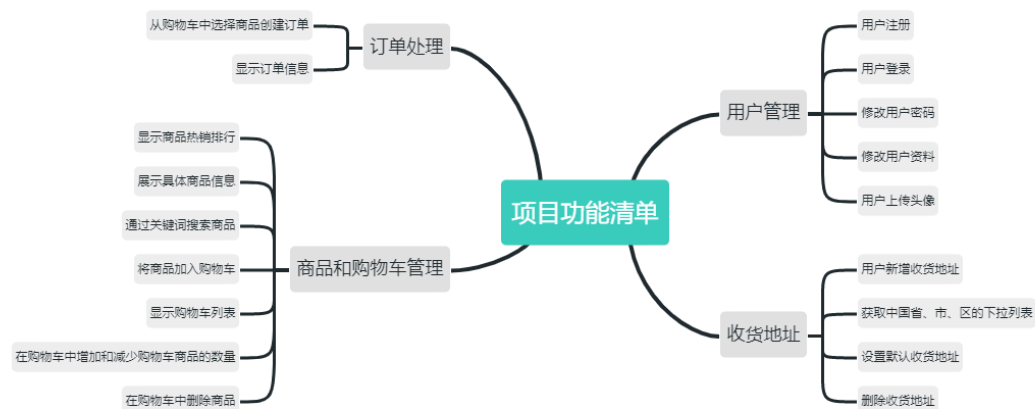
在 21 世纪，随着互联网的发展，电商迅速发展，截止 2021 年 6 月，我国网络购物用户规模达 8.12 亿人，同时，2020 年全国电子商务年交易额达 37.21 万亿元人民币。淘宝、京东、美团等电商平台在人们的生活的方方面面都有涉足，甚至路边的小餐馆支持的扫码点餐本质上也是电商，电商可谓是无处不及。

## 2.数据库课程相关性

在电商中，数据库的应用十分广泛：用户信息的储存(用户登录信息，头像、地址等个人资料，网站上的商品信息，购物车商品列表，用户的订单信息，商品的销售排行榜的更新等，无一不用到数据库。

## 3.欲实现的主要功能

一个完整的网上商城项目功能过于繁杂，考虑到时间和团队规模的限制，我们小组挑取了最核心的功能进行实现，具体功能见如下脑图：



## 二、技术路线

### 1.主要技术

- (1) 数据库：MySQL + navicat 可视化
- (2) 服务器架构：Spring Boot 2.4.7 + MyBatis 2.1.4 + AJAX
- (3) 前端页面：bootstrap 渲染前端页面的整体风格，采用静态 HTML 页面结合 JavaScript 脚本的方式显示页面和处理请求，采用 AJAX 与后端进行通信交互

### 2.开发平台

- (1) 操作系统：Windows 10
- (2) Java 开发包：JDK 8+
- (3) 项目管理工具：Maven 3.6.3 管理项目所需的依赖项
- (4) 项目开发工具：IntelliJ IDEA Ultimate 2020.3.2 x64
- (5) 浏览器：Google Chrome, Edge etc.

### 3.关键技术及难点

- (1) 前端和后端通信采用了异步通信技术 AJAX，一个典型的 AJAX 函数如下

```
$.ajax({
  url: "/products/hot_list",
  type: "GET",
  dataType: "JSON",
  success: function(json) {
    let list = json.data;
    console.log("count=" + list.length);
    for (let i = 0; i < list.length; i++) {
      console.log(list[i].title);
      let html = '<div class="col-md-12">'
        + '<div class="col-md-7 text-row-2"><a href="product.html'
        + '<div class="col-md-2">¥#{price}</div>'
        + '<div class="col-md-3">';

      html = html.replace(/#{id}/g, list[i].id);
      html = html.replace(/#{title}/g, list[i].title);
      html = html.replace(/#{price}/g, list[i].price);
      html = html.replace(/#{image}/g, list[i].image);

      $("#hot-list").append(html);
    }
  }
});
```

- (2) 密码采用了 MD5 加密技术，复用现有的加密算法，核心代码如下：

```

// 补全数据：加密后的密码
// 密码加密处理的实现：md5算法的形式：1roifjujasdfi-afdjhlj-ajflds-qrttg
// (串 + password + 串) ---- md5算法进行加密，连续加载3次
// 盐值 + password + ---- 盐值就是一个随机的字符串
String oldpassword = user.getPassword();
// 获取盐值，随机生成一个
String salt = UUID.randomUUID().toString().toUpperCase();
// 将密码和盐值作为一个整体进行加密处理，忽略了原有密码的强度
String md5Password = getMd5Password(oldpassword, salt);
// 将加密之后的密码重新补全设置到user对象中
user.setPassword(md5Password);

// 补全数据：盐值
user.setSalt(salt);

```

```

// 从查询结果中获取盐值
String salt = result.getSalt();
// 调用getMd5Password()方法，将参数password和salt结合起来进行加密
String md5Password = getMd5Password(password, salt);
// 判断查询结果中的密码，与以上加密得到的密码是否不一致
// 是：抛出PasswordNotMatchException异常
if (!result.getPassword().equals(md5Password))
{
    throw new PasswordNotMatchException("用户密码错误");
}

```

### (3) 用户头像保存的 cookie 技术

```

success: function(json) { //这里function参数json经常容易漏掉
    if (json.state == 200) {
        alert("登陆成功!");
        //使用相对路径跳转到主页index.html
        location.href = "index.html";
        // 将服务器返回的头像设置到cookie中 (cookie需要导入jquery文件)
        // key, value, time(unit: day)
        $.cookie("avatar", json.data.avatar, {expires: 7});
    } else {
        alert("登陆失败! " + json.message);
    }
},
error: function (xhr) {
    alert("登陆时产生未知的异常 " + xhr.message);
}
}

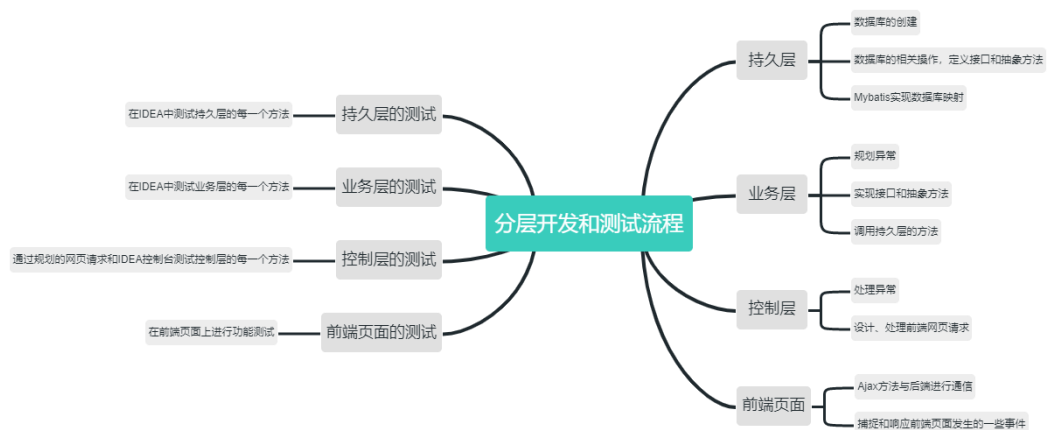
```

(4) 实现购物车采用了 HttpSession 技术

```
/**
 * 获取session对象中的uid
 * @param session session对象
 * @return 当前登录用户的uid
 */
17 个用法  Lucca
protected final Integer getUserIdFromSession(HttpSession session)
{
    return Integer.valueOf(session.getAttribute(s: "uid").toString());
}

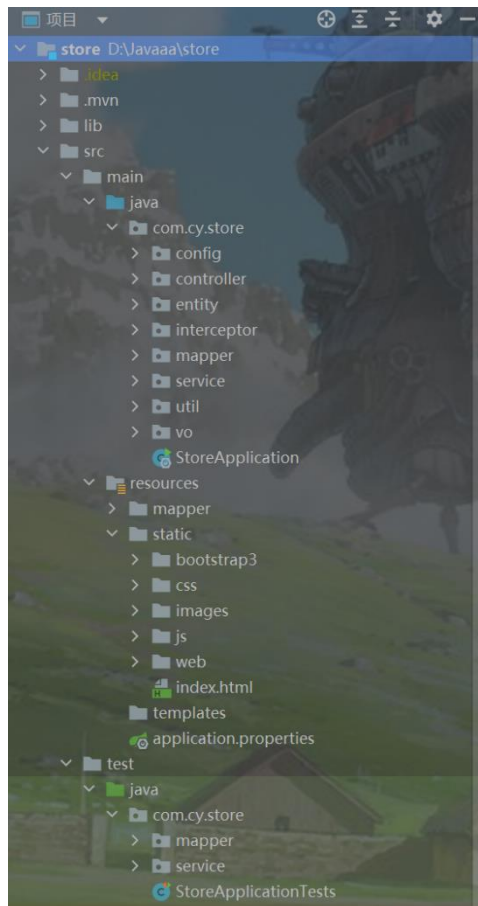
/**
 * 获取当前登录用户的username
 * @param session session对象
 * @return 当前登录用户的用户名
 *
 * 在实现类中重写了父类的toString方法? 可能不是句柄信息输出
 */
11 个用法  Lucca
protected final String getUsernameFromSession(HttpSession session)
{
    return session.getAttribute(s: "username").toString();
}
```

(5) 通过分层开发使得各模块符合“高内聚低耦合”等设计策略，同时方便开展单元测试、集成测试



(6) 项目目录结构

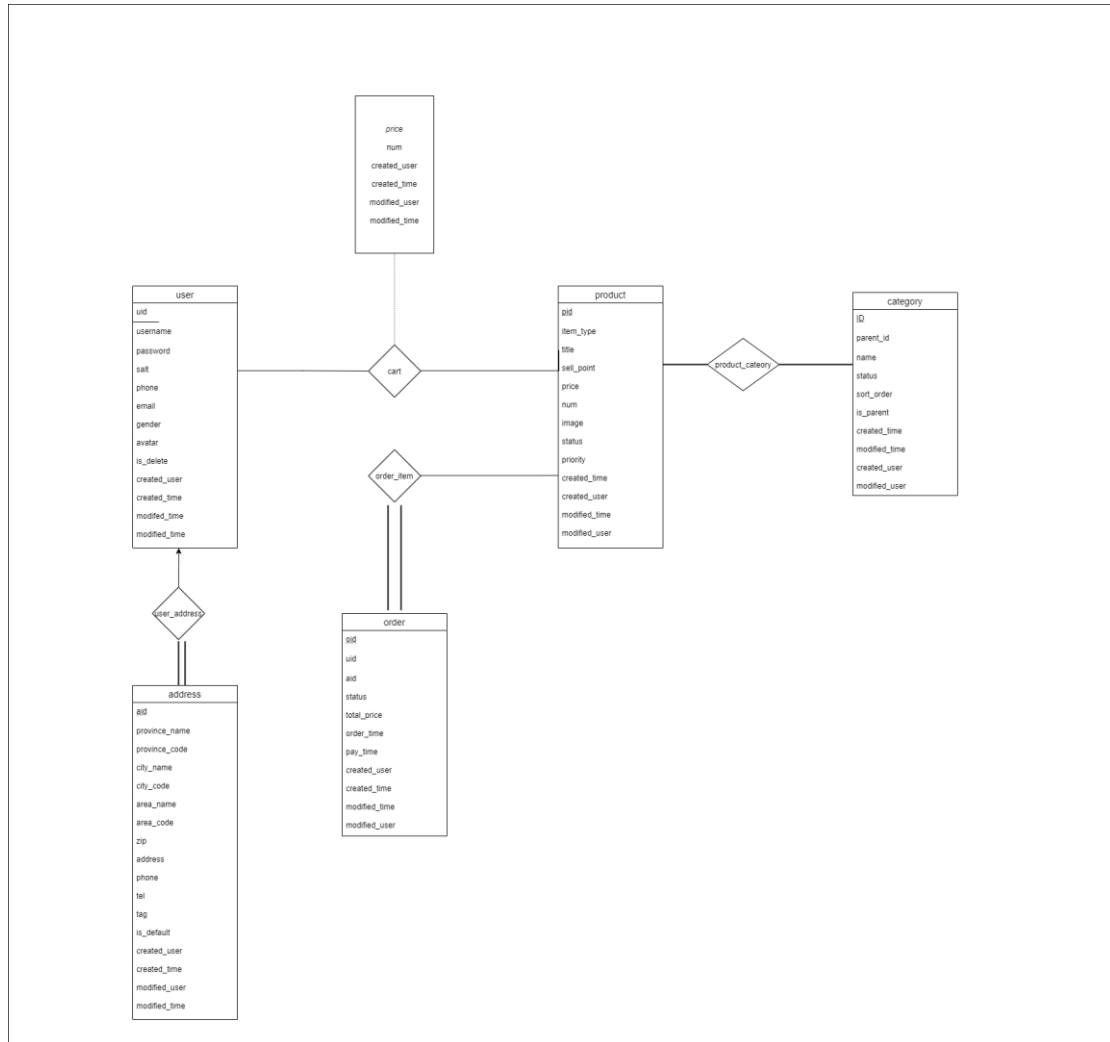




### 三、概要设计

#### 1.数据库设计

以下是数据库的 E-R 图:



其中:

user 实体集来表示所有用户的集合

product 实体集表示所有商品的集合

order 实体集表示所有订单的集合

category 实体集表示所有分类的集合

联系集 cart 表示购物车, 它关联了用户(user)和商品(product)

联系集 user\_address 表示用户地址, 它关联了用户(user)和地址(address)

联系集 product\_category 表示商品的分类, 它关联了商品(product)和分类(category)

联系集 order\_item 表示订单中的商品, 它关联了订单(order)和商品(product)

属性具体的含义将在详细设计中介绍

## 2.功能设计

以用例图的方式展现以下功能设计





## 四、详细设计

### 1.数据库设计

#### 1.1 逻辑设计:

由 E-R 图我们可以得到以下关系模式

user(uid, username, password, salt, phone, email, gender, avatar, is\_delete, created\_user, created\_time, modified\_user, modified\_time)

product(pid, item\_type, title, sell\_point, price, num, image, status, priority, created\_time, created\_user, modified\_time, modified\_user)

category(ID, parent\_id, name, status, sort\_order, is\_parent, created\_time, modified\_time, created\_user, modified\_user)

address(aid, province\_name, province\_code, city\_name, city\_code, area\_name, area\_code, zip, address, phone, tel, tag, is\_default, created\_user, created\_time, modified\_user, modified\_time)

order(oid, uid, aid, status, total\_price, pay\_time, created\_time, create\_user, modified\_time, modified\_user)

cart(uid, product, price, num, created\_time, created\_user, modified\_time, modified\_user)  
user\_address(uid, aid)

order\_item(oid, pid)

Product\_category(pid, ID)

#### 1.2 物理设计:

user 表

```
CREATE TABLE user (  
  uid INT AUTO_INCREMENT COMMENT '用户 id',  
  username VARCHAR(20) NOT NULL UNIQUE COMMENT '用户名',  
  password CHAR(32) NOT NULL COMMENT '密码',  
  salt CHAR(36) COMMENT '盐值',  
  phone VARCHAR(20) COMMENT '电话号码',  
  email VARCHAR(30) COMMENT '电子邮箱',  
  gender INT COMMENT '性别:0-女, 1-男',  
  avatar VARCHAR(50) COMMENT '头像',  
  is_delete INT COMMENT '是否删除: 0-未删除, 1-已删除',  
  created_user VARCHAR(20) COMMENT '日志-创建人',  
  created_time DATETIME COMMENT '日志-创建时间',  
  modified_user VARCHAR(20) COMMENT '日志-最后修改执行人',  
  modified_time DATETIME COMMENT '日志-最后修改时间',  
  PRIMARY KEY (uid)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

product 表:

```
CREATE TABLE product (  
  id int(20) NOT NULL COMMENT '商品 id',
```

```

item_type varchar(100) DEFAULT NULL COMMENT '商品系列',
title varchar(100) DEFAULT NULL COMMENT '商品标题',
sell_point varchar(150) DEFAULT NULL COMMENT '商品卖点',
price bigint(20) DEFAULT NULL COMMENT '商品单价',
num int(10) DEFAULT NULL COMMENT '库存数量',
image varchar(500) DEFAULT NULL COMMENT '图片路径',
status int(1) DEFAULT '1' COMMENT '商品状态  1: 上架    2: 下架    3: 删除',
priority int(10) DEFAULT NULL COMMENT '显示优先级',
created_time datetime DEFAULT NULL COMMENT '创建时间',
modified_time datetime DEFAULT NULL COMMENT '最后修改时间',
created_user varchar(50) DEFAULT NULL COMMENT '创建人',
modified_user varchar(50) DEFAULT NULL COMMENT '最后修改人',
PRIMARY KEY (id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

category 表:

```

CREATE TABLE t_product_category (
  id int(20) NOT NULL COMMENT '主键',
  parent_id bigint(20) DEFAULT NULL COMMENT '父分类 id',
  name varchar(150) DEFAULT NULL COMMENT '名称',
  status int(1) DEFAULT '1' COMMENT '状态  1: 正常    0: 删除',
  sort_order int(4) DEFAULT NULL COMMENT '排序号',
  is_parent int(1) DEFAULT NULL COMMENT '是否是父分类  1: 是    0: 否',
  created_time datetime DEFAULT NULL COMMENT '创建时间',
  modified_time datetime DEFAULT NULL COMMENT '最后修改时间',
  created_user varchar(50) DEFAULT NULL COMMENT '创建人',
  modified_user varchar(50) DEFAULT NULL COMMENT '最后修改人',
  PRIMARY KEY (id)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

address 表:

```

CREATE TABLE address (
  aid INT AUTO_INCREMENT COMMENT '收货地址 id',
  name VARCHAR(20) COMMENT '收货人姓名',
  province_name VARCHAR(15) COMMENT '省-名称',
  province_code CHAR(6) COMMENT '省-行政代号',
  city_name VARCHAR(15) COMMENT '市-名称',
  city_code CHAR(6) COMMENT '市-行政代号',
  area_name VARCHAR(15) COMMENT '区-名称',
  area_code CHAR(6) COMMENT '区-行政代号',
  zip CHAR(6) COMMENT '邮政编码',
  address VARCHAR(50) COMMENT '详细地址',
  phone VARCHAR(20) COMMENT '手机',

```

```

tel VARCHAR(20) COMMENT '固话',
tag VARCHAR(6) COMMENT '标签',
is_default INT COMMENT '是否默认: 0-不默认, 1-默认',
created_user VARCHAR(20) COMMENT '创建人',
created_time DATETIME COMMENT '创建时间',
modified_user VARCHAR(20) COMMENT '修改人',
modified_time DATETIME COMMENT '修改时间',
PRIMARY KEY (aid)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

order 表:

```

CREATE TABLE order (
    oid INT AUTO_INCREMENT COMMENT '订单 id',
    uid INT NOT NULL COMMENT '用户 id',
    aid INT NOT NULL COMMENT '地址 id'

    status INT COMMENT '状态: 0-未支付, 1-已支付, 2-已取消, 3-已关闭, 4-已完成',
    order_time DATETIME COMMENT '下单时间',
    pay_time DATETIME COMMENT '支付时间',
    created_user VARCHAR(20) COMMENT '创建人',
    created_time DATETIME COMMENT '创建时间',
    modified_user VARCHAR(20) COMMENT '修改人',
    modified_time DATETIME COMMENT '修改时间',
    PRIMARY KEY (oid)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

cart 表:

```

CREATE TABLE cart (
    uid INT NOT NULL COMMENT '用户 id',
    pid INT NOT NULL COMMENT '商品 id',
    price BIGINT COMMENT '加入时商品单价',
    num INT COMMENT '商品数量',
    created_user VARCHAR(20) COMMENT '创建人',
    created_time DATETIME COMMENT '创建时间',
    modified_user VARCHAR(20) COMMENT '修改人',
    modified_time DATETIME COMMENT '修改时间',
    PRIMARY KEY (uid, pid)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

user\_address 表:

```

CREATE TABLE address (
    aid INT NOT NULL COMMENT '收货地址 id',

```

```
uid INT NOT NULL COMMENT '用户 id',
PRIMARY KEY (aid)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

order\_item 表:

```
CREATE TABLE order_item (
oid INT NOT NULL COMMENT '订单 id',
pid INT NOT NULL COMMENT '商品 id',
PRIMARY KEY (oid, pid)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

product\_category 表:

```
CREATE TABLE product_category (
pid INT NOT NULL COMMENT '商品 id',
ID INT NOT NULL COMMENT '分类 id',
PRIMARY KEY (pid, ID)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

## 2.功能和 UI 设计

使用 bootstrap 技术渲染页面，前端采用 JavaScript 脚本处理各种事件，部分静态网页和静态资源参考了别人设计的模板。

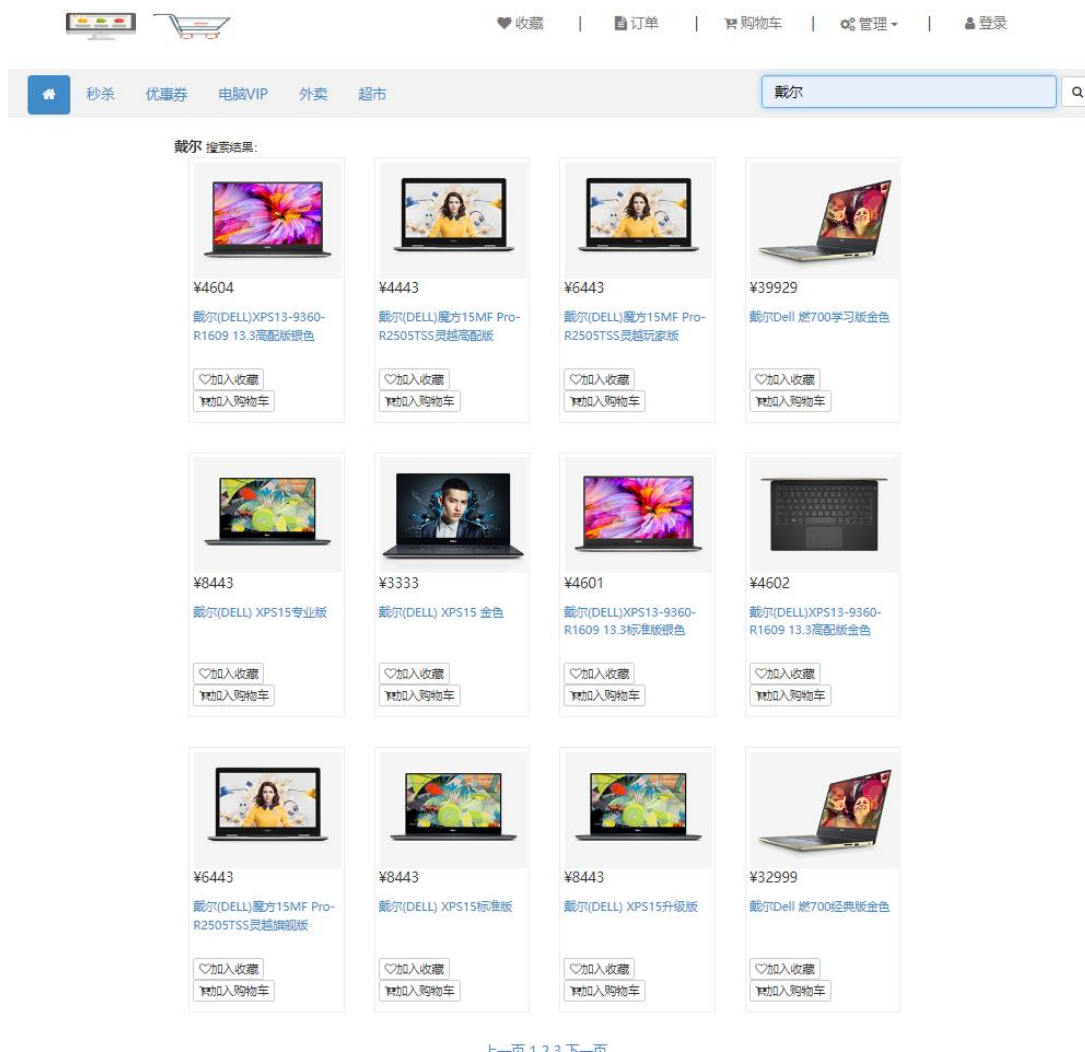
### 2.1 用户可以注册账号



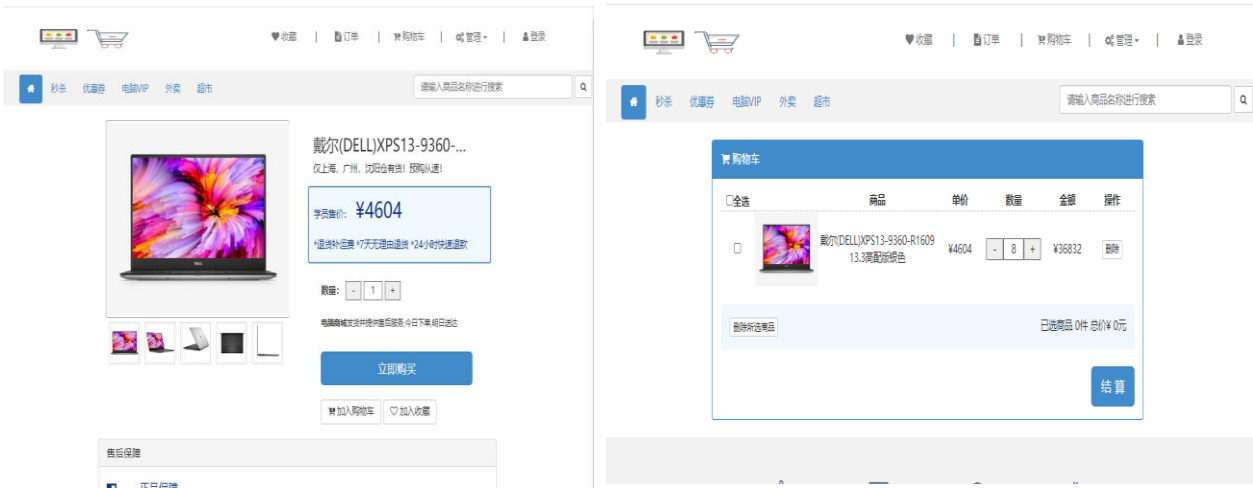
### 2.2 用户拥有账号后可以登录、修改资料、上传收货地址



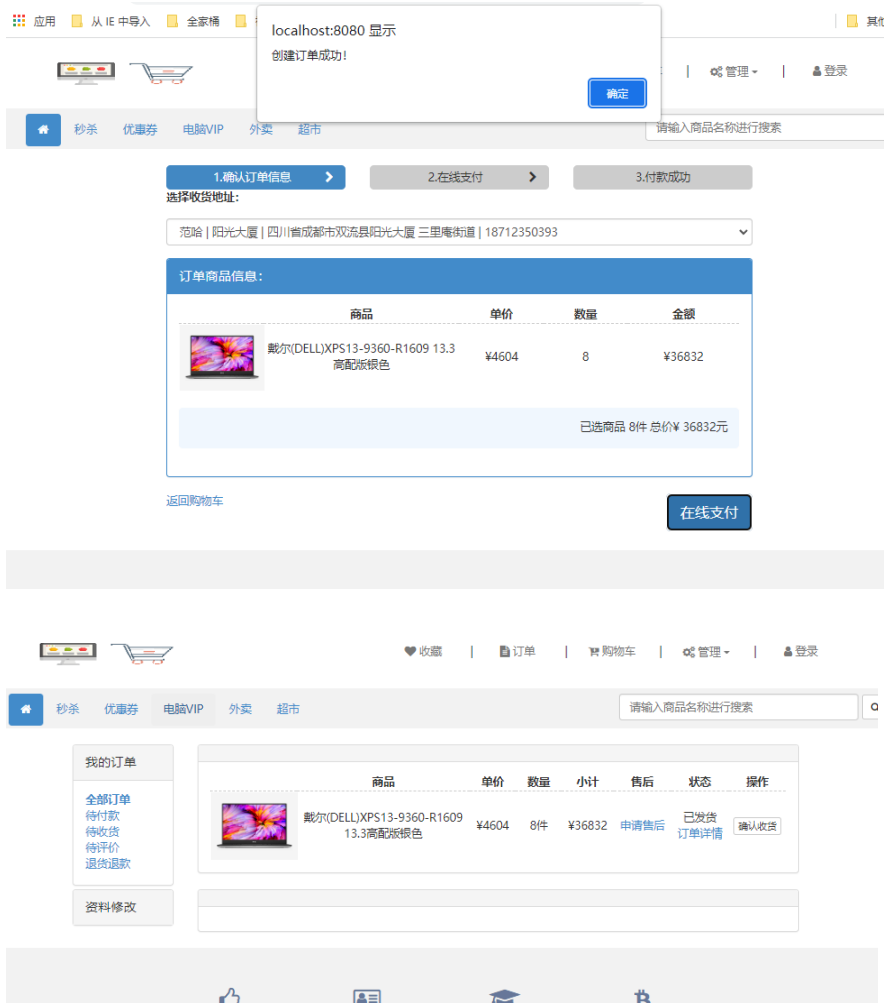
## 2.3 用户可以搜索商品



2.4 用户可以查看商品并加入购物车



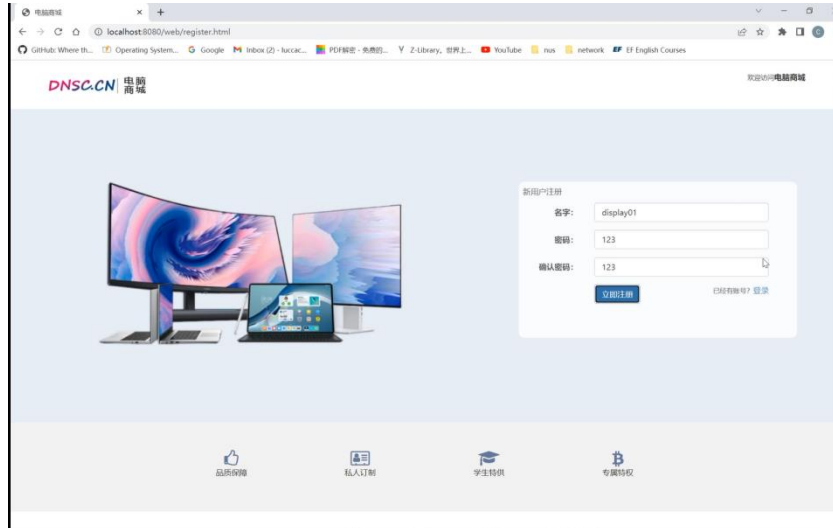
2.5 用户可以从购物车中结算创建订单、查看订单



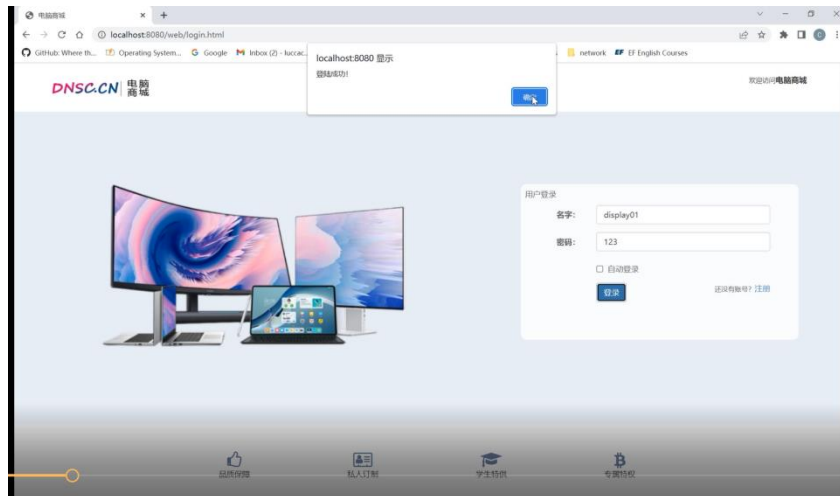
## 五、实现的主要功能

### 1、用户管理方面

(1) 用户注册: 输入用户名, 密码, 确认密码

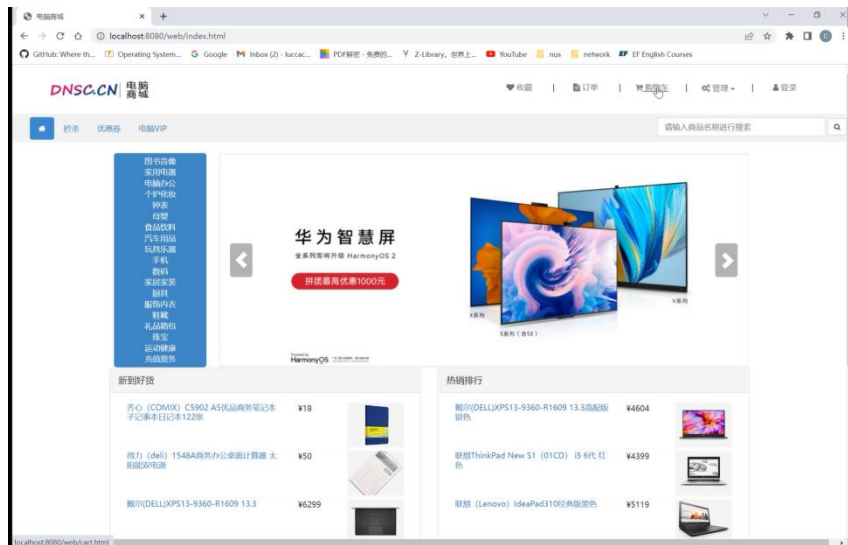


(2) 用户登录: 输入用户名和密码, 检查注册情况, 注册后进入平台主页面

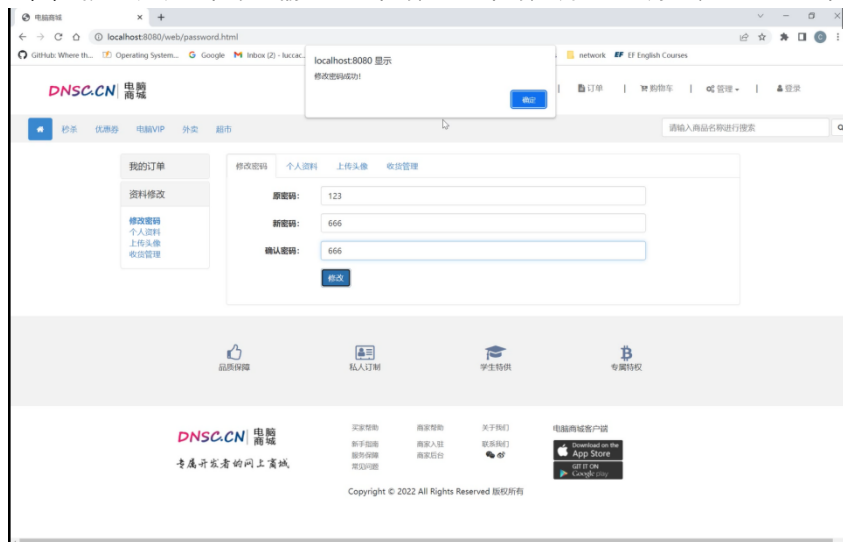


主页面如下:

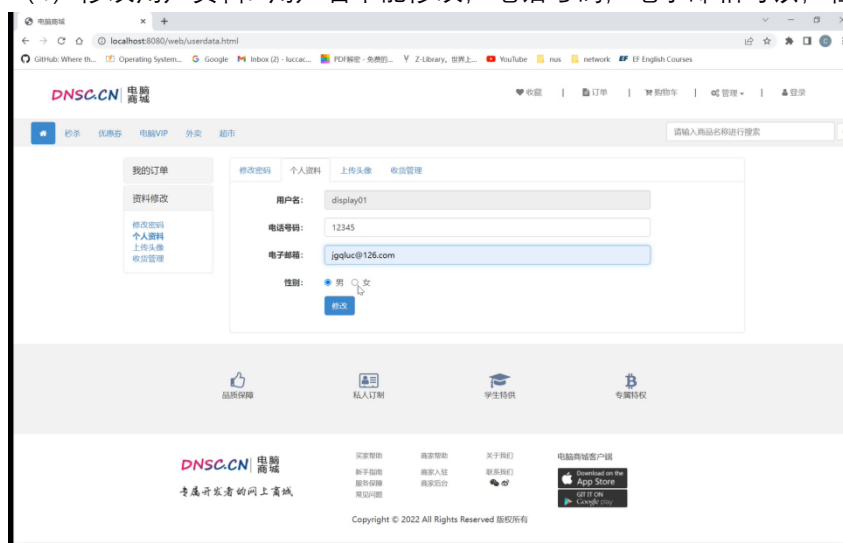




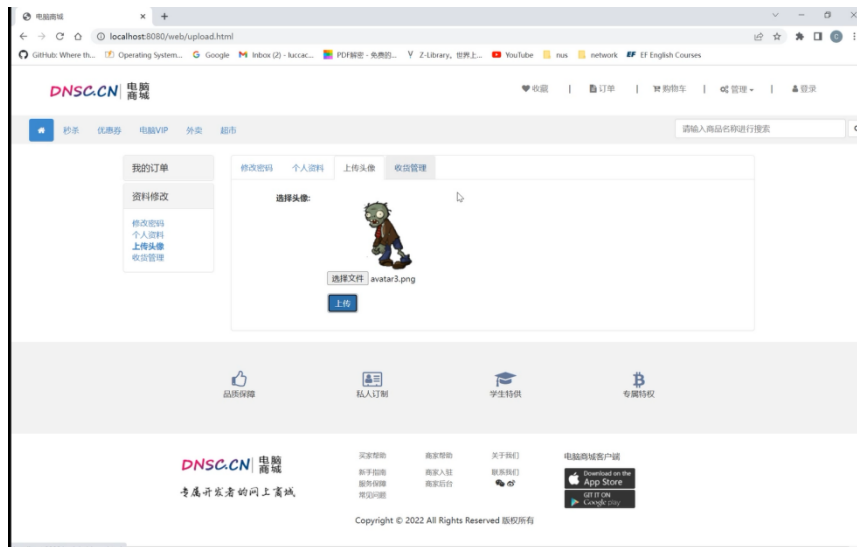
(3) 修改用户密码: 输入原密码, 新密码, 并进行确认, 如果原密码不对则提示失败



(4) 修改用户资料: 用户名不能修改, 电话号码, 电子邮箱可以, 性别可以选择男/女

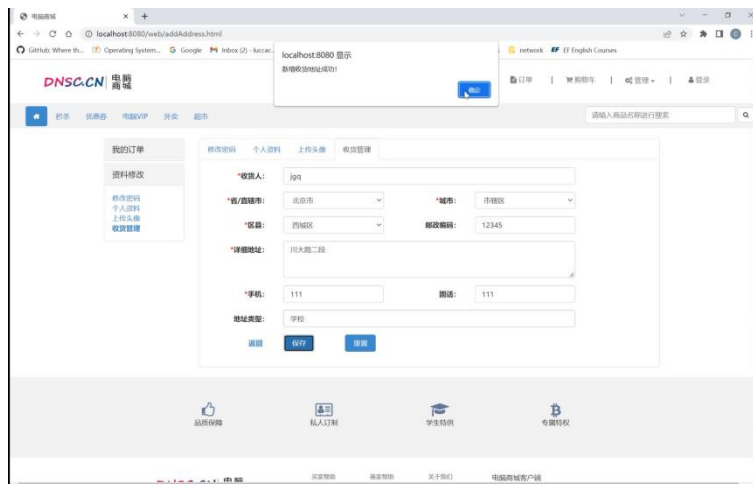


## (5) 用户上传头像:支持从本地选文件上传

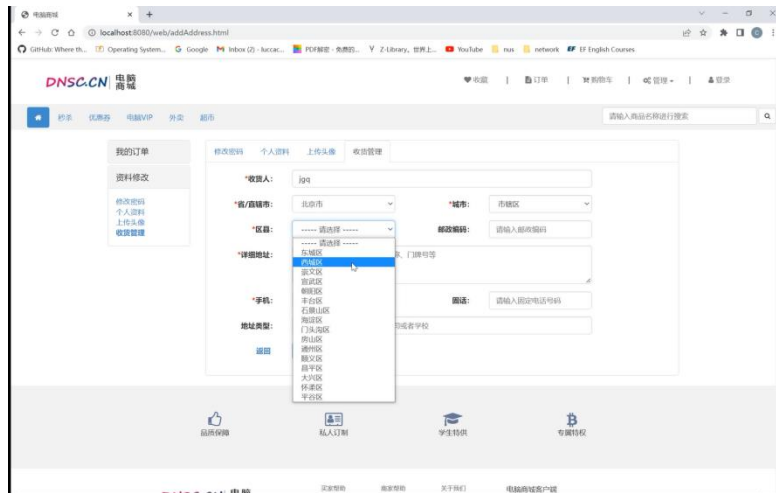


## 2、收货地址方面

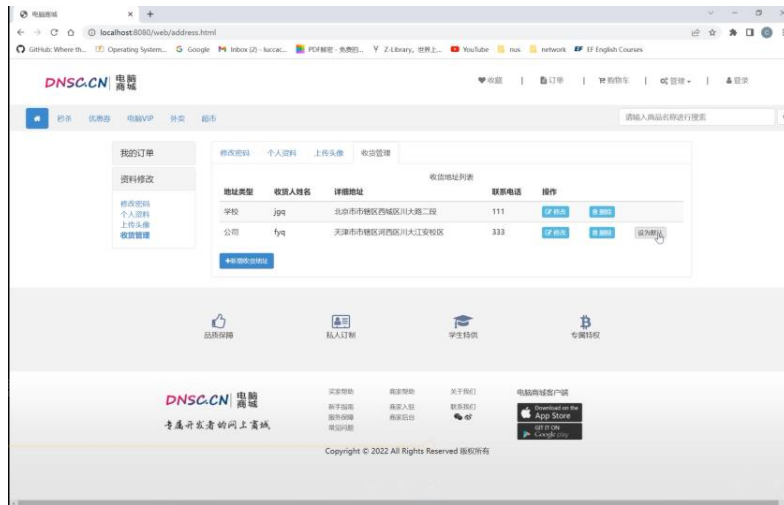
### (1) 用户新增收货地址: 填写收件人、地址、手机号码等内容



### (2) 获取中国省、市、区的下拉列表: (仅展示区/县)

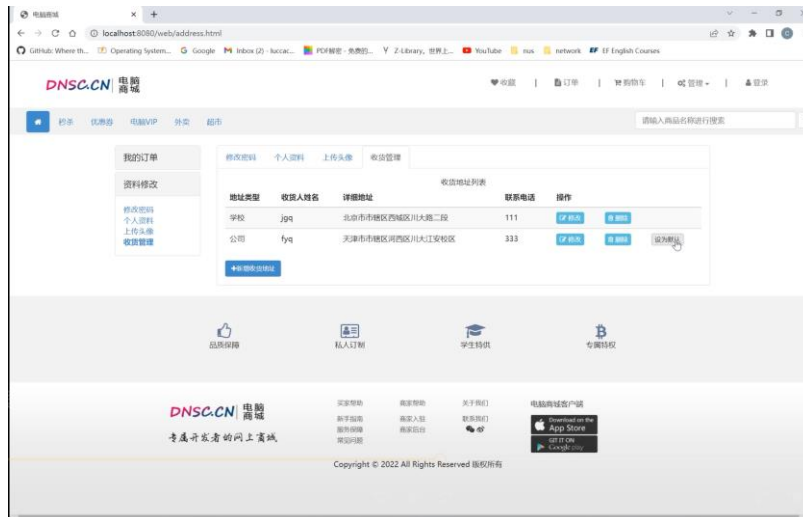


### (3) 设置默认收货地址：

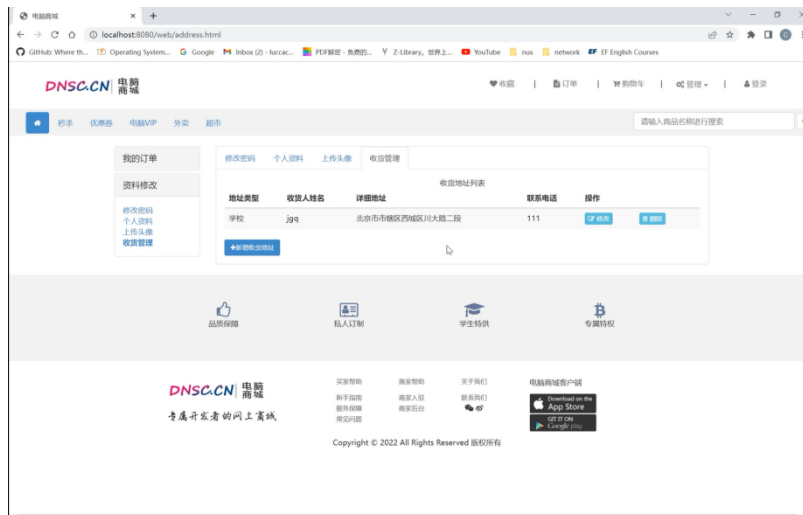


### (4) 删除收货地址：

删除之前：

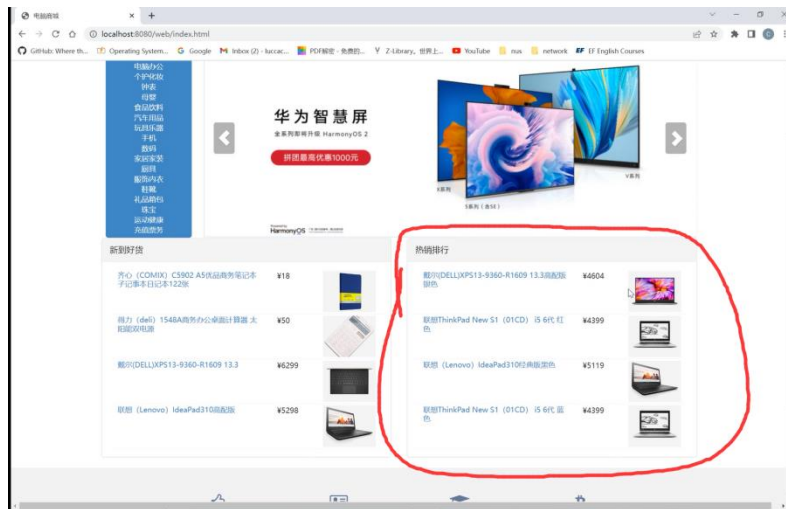


删除之后：

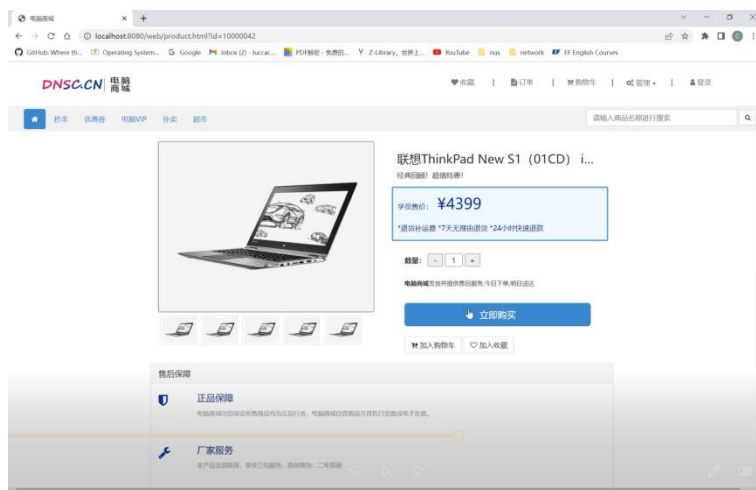


### 3、商品和购物车管理方面

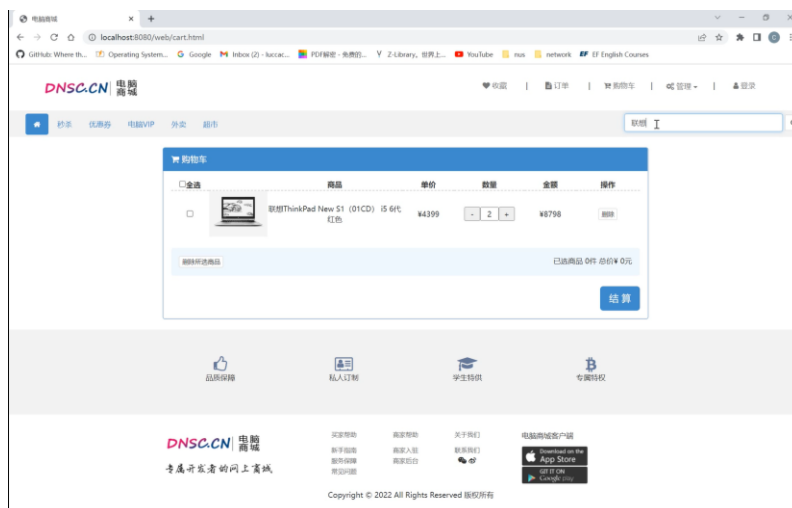
#### (1) 显示商品热销排行



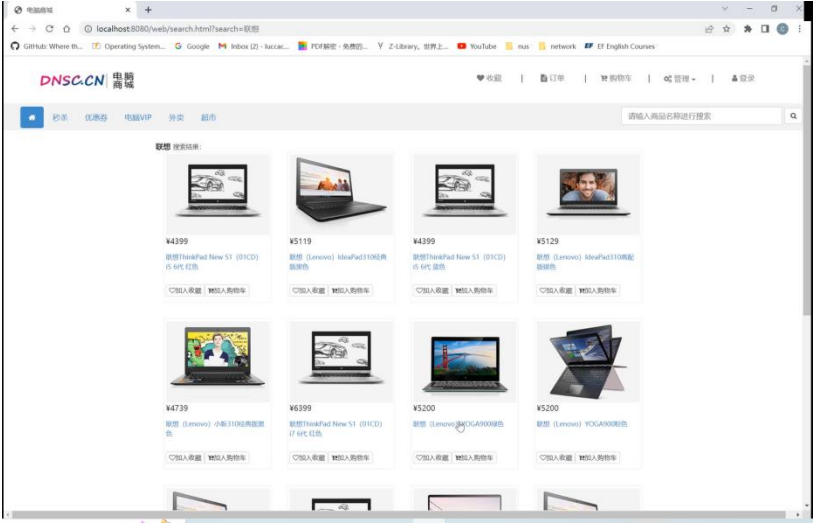
#### (2) 展示具体商品信息



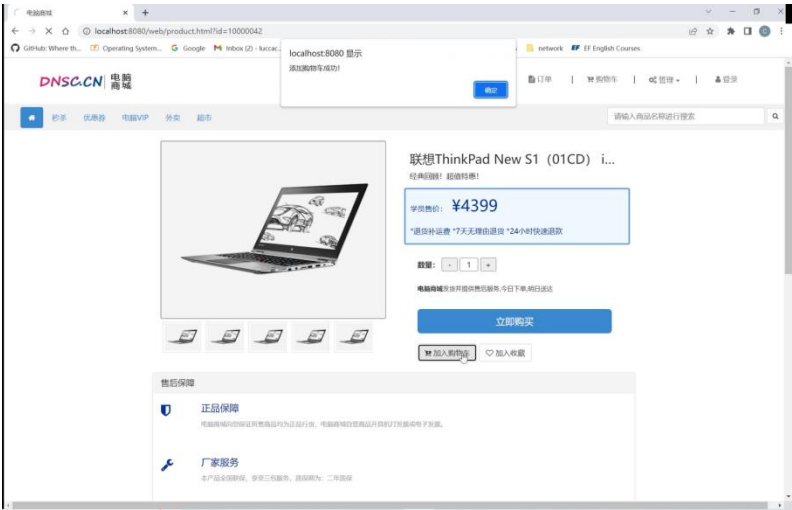
#### (3) 通过关键词搜索商品: 在右上角的搜索栏搜索商品, 返回搜索结果



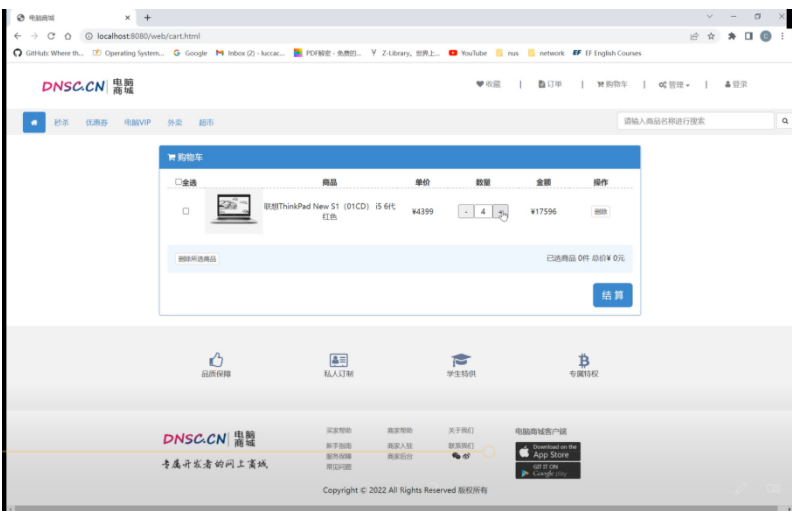
搜索“联想”返回的结果如下



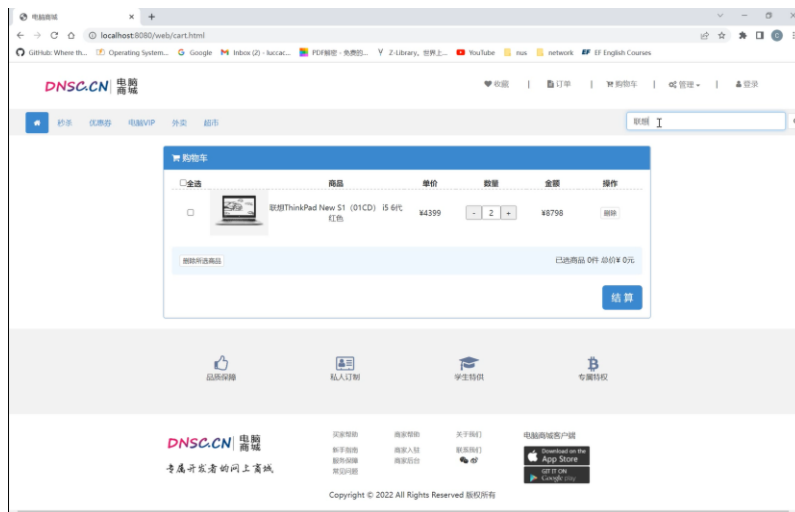
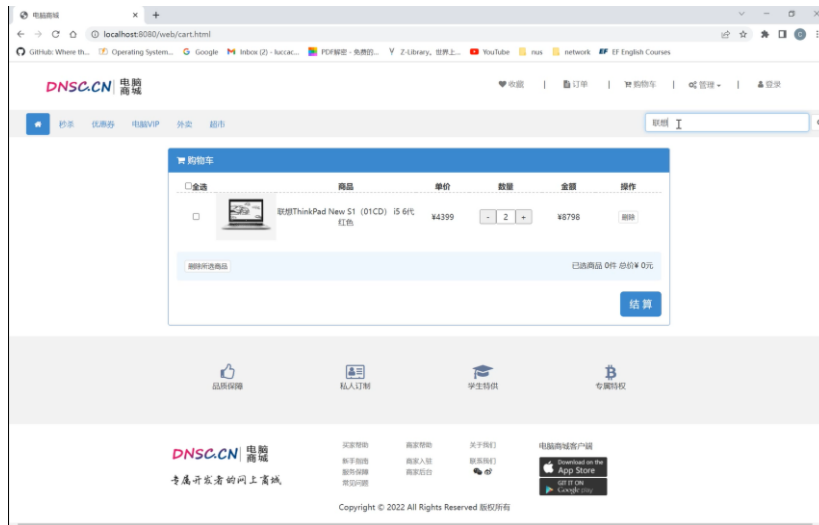
(4) 将商品加入购物车：在搜索结果页面和商品详细信息页面支持将商品加入到购物车



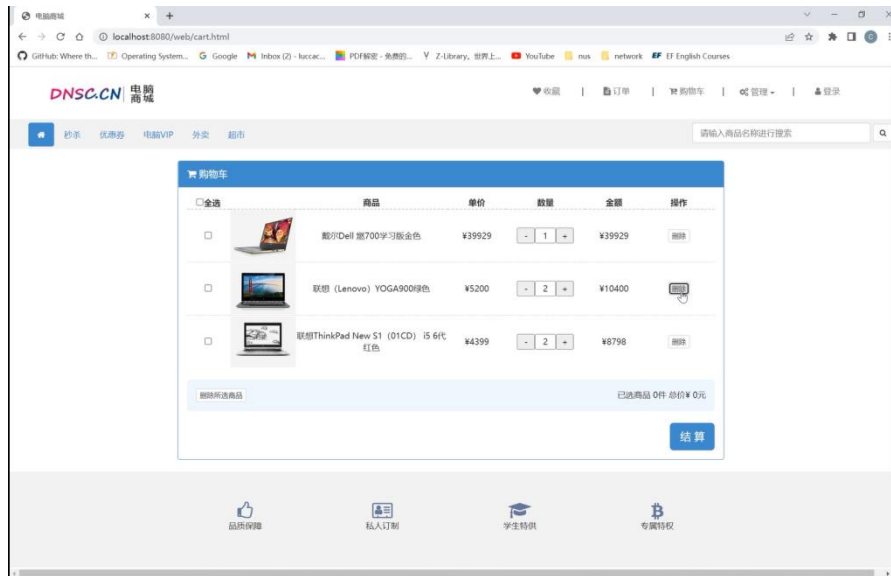
(5) 显示购物车列表



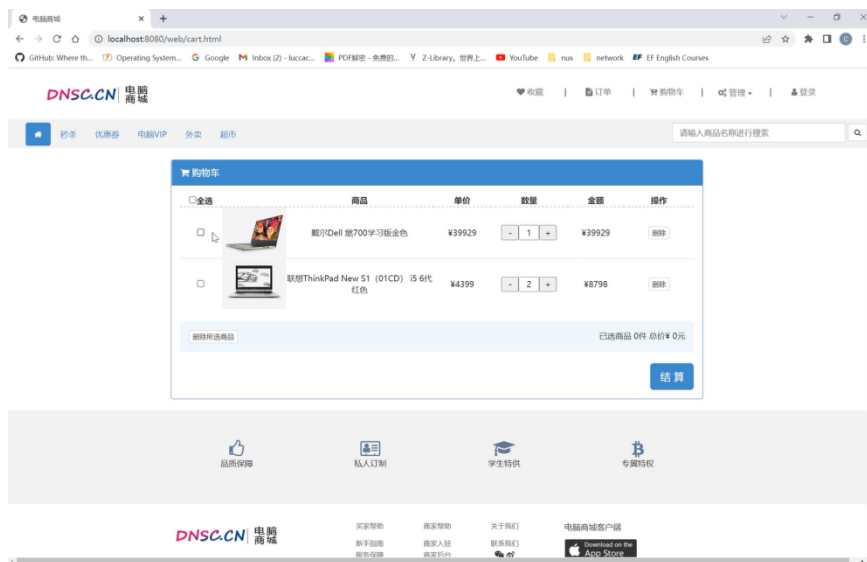
(6) 在购物车中增加和减少购物车商品的数量（示例 1->4->2）



(7) 在购物车中删除商品：删除购物车中商品，示例删除联想 YOGA9000  
删除前：



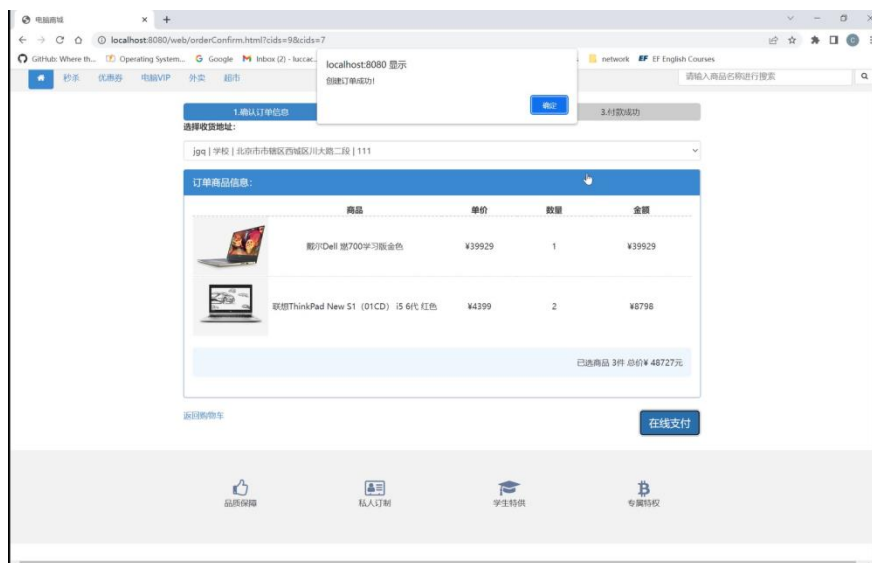
删除后：



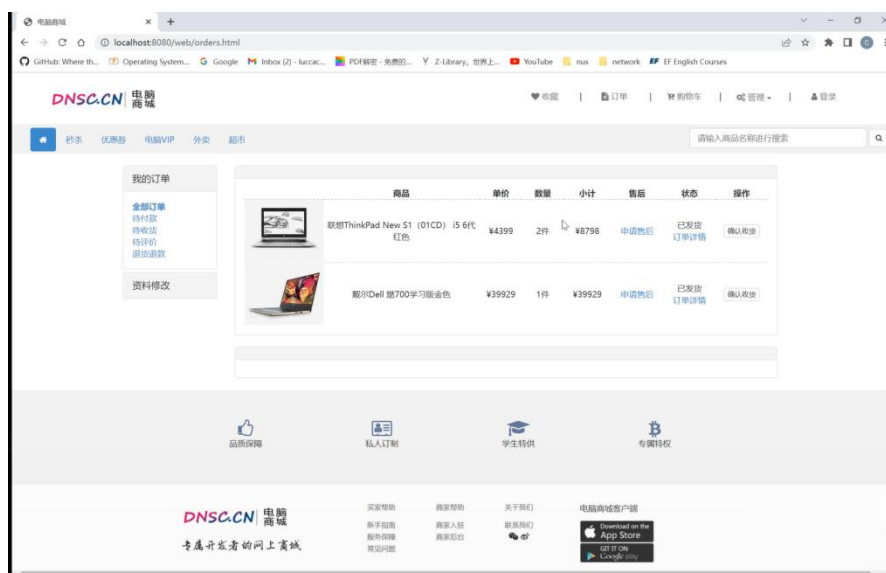
#### 4、订单处理方面

(1) 从购物车中选择商品创建订单：在购物中选择要支付的产品，点击在线支付，即可成功创建订单





(2) 显示订单信息：点击上方菜单栏中的订单即可查看订单的状态，并可以确认收货



## 六、总结

### 1.体会：

首先，对于本项目乃至所有项目，需求分析都是很重要的，我们在需求分析阶段就明确了需求，依此来设计，对我们有很大的帮助。

其次，数据库的设计是至关重要的，它关系到后端进行接口开发时实现的难度，数据库中数据的可维护性，一致性，数据冗余等问题。刚开始做设计时，先把自己能想到的需要用到的数据表画出来，并根据需求暂时给出数据表的几个大字段，这样的效率是低下的，后来我们根据需求分析的功能设计一步步设计添加表、字段，这样是高效的。

然后，组员之间的同步问题也很重要，组员可能对文件的做出不同修改，应避免同时修改一个文件的可能，同时同步自己的工作，来保持工作的一致性。

最后，选择高效开发工具往往会事半功倍。初始学习数据库时，使用命令行操作十分繁琐。做本次项目时，采用 Navicat 这种图形化界面操作数据库的软件，比命令行方便很多；同时采用 maven 来构建、管理项目，能避免出现很多错误。用 MyBatis 作为持久层框架，避免了大部分的 JDBC 代码和手动设置参数，节约了很多时间……

## 2.亮点：

- 1) 前端和后端通信采用了异步通信技术 AJAX，能够快速地将增量更新呈现在用户界面上，而不需要重载整个页面，能够更快地回应用户的操作；
- 2) 使用分层开发，使得各模块符合“高内聚低耦合”，同时方便展开测试；
- 3) 使用 springboot 框架、navicat 等管理工具，方便进行维护。

## 3.有待完善：

有些后端功能尚未实现，只是做出了静态界面。展示的时候发现项目重心偏了一点，过多侧重在用户相关的细节功能上，忽视了商城管理员上架货品等核心功能。