

《电子商城》

系统设计与开发说明

课程名称 Java 语言与 WEB 应用程序开发

开课学期 2022 至 2023 学年 第 一 学期

年 级 2021 级 专业班级 网络工程

学 生 李 俊

学 号 22202321072008

第一章 绪论

1.1 系统的目的和意义

电子商务在中国历经二十余年发展，已经逐步成为国民经济支柱，涉及基础 3 大产

业，渗透进入人们的生活。

分析电商优缺点：

- 电子商务将传统的商务流程电子化、数字化，一方面以电子流代替了实物流，可以大量减少人力、物力，降低了成本;另一方面突破了时间和空间的限制，使得交易活动可以在任何时间、任何地点进行，从而大大提高了效率。
- 电子商务所具有的开放性和全球性的特点，为企业创造了更多的贸易机会。
- 配送问题：配送费用高，交货耗时长，缺乏系统、统一化的配送企业，订单和分发的系统不够完善，影响了顾客的电商购物体验。

开发意义

电子商务一方面破除了时空的壁垒，另一方面又提供了丰富的信息资源，为各种社会经济要素的重新组合提供了更多的可能，这将影响到社会的经济布局和结构。开发电商后台管理系统的意义在于，电商管理系统中，最关键的是商品，因为商品是整个电商平台运营的关键所在，很多情况下，我们只要在后台创建内容，再通过前台进行展示即可，只要有了商品，上架就能展示在客户面前。同样的，商品也是电商后台产品的基础，很多电商平台在拥有了电商管理系统后，就能为同类型产品提供标准化的属性，这样以来就能更好的去统一产品。

第二章 需求分析

2.1 业务功能需求

2.1.1 用户注册与登陆功能

2.1.2 登陆时提供图片验证码功能

2.1.3 安全检测过滤器功能

2.1.4 对数据库的增删查改功能

2.1.5 文件或图片上传功能

2.1.6 实现类似购物车功能

2.2 数据存储需求

服务器端文件存储，存储管理员用户上传的商品图片。

数据库存储商品的信息。商品名、价格、种类、描述等等。

数据库存储用户信息。用户名、密码、电话、用户种类（只有 normal 和 admin 两种，其中 admin 是管理员用户，可以对商品进行增删查改）等等。

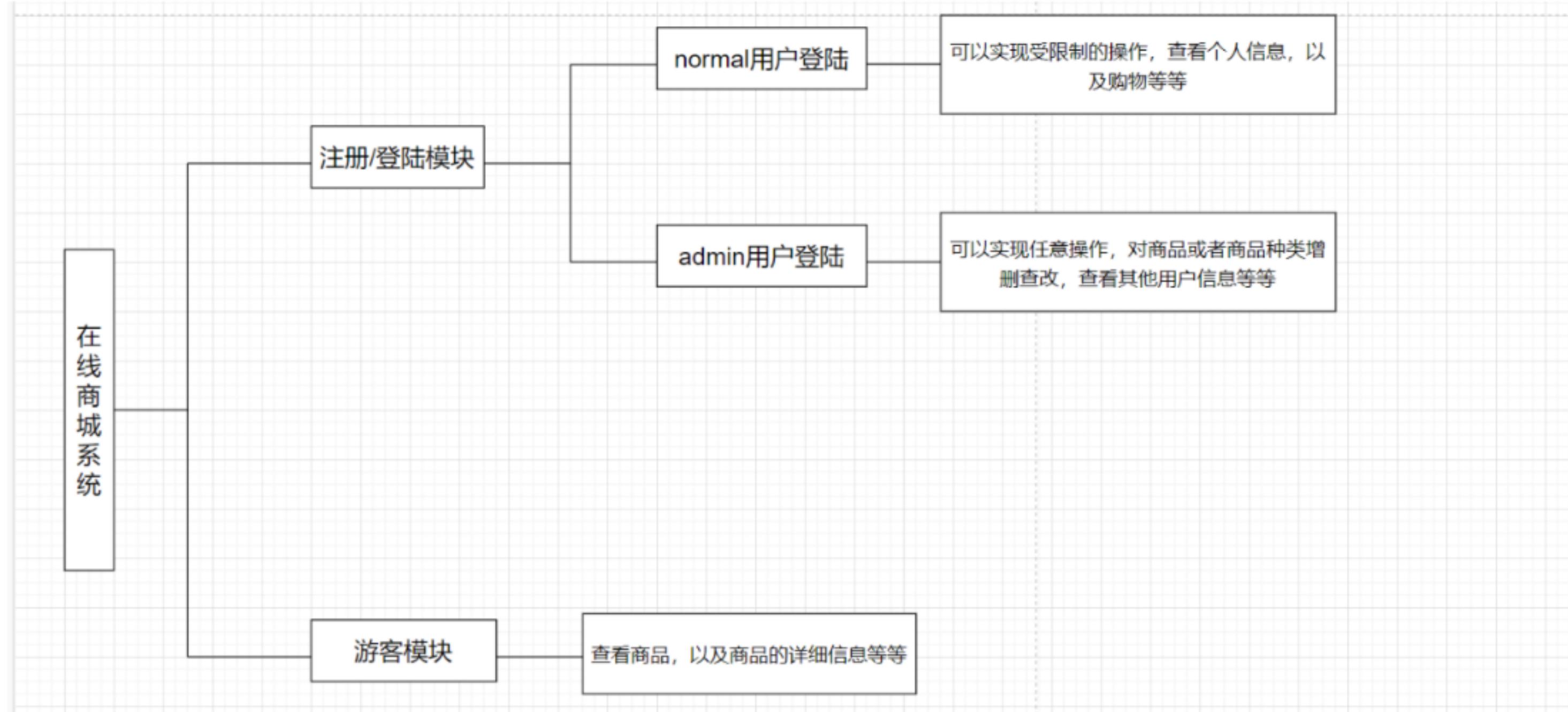
数据库存储商品种类和对应描述。

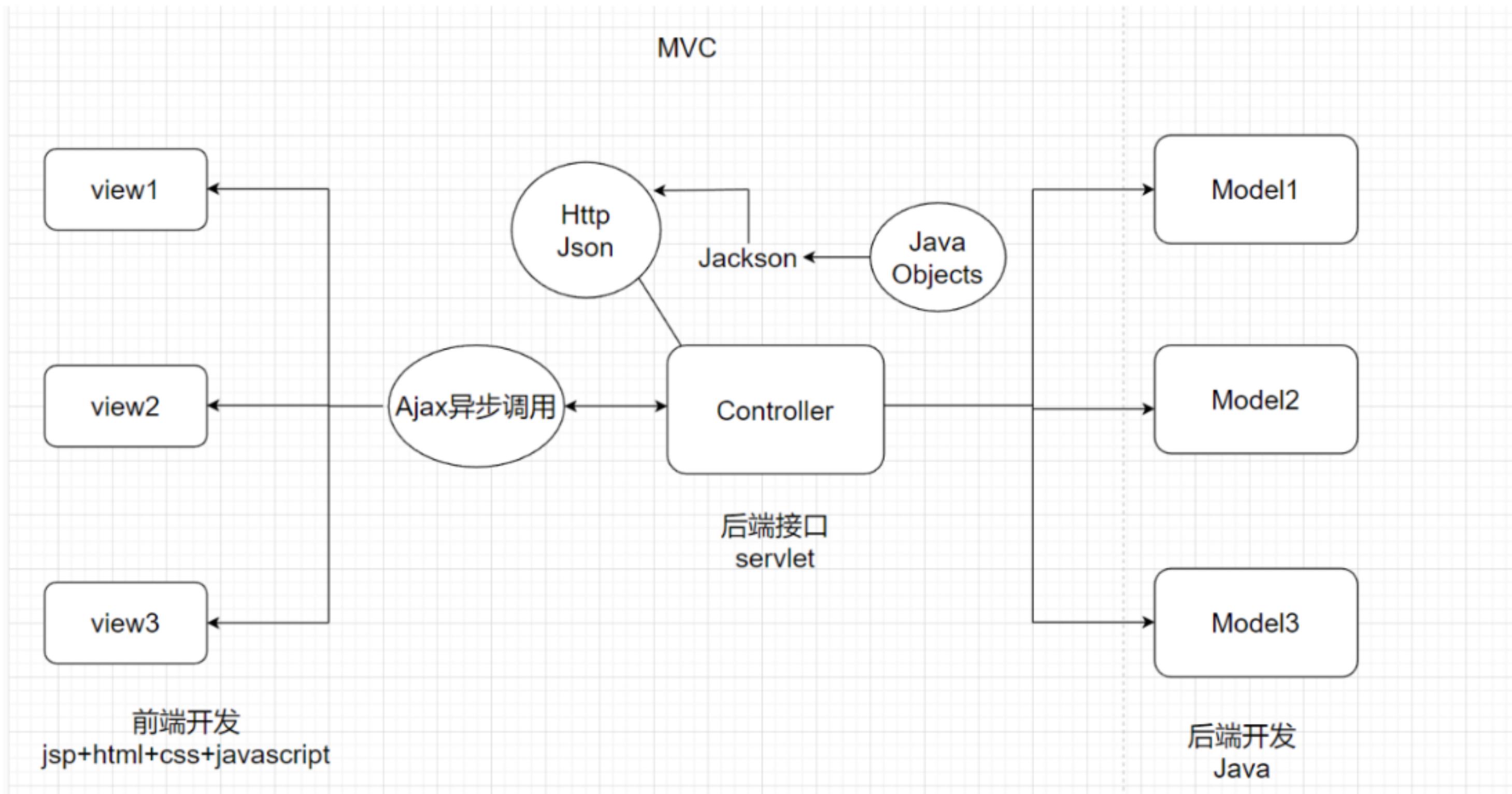
2.3 权限管理需求

需要登录之后才可以访问后台，后台分为两种，一种是 normal 用户的后台，一种是 admin 用户的后台，对应的用户只可以访问自己的后台，未登录只能浏览商品，不可以访问后台。

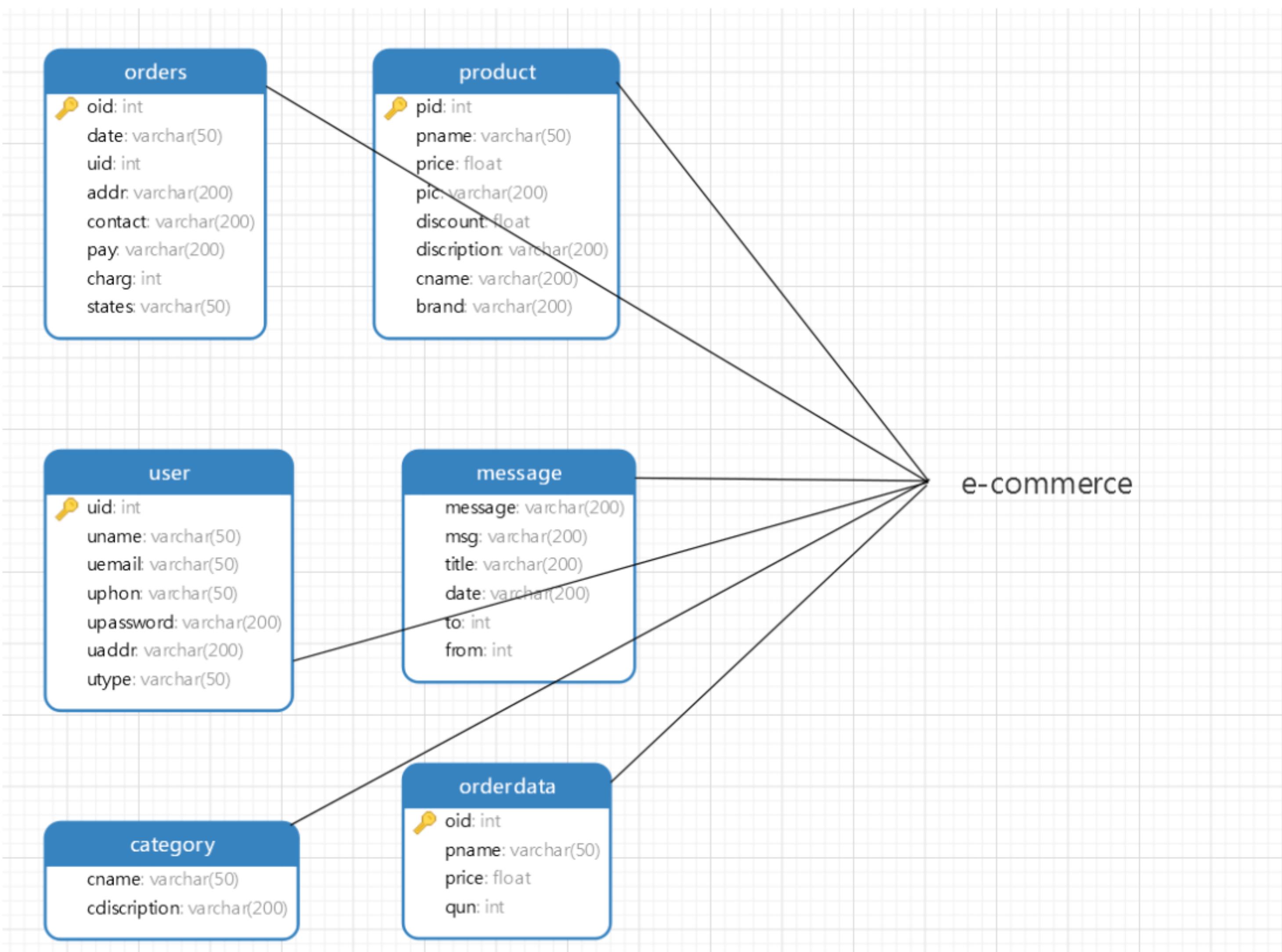
第三章 系统设计与技术

3.1 系统模块结构设计





3.2 系统数据库结构设计



如上使用 Navicat 导出的本项目数据库 ER 图，一共有六张表：



product 表格

字段拥有 pid, pname, price, pic, discount, discription, cname, brand。主键为 pid, 自增长。分别表示 id, 商品名, 商品价格, 商品图片, 商品折扣, 商品描述, 商品所属的种类名, 商品的品牌名。实现管理的主要功能, 商品的增删查改。

user 表格

字段拥有 uid, uname, uemail, uphon, upassword, uaddr, utype。主键为 uid, 自增长。分别表示 id, 用户名, 用户邮箱 (gmail), 用户的电话, 用户密码, 用户地址, 用户种类 (admin / normal, admin 相当于管理员用户)。通过该表提供用户的登录功能, 以及访问权限问题、过滤器 Filter 的拦截问题等。

orders 表格

字段拥有 oid, date, uid, addr, pay, charg, states。主键为 oid, 自增长。分别表示 id, 日期, 购买者名字, 付款, 收费, 状态。

orderdata 表格

字段拥有 oid, name, price, qun。主键为 oid, 自增长。分别表示

category 表格

描述商品种类, 有 cname 和 cdiscription 两个字段。

message 表格

拥有 message、msg、title、date、to、from 字段。描述信件。

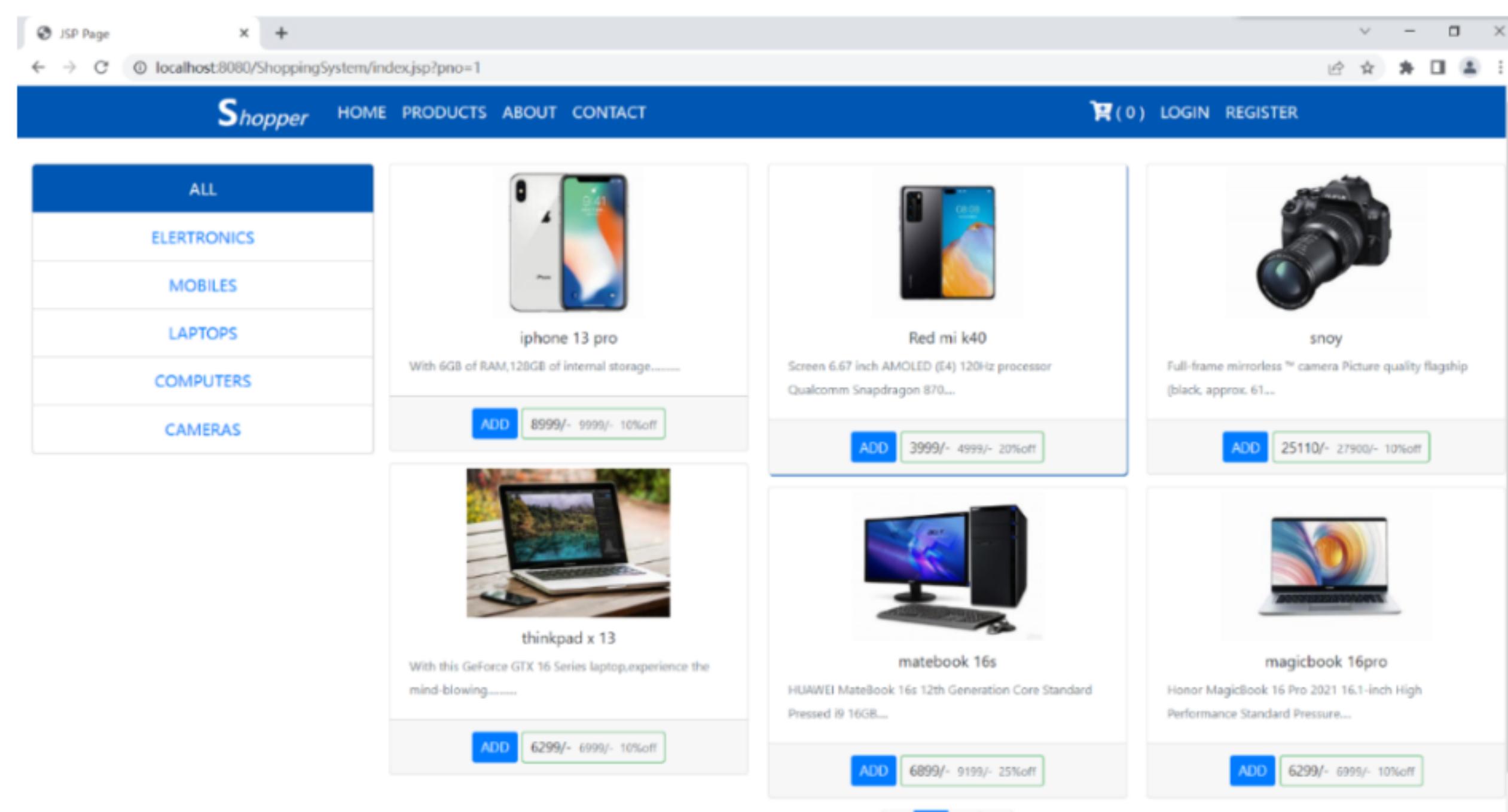
3.3 关键技术

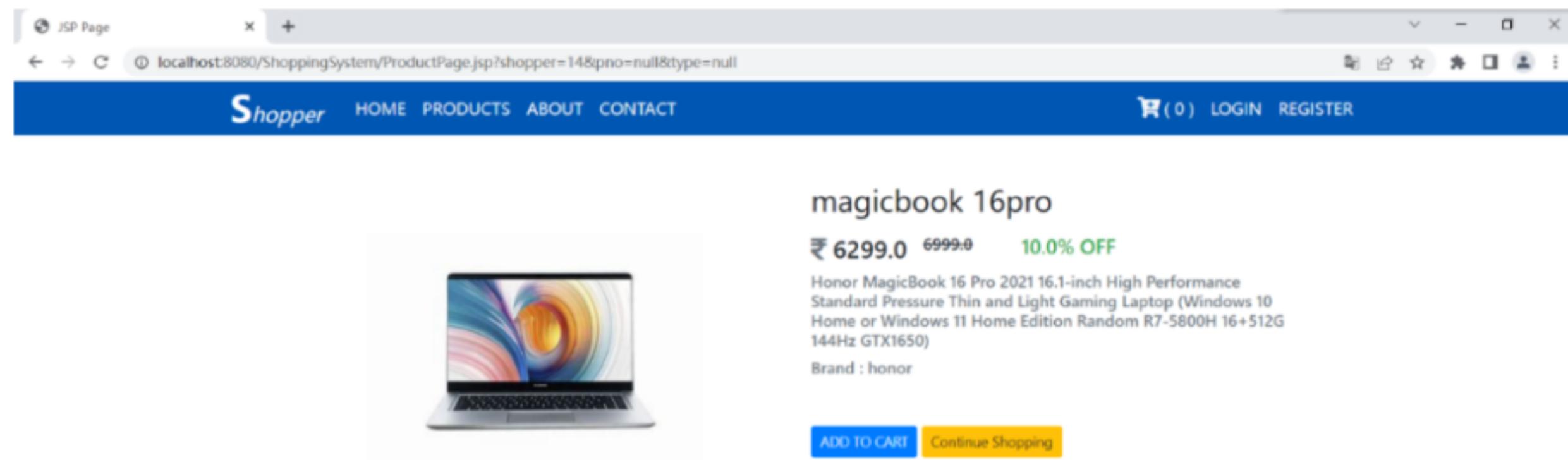
权限管理。一开始只知道根据 session, 把没有登陆的游客拦截, 不能访问被过滤的资源。但是不知道如何根据不同的用户, 登陆之后进入不同的后台, 后来查阅资料知道, 可以在用户的数据库里面添加一个字段用来标识是普通用户还是管理员用户, 这里我在 user 表里面添加了 type 字段, 管理员用户的 type 就是 “admin”, 普通用户的 type 就是 “normal”。在登陆之后, 通过 request 获取相关信息, 再到数据库里面查找 user 的信息, 如果得到的 type 是 “admin” 就进入管理员用户的后台, 如果得到的 type 是 “normal”, 就进入普通用户的后台。这样就实现了不同的用户登陆之后进入不同的后台。

第四章 系统实现

4.1 商城功能模块实现

游客功能模块





如上图，未登录情况下，打开网页可以直接浏览商品，查看商品详细信息。

注册/登陆模块

Register here

Enter the your name
name must be greater than 3 character

Enter the E-mail
e-mail enter in sequence

Enter the Contact number
Enter valid contact

Enter the Password
password greater than five character

Address :
enter the Adress

Accept the term and conditions

Submit Reset

JSP Page x + localhost:8080/ShoppingSystem/login.jsp

Shopper HOME PRODUCTS ABOUT CONTACT **(0)** LOGIN REGISTER

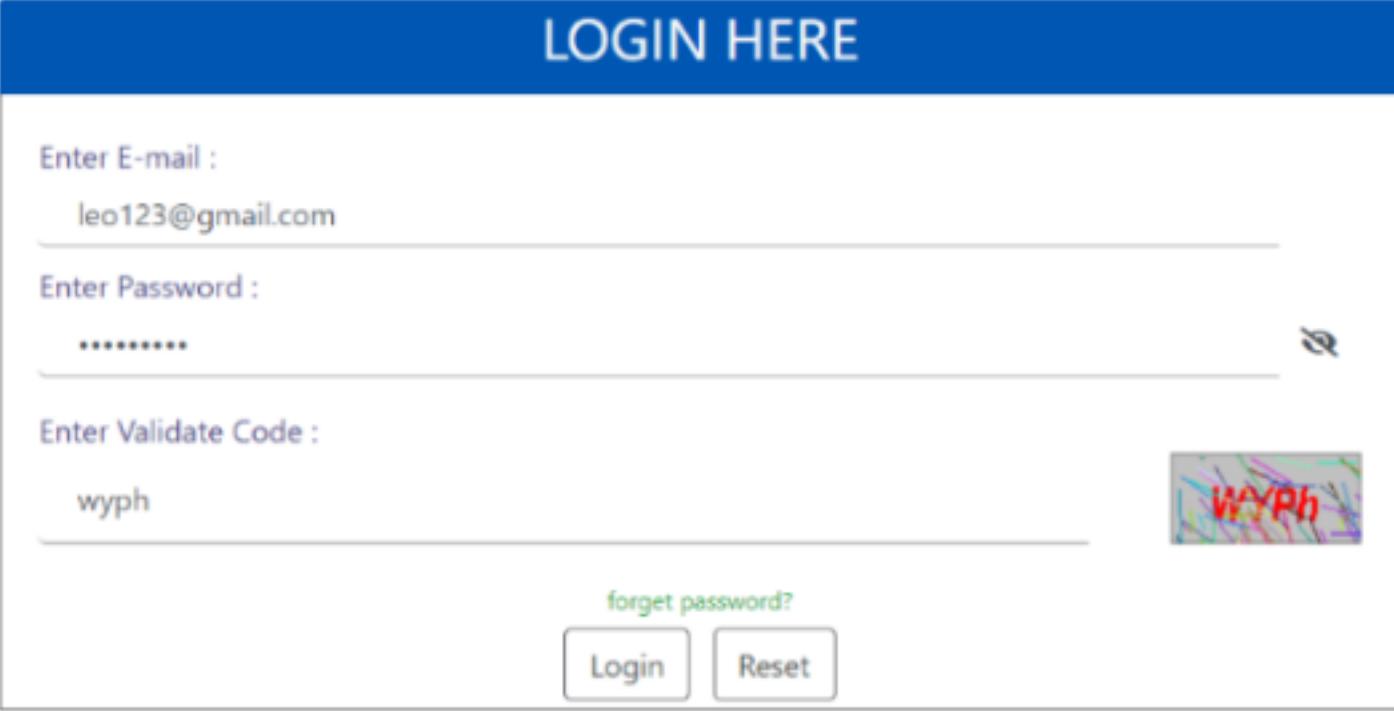
LOGIN HERE

Enter E-mail :
leo123@gmail.com

Enter Password :

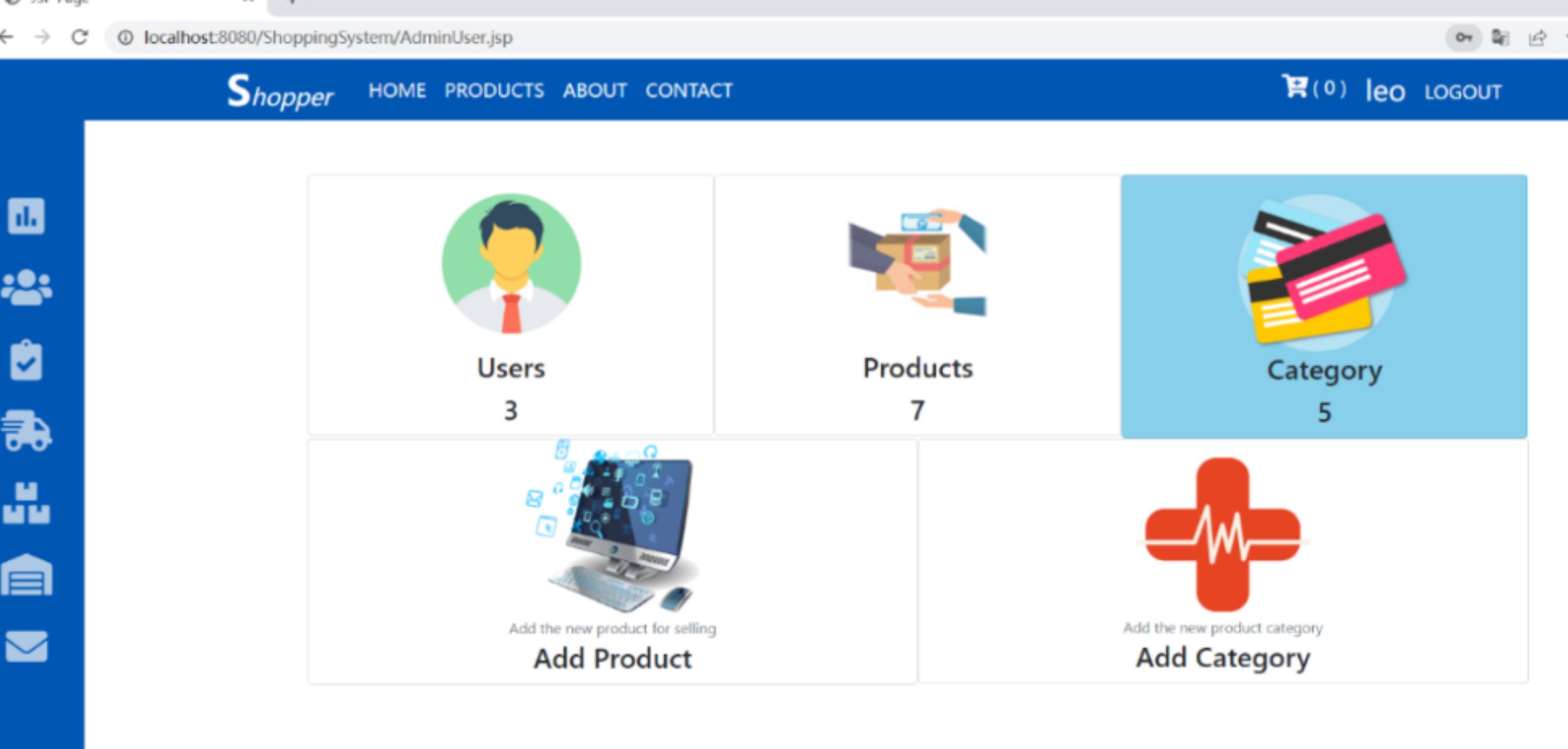
Enter Validate Code :
wyph

forget password?



JSP Page x + localhost:8080/ShoppingSystem/AdminUser.jsp

Shopper HOME PRODUCTS ABOUT CONTACT **(0)** leo LOGOUT



Users	Products	Category
3	7	5

Add the new product for selling
[Add Product](#)

Add the new product category
[Add Category](#)

The screenshot shows the 'USER DETAILES' section of the admin interface. It displays a table with three rows of user information:

ID	Name	Email	Phone	Address
1	leo	ljun3889@gmail.com	18168842456	changzhouprovince
2	leo	leo123@gmail.com	1340160279	province jiangsu
3	mingfei lu	lmf123@gmail.com	1335689452	Kassel College

A search bar at the top says 'Find the specific user by ID'. A sidebar on the left has icons for dashboard, users, products, categories, reports, and messages.

如上多图，是 **admin** 用户登陆后台可以看到的界面，可以添加商品种类，添加商品，查看所有用户信息等等。但是普通的 **normal** 用户则没有这个权限，普通用户后台只能看到如下图这些信息，同时也可以浏览商品，添加到购物车。

The screenshot shows the 'NormalUser.jsp' page. On the left, there's a sidebar with sections for 'Information' (Name: mingfei lu, E-mail: lmf123@gmail.com, Contact: 1335689452, Address: Kassel College) and 'Order Now...' (Address: Kassel College, Contact: 1335689452, Payment by: Paypal). On the right, it shows a 'Your selected products' table with five items:

Name	Price	Quantity	Cahrge price	Action
iphone 13 pro	8999.1	1	8999.1	<button>Remove</button>
Red mi k40	3999.2	1	3999.2	<button>Remove</button>
snoy	25110.0	1	25110.0	<button>Remove</button>
magicbook 16pro	6299.1	1	6299.1	<button>Remove</button>
matebook 16s	6899.25	2	13798.5	<button>Remove</button>

Total Price : 58205.902

4.2 用户注册与登陆功能实现

注册：目前只支持谷歌邮箱注册，即用@gmail.com

The screenshot shows a web browser window with four tabs open. The active tab is 'register.jsp' at 'localhost:8080'. The page has a dark blue header with the 'Shopper' logo, 'HOME', 'PRODUCTS', 'ABOUT', and 'CONTACT' links, and a shopping cart icon showing '(0)'. Below the header is a form titled 'Register here'. The form fields are:

- Name: Enter the your name (name must be greater than 3 character)
- E-mail: Enter the E-mail (e-mail enter in sequence)
- Contact number: Enter the Contact number (Enter valid contact)
- Password: Enter the Password (password greater than five character)
- Address: enter the Adress

Below the form is a checkbox for 'Accept the term and conditions' and two buttons: 'Submit' and 'Reset'.

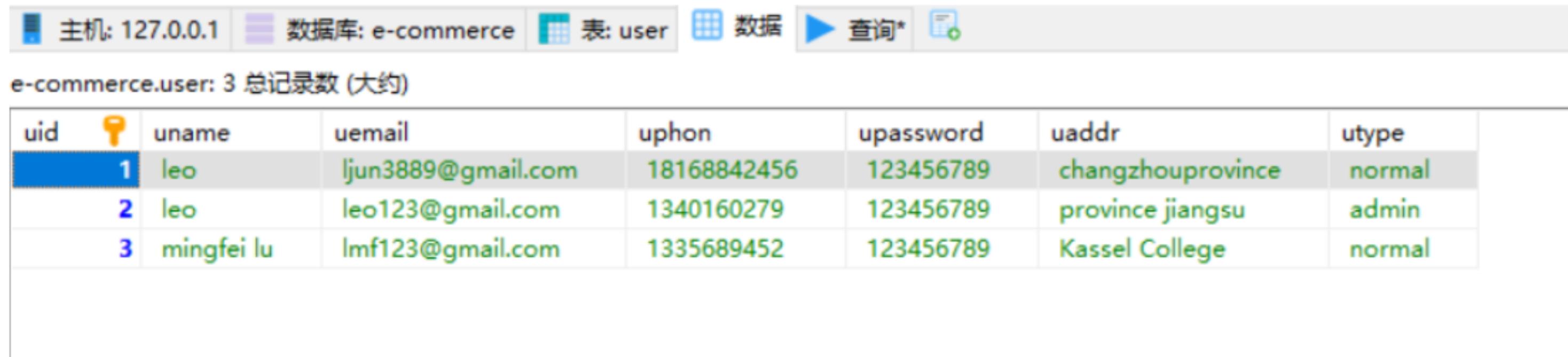
登陆：

The screenshot shows a web browser window with four tabs open. The active tab is 'login.jsp' at 'localhost:8080'. The page has a dark blue header with the 'Shopper' logo, 'HOME', 'PRODUCTS', 'ABOUT', 'CONTACT', and a shopping cart icon showing '(0)'. Below the header is a form titled 'LOGIN HERE'. The form fields are:

- E-mail: Enter your register email
- Password: Enter your register password
- Validate Code: Enter the Validate Code on the right (with a CAPTCHA image showing 'IXFZ')

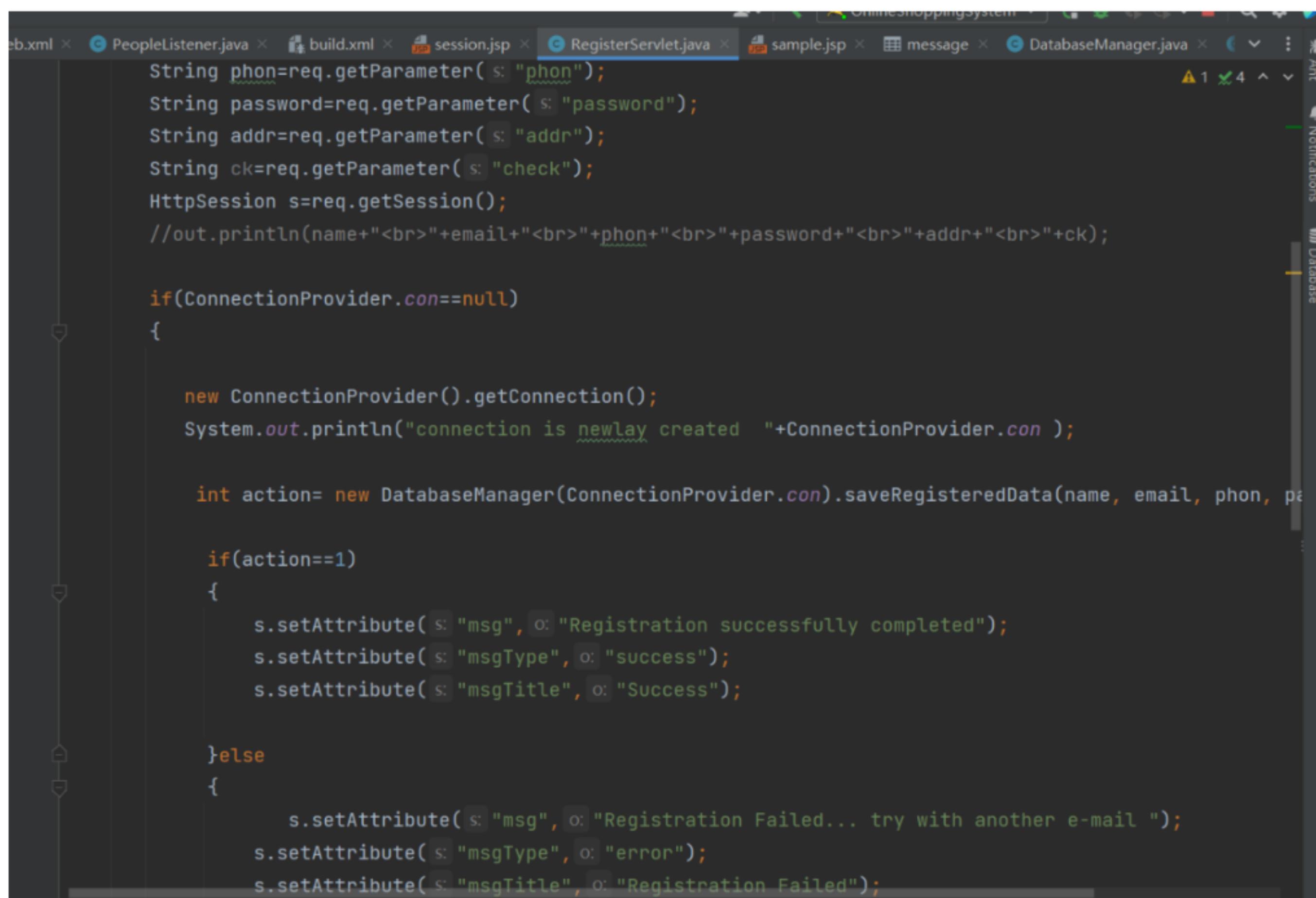
Below the form is a link for 'forget password?' and two buttons: 'Login' and 'Reset'.

数据库：



主机: 127.0.0.1	数据库: e-commerce	表: user	数据	查询*	导出
e-commerce.user: 3 总记录数 (大约)					
uid	uname	uemail	uphon	upassword	uaddr
1	leo	ljun3889@gmail.com	18168842456	123456789	changzhouprovince
2	leo	leo123@gmail.com	1340160279	123456789	province jiangsu
3	mingfei lu	lmf123@gmail.com	1335689452	123456789	Kassel College

代码展示:



```

String phon=req.getParameter( s: "phon");
String password=req.getParameter( s: "password");
String addr=req.getParameter( s: "addr");
String ck=req.getParameter( s: "check");
HttpSession s=req.getSession();
//out.println(name+"<br>"+email+"<br>"+phon+"<br>"+password+"<br>"+addr+"<br>"+ck);

if(ConnectionProvider.con==null)
{
    new ConnectionProvider().getConnection();
    System.out.println("connection is newlay created "+ConnectionProvider.con );
}

int action= new DatabaseManager(ConnectionProvider.con).saveRegisteredData(name, email, phon, pa
if(action==1)
{
    s.setAttribute( s: "msg", o: "Registration successfully completed");
    s.setAttribute( s: "msgType", o: "success");
    s.setAttribute( s: "msgTitle", o: "Success");

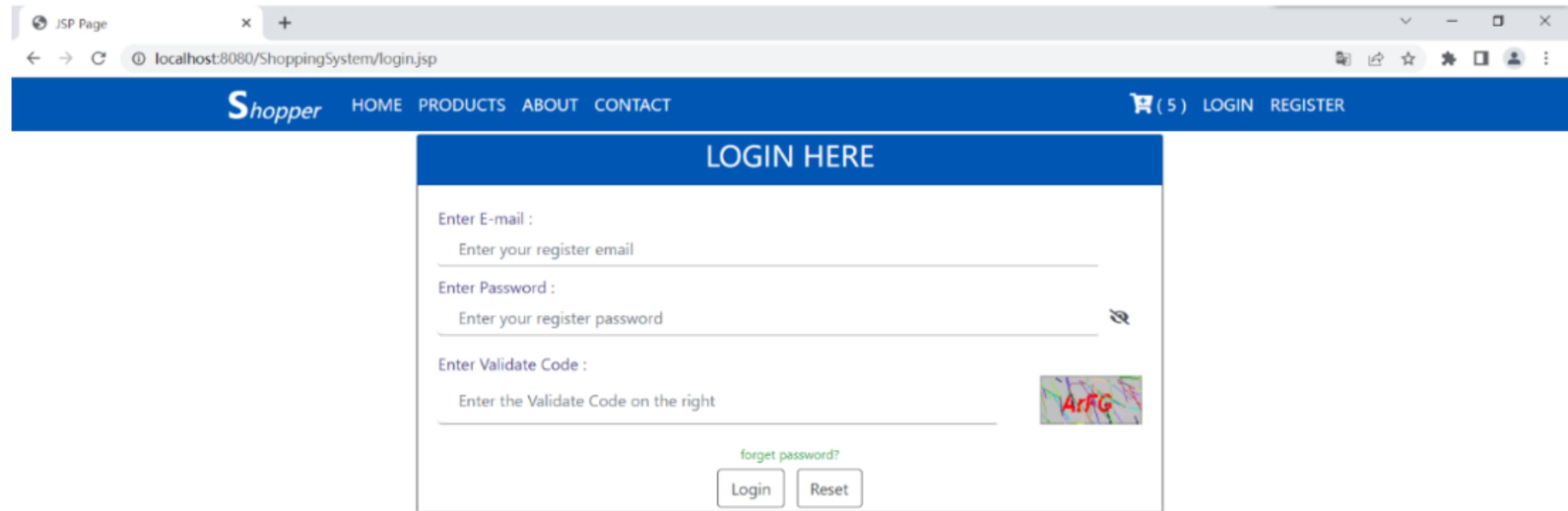
}else
{
    s.setAttribute( s: "msg", o: "Registration Failed... try with another e-mail ");
    s.setAttribute( s: "msgType", o: "error");
    s.setAttribute( s: "msgTitle", o: "Registration Failed");
}

```

此功能要通过 e-commerce 数据库，创建一个 user 表格，来存储用户信息。用户注册之后，在数据库中保存对应的信息。登录的时候，通过 submit，在 request 里面查找输入的用户信息，和数据库中相匹配，匹配无误即登陆成功，跳转页面。

4.3 登陆时提供图片验证码功能实现

验证码:



关键代码：

```

private final static int CODE_SIZE = 4;
2 usages
public final static String LOGIN_VERIFY_CODE = "login_verify_code";

public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
    String code = this.randomCode(CODE_SIZE);

    HttpSession session = request.getSession( b: true);
    session.setAttribute(LOGIN_VERIFY_CODE, code);

    BufferedImage image = this.drawImage(WIDTH, HEIGHT, code);
    try (OutputStream outputStream = response.getOutputStream()) {
        ImageIO.write(image, IMAGE_TYPE, outputStream);
    }
}

public BufferedImage drawImage(int width, int height, String code) throws IOException {
    BufferedImage image = new BufferedImage(width, height, BufferedImage.TYPE_INT_RGB);
    Graphics g2d = image.getGraphics();

    g2d.setColor(Color.LIGHT_GRAY);
    g2d.fillRect( x: 0, y: 0, width, height);

    g2d.setColor(Color.GRAY);
    g2d.drawRect( x: 0, y: 0, width: width-1, height: height-1);
}

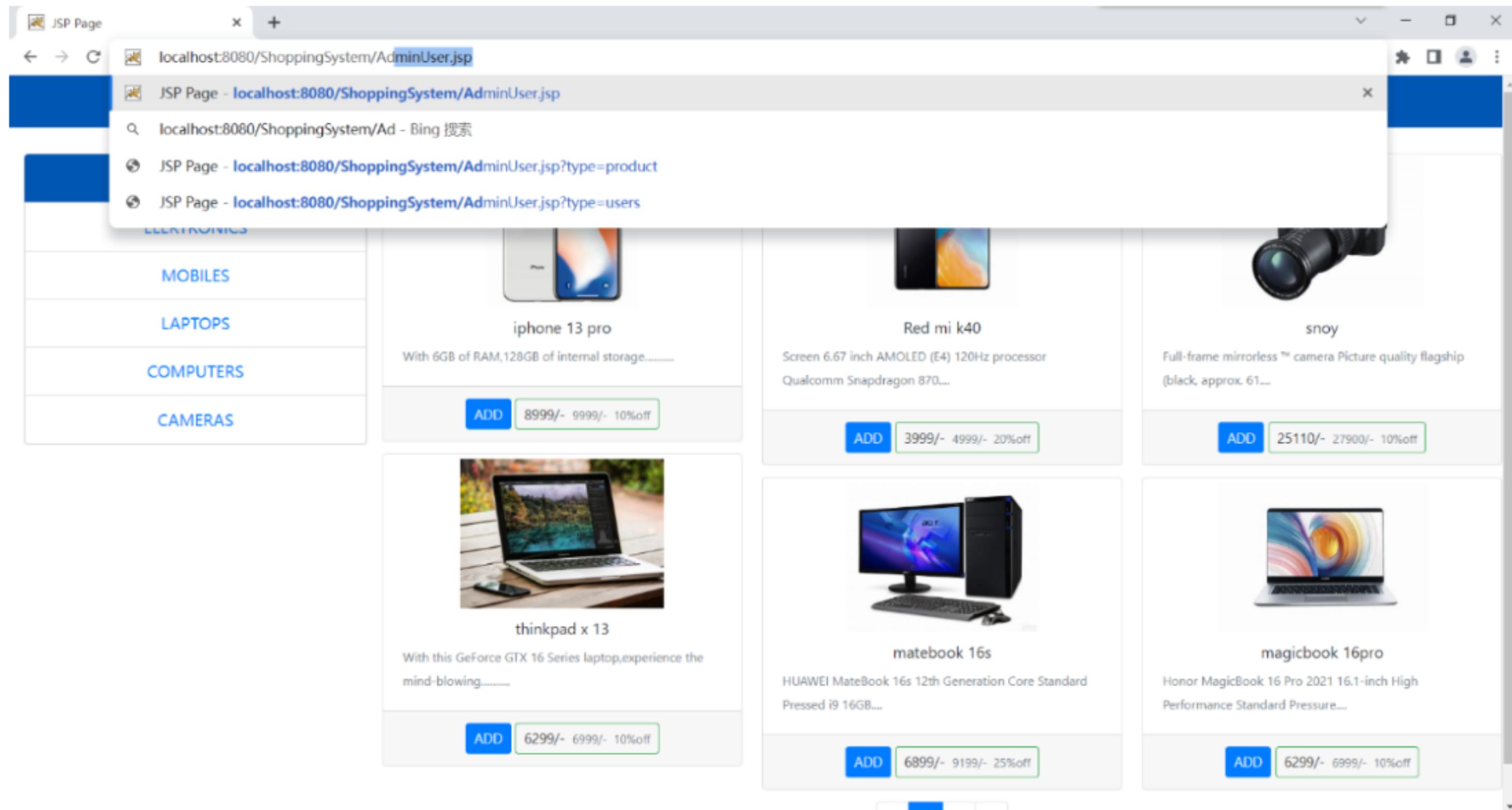
```

此功能，要先生成一张特定大小空白图片，然后再空白图片上画出随机生成的验证码，当然，这个验证码生成的同时被用一个变量保存。当输入完验证码，点击 submit 提交之后，通过比较 request 里面获得的用户输入的验证码，和 servlet 里面的变量保存的验证码，确认验证码是否输入正确（忽

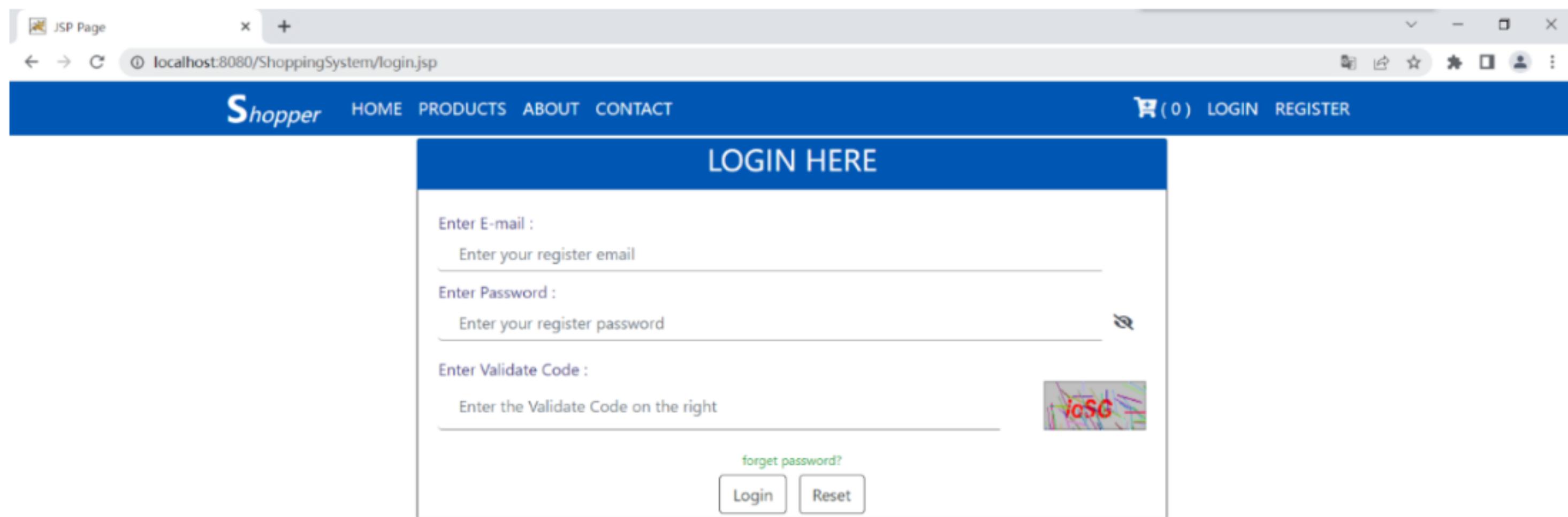
略大小写）。

4.4 安全检测过滤器功能实现

如下图，想要直接输入网址访问管理员用户的后台，不可以，会被拦截，跳转到 login 界面。

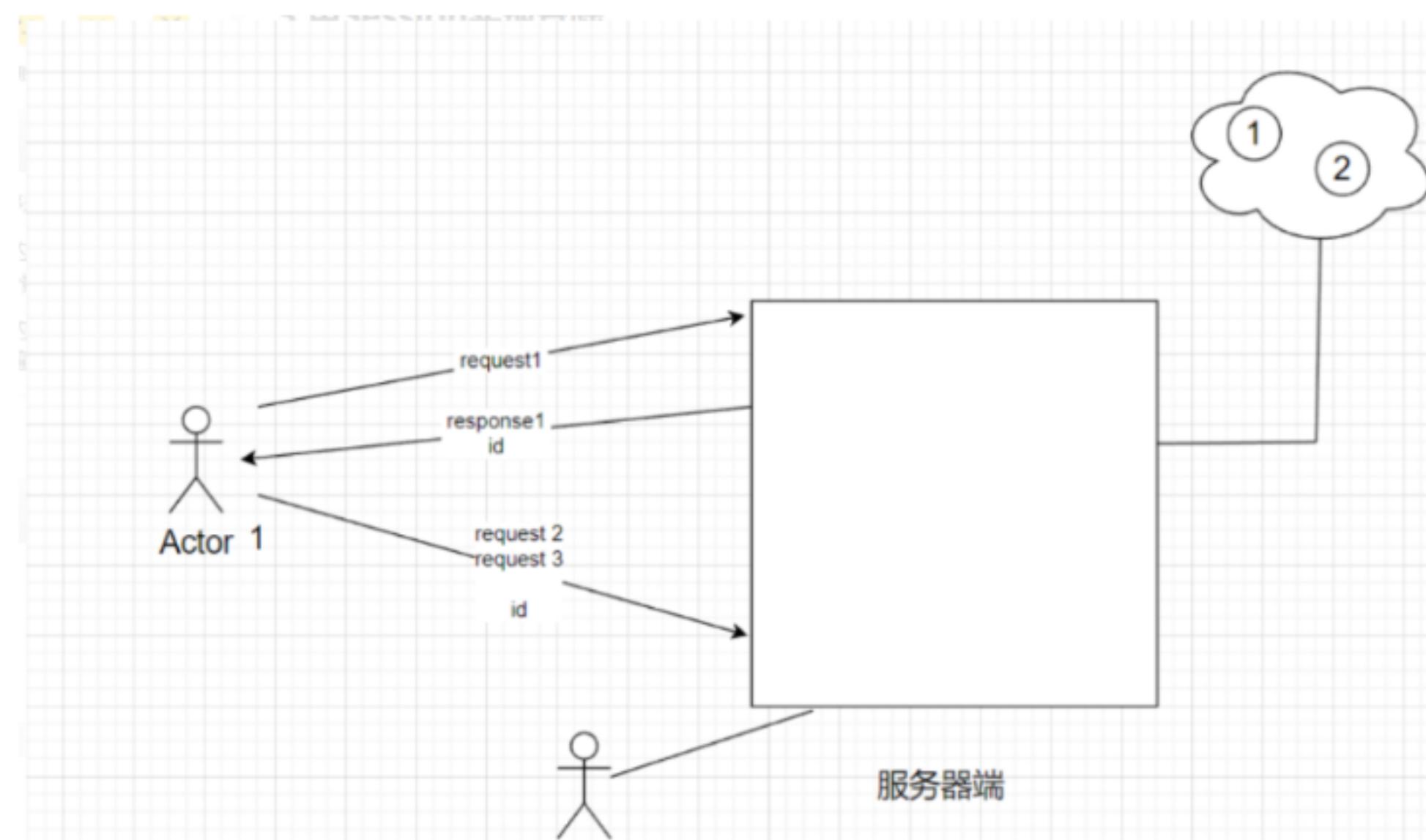


如下，被拦截



在客户端访问的时候，tomcat 会自动在服务器端创建一个 Session 存储相关信息，如下，Actor 1 绑定的 Session 是 1（圆圈），Actor 2 绑定的 Session 是 2。Session 对象和用户的每次访问都是绑定在一起的。

比如在 Session 里面创建一个 flag，标记用户是否登陆，如果未登陆却想使用被限制的资源，就跳转到 login 界面让他登陆。如果已经登陆，就可以使用网站里的资源，就可以实现平常使用的登陆功能。

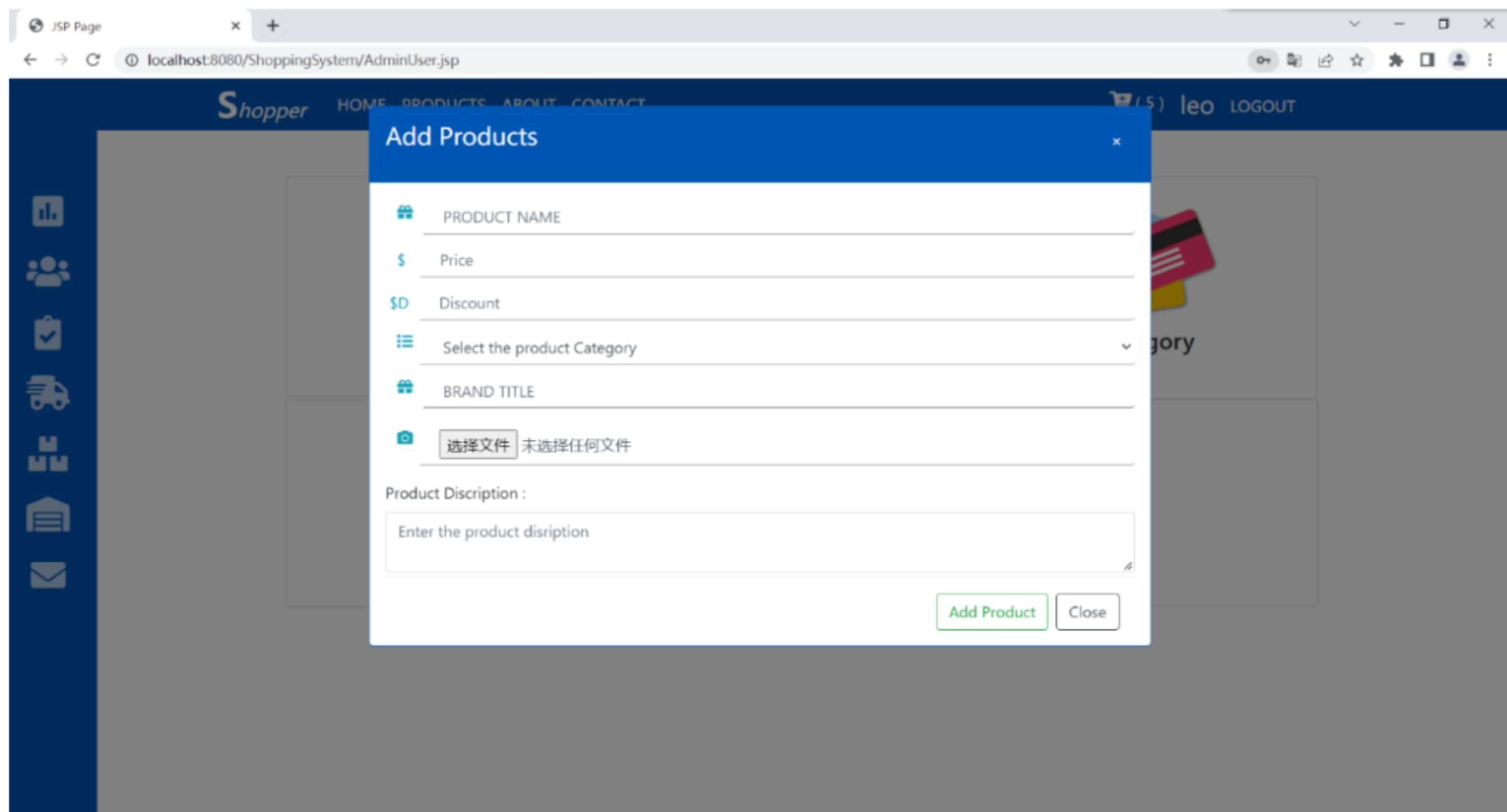


在这里进行简单的操作，使用 Filter 对所有资源进行过滤，这样子所有资源都无法直接访问，也包括一些图片资源之类的。当然，像一些图片资源、登陆界面、初始界面等等，可以直接“放行”，不需要对其进行限制。

除了直接“放行”的资源，其余资源通过 Session 来判断是否放行。如果没有 Session，那么肯定就是未登陆过，就需要登陆。如果有 Session，有的网站会一进入就创建 Session，所以有两种情况，一种是已经登陆，一种是没有登陆，但是自动创建 Session，两者区分要用 Session 里面的 flag 来判定（flag 是上文说过的来标记是否登陆过的信息）。逻辑如下，整个逻辑并不特别复杂。

使用 Filter 进行过滤之后，即使得到了后台网址，也不可以直接访问，要先登录，安全性大大提高。就好比我们平时登陆校园网，如果直接输入同学给的内部链接，不可以直接访问操作界面，跳出来的是登陆界面。

4.5 数据库的增删查改功能实现



This screenshot shows a table titled 'PRODUCT DETAIL' listing five products. The columns are ID, Name, Price, Discount, Brand, and Action (with 'Edit' and 'Delete' buttons). The products listed are:

ID	Name	Price	Discount	Brand	Action
9	iphone 13 pro	9999.0	10.0	apple	<button>Edit</button> <button>Delete</button>
10	thinkpad x 13	6999.0	10.0	lenovo	<button>Edit</button> <button>Delete</button>
11	Red mi k40	4999.0	20.0	xiaomi	<button>Edit</button> <button>Delete</button>
12	matebook 16s	9199.0	25.0	huawei	<button>Edit</button> <button>Delete</button>
13	snoy	27900.0	10.0	sony	<button>Edit</button> <button>Delete</button>

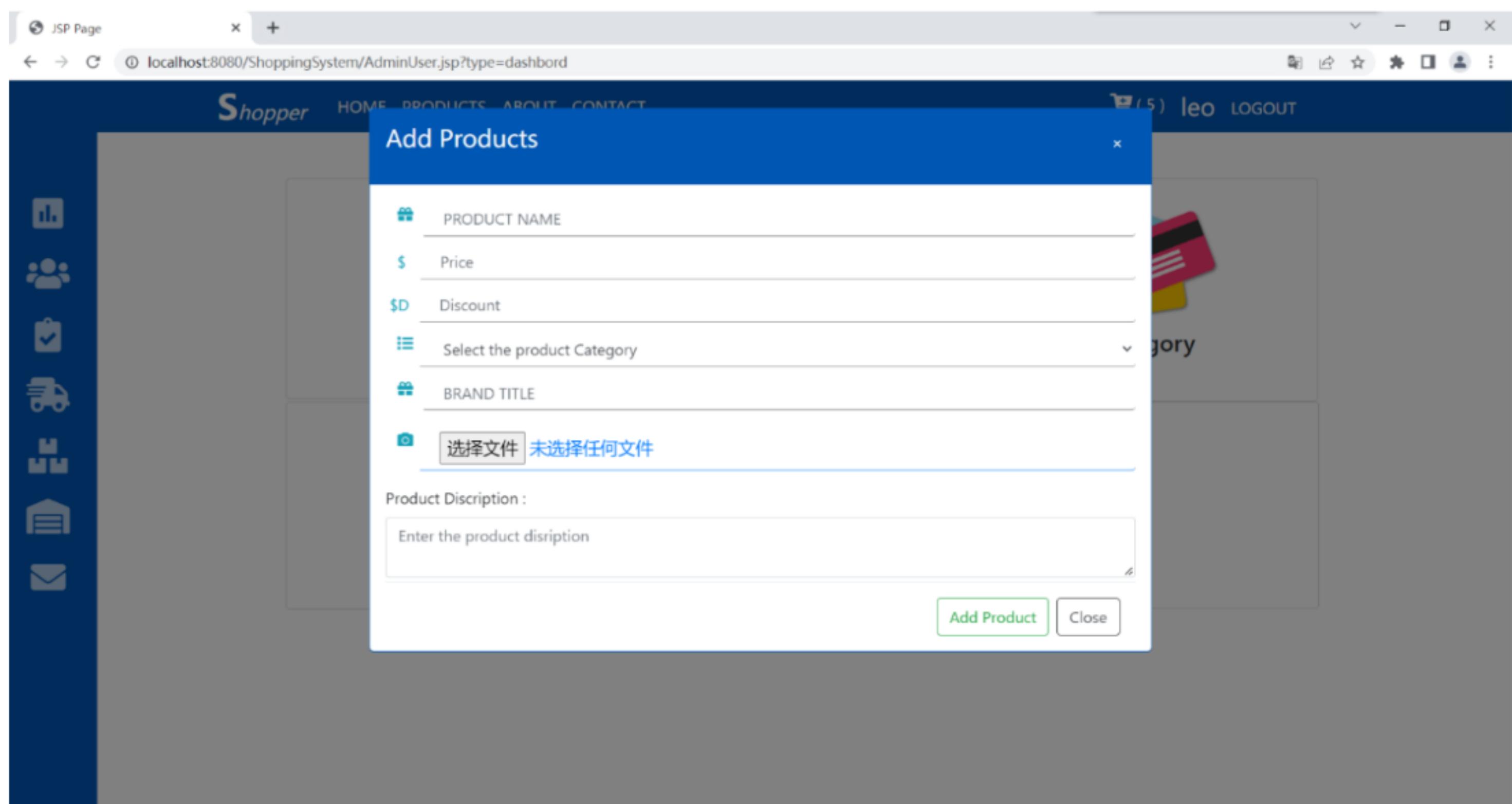
查不方便展示，比如登陆的时候，从数据库里面找用户输入的账号密码，有相匹配的就算找到了。
这就是查。

部分关键代码展示

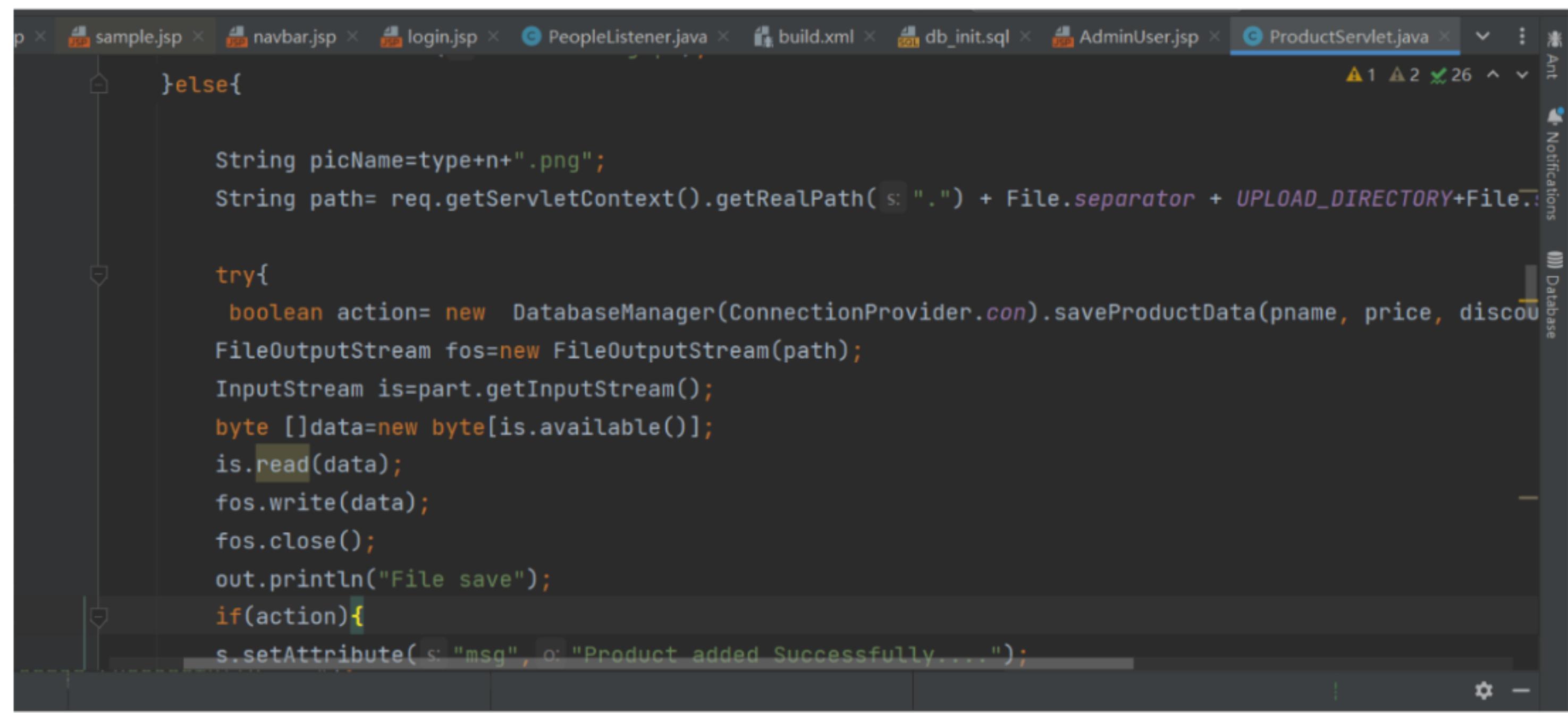
```
usage
public boolean saveProductData(String pname, float price, float discount, String type, String location, String pic, String description, String cname, String brand) {
    boolean a=false;
    try {
        String Query="insert into product(pname,price,discount,pic,description,cname,brand) values('"+pname+"','"+price+"','"+discount+"','"+pic+"','"+description+"','"+cname+"','"+brand+"')";
        Statement st=con.createStatement();
        st.executeUpdate(Query);
        a=true;
    } catch(Exception e) {
        System.out.println("(DBManager) save product Generate error : "+e.getMessage());
    }
    return a;
}
```

该功能主要在 DatabaseManager 里面进行操作。最核心的就是 sql 语句。比如，增加商品，在网页中输入相关信息之后，submit，然后由 request 获取，再将对应的信息通过 sql 语句增加到数据库里面。删除则更加容易，管理员用户在商品后台，点击 delete，就可以删除，其逻辑就是点击 delete 的时候，执行删除的 sql 语句。至于查也很容易，上文提到的登陆的时候，匹配输入的账号密码和数据库里面的账号密码就是此操作。改则可以体现在商品信息更改的情况下，网页输入对应信息，submit 之后，通过 request 获取输入的信息，并执行 update 的 sql 语句。

4.6 文件或图片上传功能实现



代码：



The screenshot shows a Java code editor with several tabs at the top: sample.jsp, navbar.jsp, login.jsp, PeopleListener.java, build.xml, db_init.sql, AdminUser.jsp, ProductServlet.java. The ProductServlet.java tab is active. The code is as follows:

```
    }else{

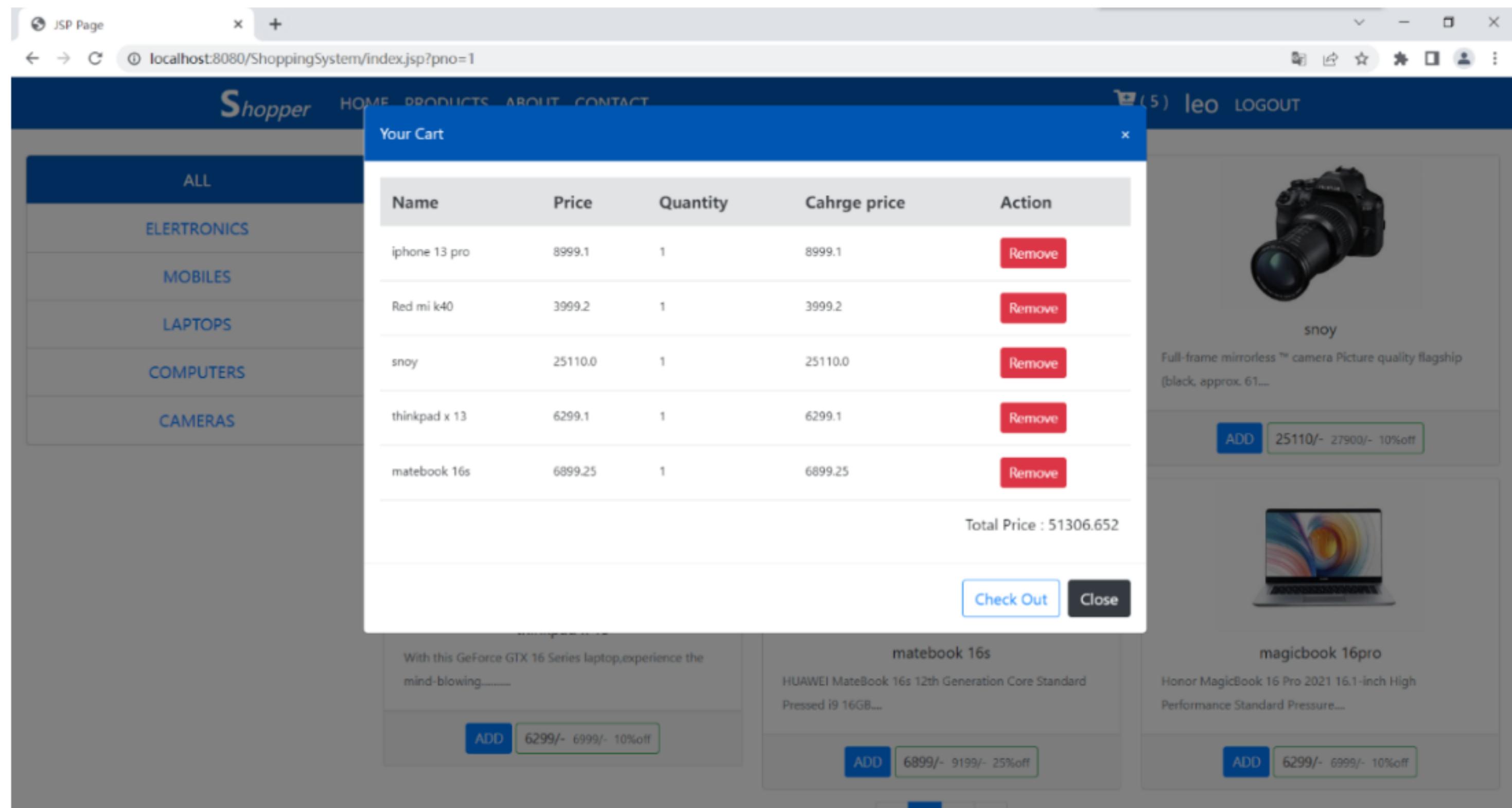
        String picName=type+n+".png";
        String path= req.getServletContext().getRealPath( s: "..") + File.separator + UPLOAD_DIRECTORY+File.separator+picName;

        try{
            boolean action= new DatabaseManager(ConnectionProvider.con).saveProductData(pname, price, discount, category, description);
            FileOutputStream fos=new FileOutputStream(path);
            InputStream is=part.getInputStream();
            byte []data=new byte[is.available()];
            is.read(data);
            fos.write(data);
            fos.close();
            out.println("File save");
            if(action){
                s.setAttribute( s: "msg", o: "Product added Successfully....");
            }
        }
    }
}
```

该功能需要在 WebRoot 目录下面准备一个 uploads 文件夹，用来存放上传的图片。在 admin 用户登录之后，可以在后台添加商品，添加商品的同时要上传照片，给图片根据类型命名并保存到 uploads 里面，另一方面，图片名要保存至数据库的 product 表中。打开网页之后，根据数据库的 product 表里面存储的图片名，在服务器端对应保存路径中寻找资源即可。

4.7 类似购物车功能实现





代码：

```

if(array==null){
    ArrayList<selectedProduct> Cart=new ArrayList<>();
    selectedProduct obj=new selectedProduct(name,id,1,price);
    Cart.add(obj);
    session.setAttribute("cart",Cart);
    //first product
    session.setAttribute("swal","fp");

    if(type.equalsIgnoreCase("all") || type==null){
        response.sendRedirect("index.jsp?pno="+pno);
    }else if(type.equalsIgnoreCase("productPage")){
        response.sendRedirect("ProductPage.jsp?shopper="+pno+"&pno=null&type=null");
    }else{
        response.sendRedirect("index.jsp?type="+type);
    }
}

else{
    boolean action=true;
    for(Object cartData:array)
    {
        selectedProduct data=(selectedProduct)cartData;
        if(data.getPid()==id){
}
}
}

```

当一个用户点击购买一件商品时，将该商品的对象存放到一个 list 集合中。之后就将这个集合存进 session。当要再购买一件商品时，同理，将该商品存进 session 中的 list 集合。下单时，就将 session 中的 list 集合遍历存放到订单明细表中。由于购物车是实时操作的数据，如果通过在服务器数据库中建表的形式保存购物车数据，那么将导致频繁操作数据库，用户量多的情况下会对数据库造成极大

负担，容易出现并发错误，或者直接崩溃。session 效率高，会话是针对各个链接的，便于管理，但是存在生命有效期，如果 session 超时，购物车中的数据将会丢失。

4.8 电子商城系统实现的不足

没有做到统计在线用户人数。

这个网站就类似于一个商家 (admin 用户) 开的店铺，如果可以做到在这个网站，可以注册不同的商家，开不同店铺，就类似于淘宝、京东等其他购物平台，就更好了。

如果可以做到用户和商家可以实时沟通，咨询商品的信息，系统也就更加完善了。