**Java Web开发**

**实验五**

**作业报告**

|  |  |
| --- | --- |
| 学 院 | 计算机 |
| 所在系 班 级 | 2017软件工程系6班 |
| 学 号 姓 名 | 04171411 |
| 指 导 教 师 | 郭晓燕 |
| 完 成 时 间 | 2019年4月10号 |

# 实验题目

1、 练习老师在理论课上演示的购物车实例，理解购物功能如何实现，理解如何利用session对象保存购物信息，如何设计封装数据类。（请参考“第四章会话管理课上的购物车实例代码.docx”文档）；

如果上课听懂了购物车的原理或者部分理解，可以参考我的实例完成第2题，此题可忽略不写。

2、 第四章课后练习3购物车，实现一个购物车应用程序，可以购物，显示购物数量并查看购物车内容。（此题模版test4E，提供给大家参考。）

提示：采购此书做成超链接，并用URL重写带关键信息

a) Shopping.view?bookID=‘0001’，其中0001是你选中图书对应的ID号，每点击一次就在相应的购书数量上加一，并返回当前的选购页面。

b) 除了要保存图书的ID号、书名，还要保存购书数量，所以session对象保存的信息需要考虑清楚，最好是一个对象

数组或列表。

# 分析与设计

# 三、关键源码（部分）

Login.html

<body>

<form action="/day8\_shopping\_car/login" method="post">

姓名：<input type="text" name="username" /><br />

密码：<input type="password" name="password" /><br />

<input type="submit" value="登录" />

</form>

</body>

Buy.html

<body>

<form action="/day8\_shopping\_car/buy" method="post">

货品：<select name="product">

<option value="鼠标">鼠标</option>

<option value="光盘">光盘</option>

<option value="手机">手机</option>

</select><br />

数量：<input type="text" name="number" /><br />

<input type="submit" value="购买" />

</form>

</body>

Loginservlet.java

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("UTF-8");

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

String username = request.getParameter("username");

String password = request.getParameter("password");

HttpSession session = request.getSession();

User user = (User) session.getAttribute("user");

if (user == null

&& username != null && !"".equals(username)

&& password != null &&!"".equals(password)) {

user = new User();

user.setPassword(password);

user.setUsername(username);

session.setAttribute("user", user);

out.println("欢迎回来：<b>" + user.getUsername() + "</b> <a href='/day8\_shopping\_car/logout'>注销</a><br/>");

out.println("<a href='/day8\_shopping\_car/buy.html'>购物</a><br/>");

out.println("<a href='/day8\_shopping\_car/list'>管理购物</a>");

} else if (user != null && username == null && password == null) {

out.println("欢迎回来：<b>" + user.getUsername() + "</b> <a href='/day8\_shopping\_car/logout'>注销</a><br/>");

out.println("<a href='/day8\_shopping\_car/buy.html'>购物</a><br/>");

out.println("<a href='/day8\_shopping\_car/list'>管理购物</a>");

}else {

out.println("<a href='/day8\_shopping\_car/login.html'>重新登录</a>");

}

}

Logoutservlet.java

protected void service(HttpServletRequest request,

HttpServletResponse response) throws ServletException, IOException {

request.setCharacterEncoding("UTF-8");

response.setContentType("text/html;charset=UTF-8");

HttpSession session = request.getSession();

session.invalidate();

response.sendRedirect("/day8\_shopping\_car/login.html");

}

}

Listservlet.java

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("UTF-8");

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

HttpSession session = request.getSession();

Car car = (Car) session.getAttribute("car");

User user = (User) session.getAttribute("user");

if (user == null) {

out.println("还没登录。<a href='/day8\_shopping\_car/login.html'>请登录</a>");

return;

}

out.println("<b>" + user.getUsername() + "</b> 的购物车<br/>");

if (car != null) {

for (CarItem carItem : car.list()) {

out.println("商品：" + carItem.getProduct() + " 数量：" + carItem.getNumber() + " <a href='/day8\_shopping\_car/delete?id=" + carItem.getId() + "'>删除</a>");

out.println("<hr>");

}

} else {

out.println("还没购物车，所以没内容");

}

}

Deleteservlet.java

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("UTF-8");

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

HttpSession session = request.getSession();

User user = (User) session.getAttribute("user");

Car car = (Car) session.getAttribute("car");

String id = request.getParameter("id");

if (user != null) {

if (car != null && id != null && !"".equals(id)) {

car.remove(id);

out.println("删除成功<br/>");

out.println("<a href='/day8\_shopping\_car/buy.html'>继续购买</a><br/>");

out.println("<a href='/day8\_shopping\_car/list'>管理列表</a><br/>");

} else {

out.println("也还没车车...");

}

} else {

out.println("还没登录，禁止操作。<a href='/day8\_shopping\_car/login.html'>返回登录</a>");

}

}

}

Buyservlet.java

protected void service(HttpServletRequest request, HttpServletResponse response)

throws ServletException, IOException {

request.setCharacterEncoding("UTF-8");

response.setContentType("text/html;charset=UTF-8");

PrintWriter out = response.getWriter();

String product = request.getParameter("product");

Integer number = null; // 数量

try {

number = Integer.parseInt(request.getParameter("number"));

} catch (NumberFormatException e) {

out.println("数量非法，<a href='/day8\_shopping\_car/buy.html'>重新填写</a><br/>");

return;

//e.printStackTrace();

}

// 通过JSESSIONID,获取session对象，如果没有获取到，则新创建一个session对象

HttpSession session = request.getSession();

// 在session中获取user属性的对象

User user = (User) session.getAttribute("user");

// 在session中获取car属性的对象

Car car = (Car) session.getAttribute("car");

// 只有当用户已登录，且货品和数量不为空时，才允许添加项到购物车

if (user != null && product != null && number != null) {

CarItem carItem = new CarItem(); // 创建购物车项

// UUID.randomUUID().toString()，一个随机且不重复的字符串，方便购物车项的查询

carItem.setId(UUID.randomUUID().toString());

carItem.setProduct(product);

carItem.setNumber(number);

// 如果购物车为空，则创建一个购物车，并添加到Session

if (car == null) {

car = new Car();

session.setAttribute("car", car);

}

car.add(carItem);

out.println("购买成功<br/>");

out.println("<a href='/day8\_shopping\_car/buy.html'>继续购买</a><br/>");

out.println("<a href='/day8\_shopping\_car/list'>管理列表</a><br/>");

} else {

out.println("<a href='/day8\_shopping\_car/login.html'>请登录</a><br/>");

}

}

Car.java

public class Car {

// 用于存放CarItem（购物车项）的list

private List<CarItem> list = new ArrayList<CarItem>();

/\*\*

\* 获取购物车中的所有购物车项

\* @return 包含所有购物车项的List

\*/

public List<CarItem> list() {

return list;

}

/\*\*

\* 添加购物车项到购物车

\* @param carItem 需要添加的购物车项

\*/

public void add(CarItem carItem) {

this.list.add(carItem);

}

/\*\*

\* 从购物车中删除购物车项

\* @param id 需要删除的购物车项ID

\*/

public void remove(String id) {

for (int i = 0; i < list.size(); i++) {

if (list.get(i).getId().equals(id)) {

list.remove(i);

break;

}

}

}

Cartitem.java

public class CarItem {

private String id;

private String product;

private Integer number;

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getProduct() {

return product;

}

public void setProduct(String product) {

this.product = product;

}

public Integer getNumber() {

return number;

}

public void setNumber(Integer number) {

this.number = number;

}

User.java

public class User {

private String username;

private String password;

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

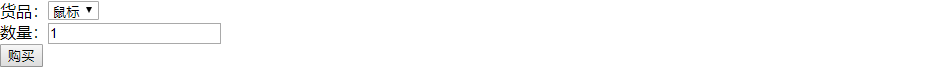
@Override

public String toString() {

return "User [" + username + ", " + password + "]";

}

# 四、运行效果图



# 收获与问题

学会使用会话管理