

安徽工业大学《web程序设计》实验报告

姓名 邹洋 班级 软171 学号 179074181

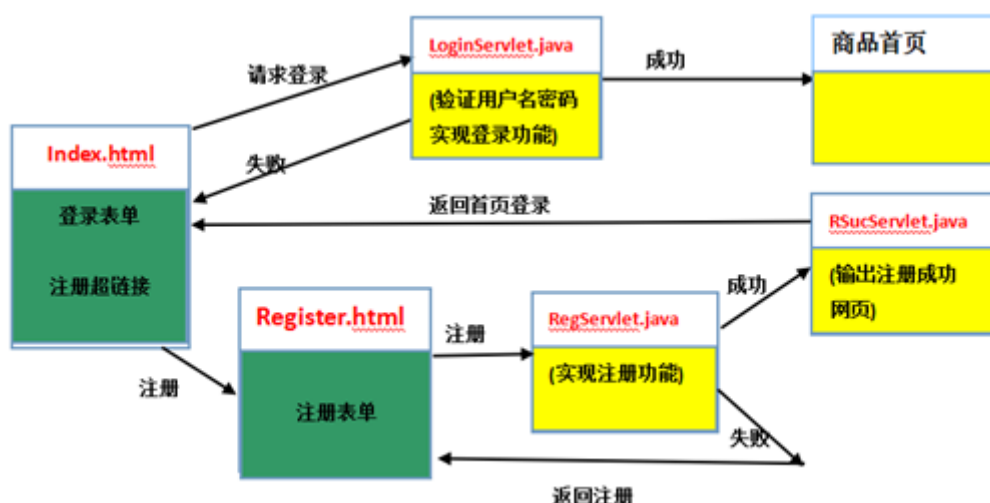
实验二~五

一、实验目的

- 1、掌握jdk +tomcat+eclipse的web开发环境的搭建与配置。
- 2、理解servlet技术中的概念，掌握servlet的编写。
- 3、熟悉jsp页面基本语法，九大内置对象的原理，并能运用他们解决实际问题。
- 4、掌握javaBean及其标签的使用方法。
- 5、掌握el表达式，jstl核心标签的使用。
- 6、理解mvc开发模型，掌握servlet+jsp+javaBean的开发模式。

二、实验内容及要求

1、实现系统的“会员注册”与“会员登录”功能。具体流程如下，用户可以在系统首页进行会员登录，或者是单击“会员注册”超链接，打开新会员注册页面，进行新会员的注册。如果注册失败，会显示相关失败原因。如果注册成功，则会显示注册会员的姓名与成功信息。注册成功的用户可以返回进行登录，如果登录失败，会回到首页重新登录；如果登录成功，则会进入会员功能页面。



要求:

- 1、注册和登录页面中前端可以用js代码验证用户名和密码是否为空，如果用户名或密码为空 则直接给出提示，不提交后台servlet处理。**注意：页面需要用css美化 and 布局。**
- 2、注册与登陆都需要操作数据库实现相应的功能。编写UserDao类，实现对用户表的查询和添加。登陆和注册的servlet类负责调用Dao层的相应方法实现功能。
- 3、登录功能模块利用session会话技术实现**验证码**功能（选做）。

(提示：此功能群文件里已上传CheckServ.java可以随机生成验证码返回到页面，并通过下面语句存到session里

```
session.setAttribute("checkcode", new String(rands)); )
```

4用jdbc技术连接数据库，实现商品的dao层编码，能对商品表进行查询。

5、登陆成功后，进入商品首页，显示所有商品信息。用javabean+el+jstl来显示。注意：**商品首页需要用css美化和布局。**

三、前台页面代码 (html、css、javascript)

index.jsp代码

```
<div>
  <div class="signin">
    <form class="" action="LoginServlet" method="post" onsubmit="return
checkForm(this);">
      <h1>登录</h1>
      <input type="text" name="name1" placeholder="用户名" class="tex1">
      <input type="text" name="passwd1"placeholder="密码" class="tex1">
      <input type="text" name="check1"placeholder="验证码" class="tex1">
      
      <input type="submit" value="登录" class="btn">
    </form>
    <a href="Register.jsp">注册账号</a>
  </div>
</div>
</body>
<script type="text/javascript">
  // 验证输入不为空的脚本代码
  function checkForm(form) {
    if(form.name1.value == "") {
      alert("用户名不能为空!");
      form.name1.focus();
      return false;
    }
    if(form.passwd1.value == "") {
      alert("密码不能为空!");
      form.passwd1.focus();
      return false;
    }
    return true;
  }
</script>
```

注册界面代码

```
<body>
  <div class="signin">
    <form action="RegServlet" method="post" onsubmit="return
checkForm(this);">
      <h1>注册</h1>
      <input type="text" name="name2" placeholder="用户名" class="tex1">
      <input type="text" name="passwd2"placeholder="密码" class="tex1">
      <input type="text" name="ypasswd"placeholder="再输一次密码"
class="tex1">
      <input type="submit" value="注册" class="btn">
    </form>
  </div>
</body>
```

```

        </form>
    </div>
</body>
<script type="text/javascript">
    // 验证输入不为空的脚本代码
    function checkForm(form) {
        if(form.name2.value == "") {
            alert("用户名不能为空!");
            form.name2.focus();
            return false;
        }
        if(form.passwd2.value == "") {
            alert("密码不能为空!");
            form.passwd2.focus();
            return false;
        }
        return true;
    }
</script>

```

css style美化代码

```

body{
    margin: 0;
    padding: 0;
    background: url(image/1.png);
}
.signin{
    background-color: rgba(120,120,120,0.1);
    width: 300px;
    padding: 20px;
    text-align: center;
    position: absolute;
    top: 50%;
    left: 20%;
    transform: translate(-50%,-50%);
    overflow: hidden;
}
.signin h1{
    margin-top: 10px;
    color: rgb(174, 228, 48);
    font-size: 35px;
}
.signin input{
    display: block;
    width: 100%;
    padding: 0 10px;
    height: 44px;
    text-align: center;
    box-sizing: border-box;
    outline: none;
    border: none;
}
.tex1{
    margin: 20px 0;
    background: rgb(255,255,255,.5);
    border-radius: 6px;
}

```

```

}
.btn{
    margin-top: 30px;
    margin-bottom: 20px;
    background: rgb(10, 82, 238);
    color: #fff;
    border-radius: 6px;
    cursor: pointer;
}
.btn:hover{
    transform: scale(0.95);
}
.signin a{
    text-decoration: none;
    color: #000;
    font-size: 14px;
    padding: 10px;
    display: block;
}
.signin a:hover{
    transform: scale(0.95);
}

```

商品界面代码

```

<div data-options="region:'north',border:false"
style="height:60px;background:#B3DFDA;padding:10px">
    <a style="display: block; text-align: center;font-size: 30px;">商城</a>
</div>
<div data-options="region:'west',split:true,title:'商品分类'"
style="width:150px;padding:10px;">
    <ul style="list-style-type:none">
        <li><a href="#">鞋子</a></li>
        <li><a href="#">衣服</a></li>
        <li><a href="#">手机</a></li>
        <li><a href="#">其他</a></li>
    </ul>
</div>
<div data-options="region:'east',split:true,collapsed:true,title:'公告'"
style="width:100px;padding:10px;">
    <a>热门新闻 :</a>
    <a>华为P30新</a>
    <a>NIKE活动</a>
</div>
<div data-options="region:'south',border:false"
style="height:50px;background:#A9FACD;padding:10px;">
    <a style="display: block; text-align: center;">Copyright © 2010 - 2019
Y.3.com 版权所有 | 消费者维权热线: 17353766138经营证照 | (京)网械平台备字(2018)第00003号 | 营业
执照</a>
</div>
<div data-options="region:'center',title:'商品展示'">
<%
    GoodsDao item=new GoodsDao();
    List<Goods> goods =item.selectAll();
    List<Goods> good =new ArrayList<Goods>();
    for( int i=0;i<goods.size();i++)

```

```

    {
        good.add(goods.get(i));
    }
    int n=goods.get(0).getId();
    session.setAttribute("id", n);
    /*int i=1;int n=goods2.getId();String
    str="id";str+=i;i++;session.setAttribute(str, n);*/
    %>
    <% for(Goods goods2:good){ %>
        <table style="float:left;">
            <tr style="valign:top;">
                <td rowspan="3"></td>
            </tr>
            <tr style="valign:top;">
                <td><b><a href="goodDetail.jsp"><%= goods2.getName() %></a></b>
</td>
            </tr>
            <tr style="valign:top;">
                <td><b><%= goods2.getPrice() %></b></td>
            </tr>
        </table>
    <%} %>
</div>

```

四、后台设计与源码

连接数据库Dao

```

public Connection getConnect() throws SQLException,ClassNotFoundException{
    //注册驱动
    Class.forName("com.mysql.cj.jdbc.Driver");
    String url="jdbc:mysql://localhost:3306/WEB?serverTimezone=UTC";
    String username="root";
    String password="haimian.";
    return DriverManager.getConnection(url,username,password);
}

```

用户Dao

```

public User() {}
    public User(String a,String b) {name=a;passwd=b;}
    public boolean correct(String name1,String passwd1) throws
SQLException,ClassNotFoundException{
        boolean F=false;
        ConnectionDB db =new ConnectionDB();
        Connection conn =db.getConnect();
        String sql ="select * from user where(name=? and passwd=?)";
        PreparedStatement pSt =conn.prepareStatement(sql);
        pSt.setString(1, name1);
        pSt.setString(2, passwd1);
        ResultSet rs =pSt.executeQuery();
        if(rs.next()) {
            F=true;
        }
        if(rs!=null)rs.close();
    }
}

```

```

        if(pst!=null)pst.close();
        if(conn!=null)conn.close();
        return F;
    }

```

注册Dao

```

public boolean option(String name2,String passwd2) throws
ClassNotFoundException, SQLException {
    boolean F=false;
    ConnectionDB db1=new ConnectionDB();
    Connection conn = db1.getConnect();
    String sql ="insert into user values(?,?)";
    PreparedStatement pst1=conn.prepareStatement(sql);
    pst1.setString(1, name2);
    pst1.setString(2, passwd2);
    int n=pst1.executeUpdate();
    if(n!=0) {F=true;}
    if(pst1!=null)pst1.close();
    if(conn!=null)conn.close();
    return F;
}

```

商品Dao

```

public List<Goods> selectAll() throws SQLException,ClassNotFoundException{
    ConnectionDB db =new ConnectionDB();
    Connection conn =db.getConnect();
    PreparedStatement pSt =null;
    ResultSet rSet =null;
    List<Goods>list=new ArrayList<Goods>();
    try {
        String sql ="select * from items";
        pSt =conn.prepareStatement(sql);
        rSet=pSt.executeQuery();
        while (rSet.next()) {
            Goods goods =new Goods();
            goods.setId(rSet.getInt(1));
            goods.setName(rSet.getString(2));
            goods.setCity(rSet.getString(3));
            goods.setPrice(rSet.getInt(4));
            goods.setNumber(rSet.getInt(5));
            goods.setPicture(rSet.getString(6));
            list.add(goods);
        }
    }
}

public Goods findById(int id) throws SQLException,ClassNotFoundException{
    ConnectionDB db =new ConnectionDB();
    Connection conn =db.getConnect();
    PreparedStatement pSt =null;
    ResultSet rSet =null;
    Goods goods =null;
    try {
        String sql ="select * from items where id=?";
        pSt =conn.prepareStatement(sql);
    }
}

```

```

        pst.setInt(1, id);
        rSet = pst.executeQuery();
        if(rSet.next()) {
            goods =new Goods();
            goods.setId(rSet.getInt("id"));
            goods.setName(rSet.getString("name"));
            goods.setCity(rSet.getString("city"));
            goods.setPrice(rSet.getInt("price"));
            goods.setNumber(rSet.getInt("number"));
            goods.setPicture(rSet.getString("picture"));
        }
    } catch (Exception e) {
        // TODO: handle exception
    }
    finally {
        if (rSet!=null) {
            rSet.close();
        }
        if (conn!=null) {
            conn.close();
        }
        if(pst!=null) {
            pst.close();
        }
    }
    return goods;
}

```

登录servlet

```

request.setCharacterEncoding("UTF-8");
String name =request.getParameter("name1");
String passwd =request.getParameter("passwd1");
String check =request.getParameter("check1");
HttpSession session=request.getSession();
String checkcode=(String) session.getAttribute("checkcode");
User uu =new User(name,passwd);
RequestDispatcher dis=null;
try {
    if(!checkcode.equalsIgnoreCase(check))
{dis=request.getRequestDispatcher("index.jsp");}
    else if(uu.correct(uu.getName(), uu.getPasswd())) {
        dis=request.getRequestDispatcher("goods.jsp");
    }
    else
        dis=request.getRequestDispatcher("index.jsp");
    dis.forward(request, response);

} catch (ClassNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
}

```

注册servlet

```

request.setCharacterEncoding("UTF-8");
String name =request.getParameter("name2");
String passwd =request.getParameter("passwd2");
Register reg = new Register(name,passwd);
RequestDispatcher dis1=null;
try {
    if(reg.option(reg.getName(), reg.getPasswd())) {;
        dis1=request.getRequestDispatcher("RsucServlet");
        dis1.forward(request, response);
    }
} catch (ClassNotFoundException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
} catch (SQLException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}

```

验证码servlet

```

HttpSession session=request.getSession();
response.setContentType("image/jpeg");
ServletOutputStream servletOutputStream=response.getOutputStream();
//设置浏览器不要缓存此图片
response.setHeader("Pragma", "No-cache");
response.setHeader("Cache-Control", "no-cache");
response.setDateHeader("Expires", 0);
//创建内存图片并获得其图形上下文
BufferedImage image=new BufferedImage(width, height,
BufferedImage.TYPE_INT_RGB);
Graphics graphics=image.getGraphics();
//产生随机的验证码
char[] rands=generCode();
//产生图像
drawRands(graphics, rands);
//结束图像的绘制过程，完成图像
graphics.dispose();
//将图像输出到客户端
ByteArrayOutputStream byteArrayOutputStream=new ByteArrayOutputStream();
ImageIO.write(image, "JPEG", byteArrayOutputStream);
byte[] buf=byteArrayOutputStream.toByteArray();
response.setContentLength(buf.length);
servletOutputStream.write(buf);
byteArrayOutputStream.close();
servletOutputStream.close();

//将当前的验证码写入到session中
session.setAttribute("checkcode", new String(rands));
}

private char[] generCode(){
    //定义验证码的字符表
    String chars="0123456789abcdefghijklmnopqrstuvwxyz";
    char[] rands=new char[4];
    for(int i=0;i<4;i++){
        int rand=(int)(Math.random()*36);
        rands[i]=chars.charAt(rand);
    }
}

```



```

    }
    return rands;
}
private void drawRands(Graphics g,char[] rands){
    g.setColor(Color.WHITE);
    g.setFont(new Font(null, Font.ITALIC|Font.BOLD, 18));
    //在不同高度上输出验证码的每个字符
    g.drawString(""+rands[0], 1, 17);
    g.drawString(""+rands[1], 16, 15);
    g.drawString(""+rands[2], 31, 18);
    g.drawString(""+rands[3], 46, 16);
}
}

```

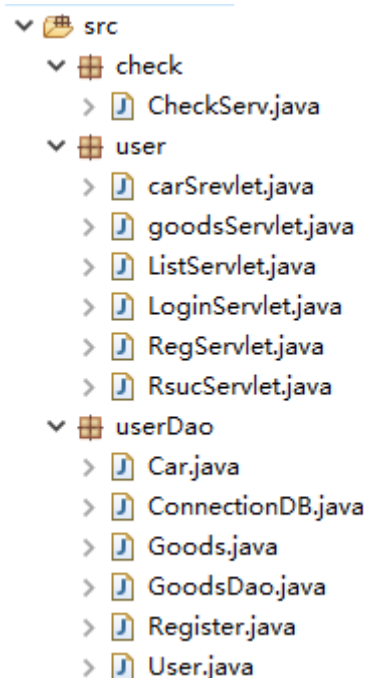
登录成功servlet

```

response.setContentType("text/html;charset=UTF-8");
request.setCharacterEncoding("UTF-8");
response.setCharacterEncoding("UTF-8");
PrintWriter out =response.getWriter();
out.println("<html>");
out.println("<head><meta charset=\"UTF-8\"><title>注册成功</title></head>");
out.println("<body>");
out.println("<a href=\"http://localhost:8080/JD/\" style=\"color: blue;\">注册成功 返回登录...</a>");
out.println("</body>");
out.println("</html>");

```

src截图



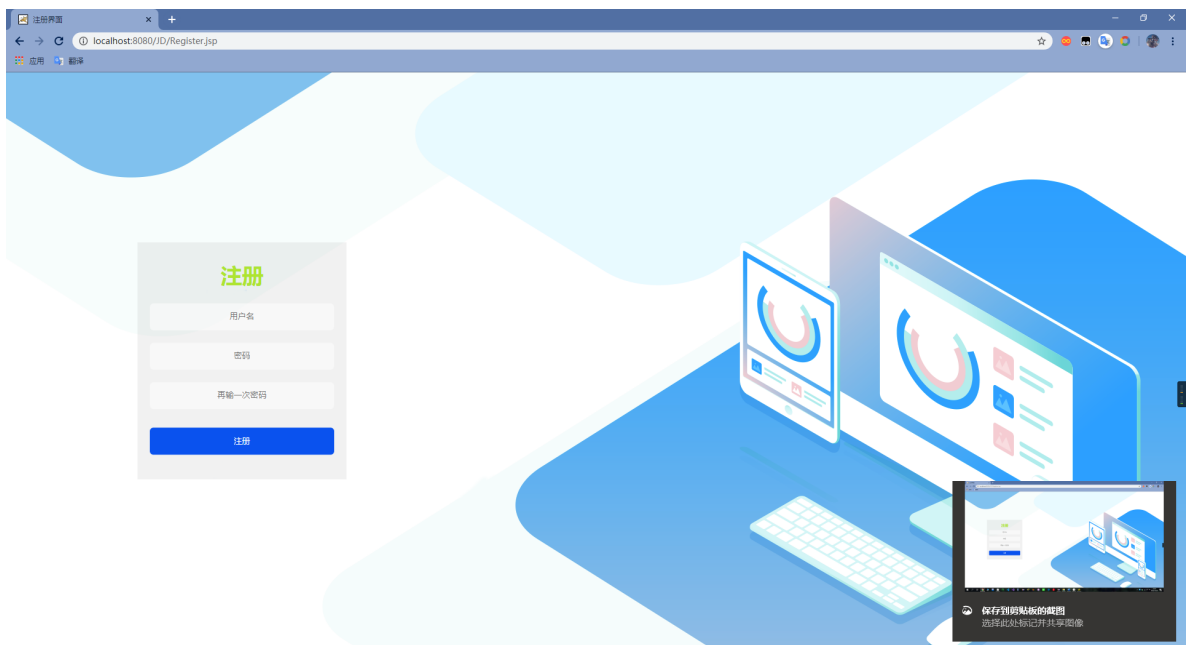
五、实验结果

此处附上系统运行页面截图、实验结果

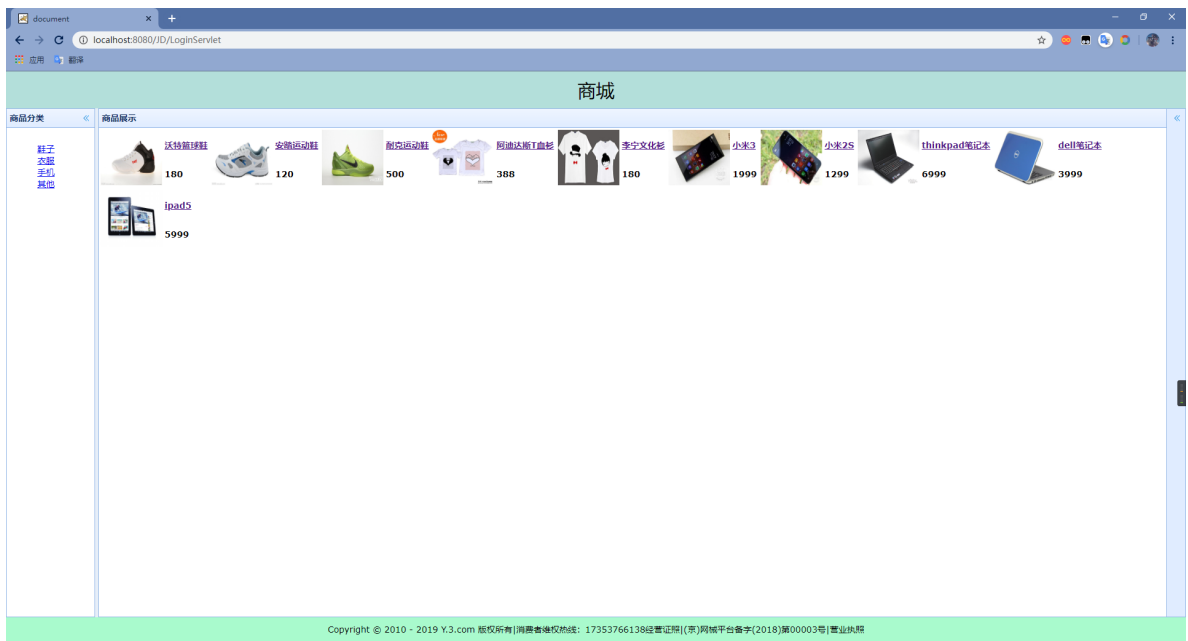
登录界面



注册界面



商品界面



六、实验总结

1. 验证码的实现

开始一直以为servlet产生一张图片存在session，结果发现直接在src写servlet地址就行，session中存的是产生的随机数。用于比较验证。

2. 对数据库商品的遍历

使用jstl比较生疏，花费大量时间在商品页商品的展示