华南理工大学

《网络应用开发》课程实验报告

实验题目： 设计并实现一个电子商务网站的开发和在线部署 \_\_

姓名： 黄鸿展 学号： 202230441138 \_

班级： 计算机科学与技术2班\_

提交日期： 2024-12-6

代码托管地址： <https://github.com/Mmmouse404/web-work>

部署在线网站： <http://8.134.196.79:8080/javaweb_mouse_war_exploded/>

测试账号口令： 用户账号:123456 密码:111 商家账号:1 密码:h1 管理密码:root

|  |
| --- |
| **实验概述** |
| 【实验题目】  设计并实现一个电子商务网站的开发和在线部署.  【基本功能要求】  顾客：  用户的注册、登录、注销  展示产品列表  购买流程（浏览/查询->添加至购物车->付款->发送电子邮件确认收货）  可以查看订单状态和历史  销售：  商品目录的管理（包括最基本的增删改等操作）  订单管理、以及销售统计报表  客户管理、以及客户的 浏览/购买 日志 记录  【实验环境和工具】  开发工具：IntelliJ IDEA 2024.1.6 数据库：MySQL 8.028  操作系统：Windows11， Alibaba Cloud Linux 3.2104 LTS 64位  Web服务器：阿里云ECS服务器(2核2GiB)，tomcat 8.599  浏览器：Microsoft Edge 版本131.0.2903.70(64位)  编程框架：Java Servlet后端、Jsp+ jQuery+ css前端、tomcat作为服务器 |
| **实验内容**   1. **系统设计：**   **1.1数据库设计：**  1.1.1 User表   |  |  |  |  | | --- | --- | --- | --- | | **属性名称** | **属性类型** | **是否可为空** | **是否为主键** | | ID | int | 否 | 是 | | NAME | varchar(100) | 否 | 否 | | PASSWORD | varchar(100) | 否 | 否 | | MONEY | Double | 否 | 否 | | EMAIL | varchar(100) | 时 | 否 |   1.1.2 Merchant表   |  |  |  |  | | --- | --- | --- | --- | | **属性名称** | **属性类型** | **是否可为空** | **是否为主键** | | merchantname | varchar(100) | 否 | 否 | | ID | int | 否 | 是 | | PASSWORD | varchar(100) | 否 | 否 |   1.1.3 Goodlists表   |  |  |  |  | | --- | --- | --- | --- | | **属性名称** | **属性类型** | **是否可为空** | **是否为主键** | | id | int | 否 | 是 | | goodname | varchar(100) | 否 | 否 | | price | double | 否 | 否 | | stock | int | 否 | 否 | | kind | varchar(100) | 否 | 否 | | merchantname | varchar(100) | 否 | 否 | | description | varchar(100) | 是 | 否 |   1.1.4 Cart表   |  |  |  |  | | --- | --- | --- | --- | | **属性名称** | **属性类型** | **是否可为空** | **是否为主键** | | username | varchar(100) | 否 | 否 | | goodname | varchar(100) | 否 | 否 | | price | double | 否 | 否 | | number | int | 否 | 否 | | address | varchar(100) | 否 | 否 | | merchantname | varchar(100) | 否 | 否 | | id | int | 否 | 是 |   1.1.5 Orders表   |  |  |  |  | | --- | --- | --- | --- | | **属性名称** | **属性类型** | **是否可为空** | **是否为主键** | | ID | int | 否 | 是 | | GOODNAME | varchar(100) | 否 | 否 | | ADDRESS | varchar(100) | 否 | 否 | | NUMBER | int | 否 | 否 | | PRICE | double | 否 | 否 | | MERCHANTNAME | varchar(100) | 否 | 否 | | STATUS | varchar(100) | 是 | 否 |   **1.2网站文件架构**  网站的基本文件架构图如下：    其中：  **1.2.1 target：**文件存放源代码编译之后的结果  **1.2.2 src/main：**存放文件源代码  **-java：**存放后端java代码  -action：存放servlet文件，负责处理网页请求并转交给service  -dao：设计需要用到的java类（数据结构），生成SQL语句，与数据库进行交互  -tools：设计UTIL.java，设定账户与密码连接数据库  **-webapp：**存放前端代码  -img：存放背景图、logo等图片  -js：ajax.js/json.js/ jquery-3.4.1.min.js支持处理ajax、json、jQuery  -css：各个网页样式的css文件，美化网页  -WEB-INF：web.xml设置web应用配置  -\*.jsp文件：各个网页的设计  **1.2.3 pom.xml**：配置Maven项目所用依赖项  **1.3网站网页架构**  网站网页架基本架构图如下：    **1.4基本功能实现**  功能实现基本流程：（功能实现逻辑大体相似，调用函数）  ①网页向Servlet发出请求  ②Servlet接收请求并转发给Dao  ③Dao中通过\*\_Use文件生成相应SQL语句与数据库进行交互并返回结果  ④Servlet得到结果，返回响应  ⑤网页显示对应结果  **1.4.1 注册、登录功能**  用户与商家注册登录逻辑大体相似，按下用户/商家注册按钮弹出输入框，通过会话设置账号、名字、密码、邮箱（仅用户有）信息，并根据所选登录角色的不同跳转到用户界面或者商家界面。  **1.4.2 用户功能**  ① 个人中心  左侧展示个人信息，可以进行个人信息的输入（邮箱、密码），输入新信息后点击“修改”即可修改个人信息。通过增加金额的拖动条可以增加自己的金额，同时还可以直接点击对应的按钮跳转到对应的商品、购物车、订单页面。  ② 推荐商品  通过数据库的goodlist表直接随机选出6件商品横向排列到栏上展示图片和商品单价，名称。右上角通过增加金额的拖动条可以增加自己的金额，点入商品图片即可跳转到“展示商品”具体展示的页面。点击“刷新”按钮可以重新刷新商品。  ③ 全部商品  通过数据库的goodlist表选择出所有商品，展示同推荐商品；有翻页功能，每页展示10件商品，点击商品图片即可展示商品的全部信息。通过增加金额的拖动条可以增加自己的金额。  ④ 展示商品  通过“推荐商品”或“全部商品”点入时会通过session传入Sid（商品编号），通过Sid查询对应商品的详细信息，以卡片信息展示到页面中。通过加减按钮可以更改购入物品数量，填写地址后点击“加入购物车”，通过操作数据库即可在cart中记录。  ⑤ 购物车  购物车展示了cart中的所有商品购物车记录，含有批量删除（复选框）功能。选中复选框后即可在右侧看到“待结算的商品”，点击下单即可通过购物车的商品编号等信息，由数据库操作创建订单。还可以修改地址和数量栏改变订单的情况。在下单旁边还要清空购物车，点击即可删去该用户购物车内容。  ⑥ 订单管理  展示了历史所有订单的情况（商品名，数量，总价，地址，用户名等），支持按地址和商品名进行查询。其中订单由“未发货”“已发货”“已收货”三个状态，如果订单状态为已发货即可按“收货”功能  **1.4.3 商家功能**  ① 商品管理  商家登录后，即可看到商品管理的页面，展示了全部商品，可以按分类和名字进行搜索。可以点击“新增商品”，完成对商品信息的上传以后，通过数据库操作成功上传商品。点击修改商品即可让商品信息变为输入框，输入新信息点击修改即可完成修改。删除物品可以通过复选框完成批量删除。  ② 订单管理  点击“菜单”-“商品管理”-“商品订单”即可看到该商家的商品订单所有情况。其中：“未发货”黄色，“已发货”为蓝色，“已收货”为绿色，“未发货”订单具有“发货”功能，点击发货即可修改订单状态，同时利用SMTP协议对用户的邮箱提供邮件发货提醒。可以通过复选框实现批量删除订单。  ③ 用户订单统计  “菜单”-“用户管理”-“用户订单日志”即可看到以用户进行查询的订单情况，统计了每个用户订单共计总消费额。而点击用户的表格即可看到每个订单的具体情况，包括是否发货，也可以在此页面实现发货。可以根据用户名或地址进行指定用户订单查询。  ④ 用户管理  “菜单”-“用户管理”-“用户信息”，输入“root”密码即可登录，查看用户的所有信息，包括密码（因此需要管理员密码root，和普通商家权限分离），可以新增用户，也可进行用户的批量删除。   1. 代码实现    1. **基本数据结构**   基本用到的java类都存在Dao中，数据结构与数据库的对应，varchar改为用String，包含了Get和Set方法，以User为例：    而User\_Use则包含了User类的使用方法，主要是以数据库的增、删、查、改操作为主。    涉及的数据结构与数据库基本类似，并以\*\_Use.java作为数据库操作方法类，下面给出各个数据结构：     * 1. **基本代码流程**   以商品加入购物车为例子讲解代码实现的基本流程：    首先通过jsp和css创建一个页面的输入框，通过Goodlist的Get方法得到对应商品的信息（图片，名字，价格等），通过input标签输入框引导用户输入地址，通过加号减号进行数量的删减。而下方的<script>脚本标签则保证了加减号更新时物品的购入数量改变，同时通过查询物品的库存限制加减号最多到库存上限，并计算购入的总金额，传入#good\_total标签，更新总价。  获取了购物的信息以后，通过addToCart()函数将所有的购物信息合并在一个parem中，并通过ajax方法向addToCartservlet传入Post请求，传入parem数据，并等待响应。    在Post请求后，Servlet响应请求并执行doPost()方法，从request里通过 getParameter方法得到各个属性，并调用Cart中的Set方法设定一个Cart对象，最后通过Cart\_Use中的AddToCart()方法执行数据库操作，插入一个Cart对象，最终通过write和setStatus()方法设置响应状态并写回内容，addToCart()函数接收响应并根据响应是否成功给出信息。      由于具体代码太多，且操作流程与上述相似，这里将给出其他关键代码的说明：   * 1. **关键功能代码说明**      1. **Dao**   User：用户类，User\_Use:用户操作类，可以根据用户id或名字在数据库中更新信息或删除，可以在数据库中添加新用户  Merchant：商家类，Merchant\_Use：商家操作类，可以根据商家的id找到对应商家的信息，可以添加/删除商家信息  Goodlist：商品类，包含商品姓名、单价、图片路径等信息，Goodlist\_Use:商品操作，可以进行商品的增删改查，主要以传入Id或Goodname作为查询参数，进行数据库操作。getRandomGoods()还可以随机查询对应数量的商品。  Cart：购物车类，主要存放加入购物车的信息，Cart\_Use：负责对应用户的购物车信息，可以根据Goodlist的Id进行增删改查，还可以向Order传递对应的用户名字、商品地址参数。  Order：订单类，主要收集用户信息、物品信息、商家信息（只收集主键及地址），其他部分通过调用其他Use类进行展示。可以根据User中的Email通过javax.mail中的方法发送邮件。   * + 1. **Action**   addGoodServlet：商家增加商品  addOrderServlet：用户增加订单（购物车结算）  addToCartServlet：指定商品加入到购物车  cartActionServlet：购物车操作，主要负责删除与清空购物车  goodBatchDeleteServlet：商家批量删除商品  increaseMoneyServlet：用户增加金额  loginServlet：用户登录  logoutServlet：用户与商家注销  merchantLoginServlet：商家登录  merchantRegisterServlet：商家注册  orderBatchDeleteServlet：批量删除订单（商家用户均可）  registerServlet：用户注册  setRootSessionServlet：商家访问用户信息，开启管理员权限  shipOrderServlet：修改订单状态（发货-收货）  updateGoodServlet：商家修改商品信息  updateUserInfoServlet：修改用户信息  userBatchDeleteServlet：用户批量删除   * + 1. **Tools**   UTIL.java:创建了一个Connection对象，加载好数据库驱动后连接云服务器的mysql数据库，通过给定的用户名登录并保持连接   * + 1. **Jsp**   index.jsp：索引界面，同登陆界面  register.jsp：注册界面，包含用户与商家注册  registersucc.jsp：注册成功，等待一秒跳转到登陆界面  login.jsp：登陆界面，包含用户登录与商家登录  loginFail.jsp：登陆失败，显示错误原因，等待一秒回到登陆界面  loginSuccess.jsp：登陆成功，等待疫苗直接显示商品信息  userProfile.jsp：个人中心界面，修改个人信息，可以跳转到其他界面  mainFrame.jsp：推荐商品界面，随机推荐6个商品  allShop.jsp：所有商品界面，可以翻页，可以按分类或名字搜索（模糊）  mouseShop.jsp：具体商品界面，显示商品信息，可以加入购物车或返回  cart.jsp：购物车界面，可以勾选商品修改数量地址下单、删除商品  myOrders.jsp：用户个人订单界面，修改订单状态、删除订单  manageGood.jsp：商品展示界面，可以跳转到商品的增删改查界面  insertGood.jsp：商品添加界面，填入信息后可以上传商品  editGood.jsp：修改商品界面，弹出所有商品，输信息即可修改，可以查询  manageOrder.jsp：订单管理界面，可以修改订单状态或查询、删除  manageUserOrder.jsp：用户订单统计页面，展示各个用户的订单详情，总金额，点击用户可以查看该用户的所有订单  manageUser.jsp：用户管理界面（仅管理员），登陆后展示用户信息，可以增加或删除用户信息  header.jsp：顶部栏，便于跳转各个界面  sidebar.jsp：侧边栏，设计菜单，可以跳转到各个界面   * 1. **代码部署**  1. 在github上注册一个账号      1. 新建仓库      1. 安装git <https://git-scm.com/>      1. 利用git初始化，上传代码  |  | | --- | | 先找到你放置项目的目录，打开git cd ["你放置项目的目录"] git init git add . git add README.md git commit -m “注释语句” git remote add origin https:*//github.com/zlxzlxzlx/Test.git* "https://github.com/zlxzlxzlx/Test.git" 为你对应的github仓库地址，可以在Code-Local-Clone中找到 git push -u origin master 如果冲突，可以将-u改为-f |  1. 可以在github中看到你上传的项目文件：     我的github项目链接：<https://github.com/Mmmouse404/web-work>   1. **功能测试：**    1. **用户**       1. **登录与注册**     首先进入登陆界面，显示用户登录与商家登录界面。由于未注册，我们先点击注册按钮，跳转到注册页面：    我们注册账号为123123，用户名为123，密码为123，[邮箱为cshhz@mail.scut.edu.cn](mailto:邮箱为cshhz@mail.scut.edu.cn)的用户，跳转到如下界面：      **如果输入密码错误，则会显示错误原因回到登陆界面：**    **浏览与搜索**  用户登录，如果成功则进入用户界面：    点击右上角的增加金额，拖动下方圆标，提交即可增加对应金额：    点击上方的”全部商品”，即可看到全部商品，可以翻页：    点入具体的商品图片，跳转到具体商品展示页面    在上方的搜索栏，可以输入名字/选择分类进行搜索（模糊搜索，返回所有）     * + 1. **选购与结算**   可以按加减调整数量（不超过库存），会得到对应总价。填写地址后即可加入购物车    点击“购物车”按钮，即可跳转到购物车界面。     * + 1. **下单与收货**   购物车中展示了选择的商品，可以修改数量地址，勾选后将显示下单商品：  按下结算按钮，即可创建订单，按下删除/清空购物车按钮，对应商品将在购物车中删除：    点击“订单”按钮，即可看到自己下的单：    如果订单状态有更新，显示为“已发货”，则会显示收货按钮，按下后将更新订单状态会从“已发货”改为“已收货”。   * + 1. **个人信息修改**   点击个人中心，即可进入个人中心界面，输入新邮箱/新地址即可进行修改。     * 1. **商家**      1. **登陆与注册**   注册流程同用户。我们注册账号为1234，用户名为好耗子，密码为1234的商家，注册成功后登录，跳转到如下界面：       * + 1. **商品管理（增删改查）**   点击菜单-商品管理-新增商品（或直接新增商品）即可跳转到增加商品页面，输入信息后即可添加商品。        商品管理界面可以根据分类和名字进行模糊搜索：    点击“修改商品”，将弹出所有商品信息，输入新信息，按下修改，即可修改。    勾选商品框，点击“批量删除”，即可删除对应商品。       * + 1. **订单管理**   **点击菜单-商品管理-商品订单，即可看到所有订单信息。**    为实现用户权限分离，将新增订单功能剔除，只留下批量删除。如果订单状态为“未发货”，将会弹出“发货”按钮，修改状态通知根据对应用户的邮箱发出商品发货邮件：         * + 1. **用户订单统计**   点击用户管理-用户订单日志，即可看到不同用户订单统计情况：    点击对应用户单元格，即可弹出详细用户订单，也可以发货：    也可以输入用户名进行模糊查询：     * + 1. **用户管理（管理员特权）**   点击用户管理-用户信息，会弹出登录界面：    输入登录密码（root），即可进入界面，得到用户的信息：  否则将会跳转到管理商品界面。    可以新增用户，也可以批量删除用户。（无法指定邮箱）   1. **应用部署**    1. **文件准备**   在idea中准备好项目工件（远程需要war expoloded）    可以在本地准备好数据库直接导出sql文件，以便云端恢复。     * 1. **服务器准备**      1. **服务器购买搭建**   **登录阿里云，进行学生认证：**  [**https://myaccount.console.aliyun.com/basic-info?open=student**](https://myaccount.console.aliyun.com/basic-info?open=student)  **完成学生认证后可以在此处领取代金券购买服务器：**  [**https://university.aliyun.com/**](https://university.aliyun.com/)      地区选择华南，操作系统我选用的是Alibaba Cloud Linux 3.2104 LTS 64位。  购买以后还需购买弹性公网IP    登陆以后需要通过远程连接，初次登录可能需要设置系统密码。       * + 1. **所需工具配置**   采用工具FileZella，通过远程端口21登录连接到操作系统， 以便文件传输：     1. **安装tomcat：**   **1：登录阿里云，用File Zilla连接阿里云**  **2：创建jdk存放的位置文件夹**      **cd /usr mkdir java**  **3：将jdk8-linux-64.tar.gz包上传到java文件夹中**      **4：解压jdk8-linux-64.tar.gz**  **tar -zxvf jdk8-linux-64.tar.gz**    **5：解压完成后将解压后的文件改名字方便使用**  **mv 解压出来的文件名字 要改成什么**  **mv jdk1.8.0\_212 jdk1.8**    **6：配置jdk的环境变量**  **vim /etc/profile**    **进入后 敲i进行编辑 加入下列四条语句**  **export JAVA\_HOME=/usr/java/jdk8**  **export CLASSPATH=$JAVA\_HOME/lib/**  **export PATH=$PATH:$JAVA\_HOME/bin**  **export PATH JAVA\_HOME CLASSPATH**  **（注：export JAVA\_HOME=jdk的安装路径）**  **保存并退出 ESC退出编辑；wq保存并退出文档**    **运行改过的文件即可**  **source /etc/profile**     1. **安装TomCat**   **1：连接服务器，连接 File Zilla（将压缩包传到linux服务器上）**  **2：创建tomcat存放的文件夹**  **cd /usr/local mkdir tomcat**    **3:用File Zilla将tomcat上传到服务器中**  **4：上传完成后解压《apache-tomcat-8.5.20.tar》并改名方便使用**  **/\*解压\*/ tar -zxvf apache-tomcat-8.5.20.tar**  **/\*改名\*/ mv apache-tomcat-8.5.20 tomcat8.5**      **4:配置 tomcat 并运行tomcat**  **//拷贝 cp -p /usr/local/tomcat/tomcat8.5/bin/catalina.sh /etc/init.d/tomcat**  **/\*编辑文本\*/ vim /etc/init.d/tomcat**               1. **安装MySQL**   1. 官网下载压缩包  官网地址：<https://downloads.mysql.com/archives/community/>  2. 解压后选取需要的包上传Linux    上传到路径：/usr/local/mysql  3. 按顺序下载（rpm）   |  | | --- | | cd /usr/local/mysql  ls -l  rpm -ivh mysql-community-common-8.0.32-1.el7.x86\_64.rpm  rpm -ivh mysql-community-client-plugins-8.0.32-1.el7.x86\_64.rpm  rpm -ivh mysql-community-libs-8.0.32-1.el7.x86\_64.rpm  rpm -ivh mysql-community-client-8.0.32-1.el7.x86\_64.rpm  rpm -ivh mysql-community-icu-data-files-8.0.32-1.el7.x86\_64.rpm  yum install net-tools #安装mysql-community-server前需要安装好的组件  rpm -ivh mysql-community-server-8.0.32-1.el7.x86\_64.rpm |   4.启动mysql   |  | | --- | | systemctl status mysqld//查看状态  systemctl start mysqld//启动  systemctl enable mysqld//开机自启  netstat -tunlp | grep mysql  # 查看Linux中已经启动的服务  netstat -tunlp  ps -ef | grep mysql//查看进程 |   5.登录mysql   |  | | --- | | cat /var/log/mysqld.log | grep password  得到的密码记得复制  # 登录mysql（回车后输入密码）【这里使用上面查询到的临时密码】  mysql -uroot -p  # 修改密码为123456  SHOW VARIABLES LIKE 'validate\_password%'; #查看密码安全设置  set global validate\_password\_policy=LOW; #设置密码安全等级为低  set global validate\_password\_length=4; #设置密码至少长度为4  ALTER USER 'root'@'localhost' IDENTIFIED BY '123456';  #开启访问权限：  # 创建用户root，密码123456  create USER 'root'@'%' IDENTIFIED BY '123456';  flush privileges; #刷新  #给用户授权  grant all privileges on \*.\* to 'root'@'%';  flush privileges; #刷新  # 这里就需要修改root密码，如果不修改，密码就会被改为password  ALTER USER 'root'@'localhost' IDENTIFIED BY '123456' PASSWORD EXPIRE NEVER;  # 密码与上一个命令保持一致  ALTER USER 'root'@'localhost' IDENTIFIED with mysql\_native\_password BY '123456';  flush privileges; #刷新  exit;  # 开启防火墙  systemctl start firewalld  # 开放3306端口  firewall-cmd --add-port=3306/tcp --permanent  # 刷新添加的端口  firewall-cmd --reload |  * + 1. **安全组设置**   **打开阿里云实例下方安全组-管理规则，设置安全组：**        **注意：3306端口连接数据库，请勿开放到各个ip，否则会被黑**   |  | | --- | | //停止防火墙 systemctl stop firewalld.service  //开启防火墙 systemctl start firewalld.service  //查看默认防火墙状态  （关闭后显示not running为红色字体，开启后显示running为白色字体） firewall-cmd --state  //使防火墙的80端口开放，允许公网访问  firewall-cmd --zone=public --add-port=80/tcp –permanent |        * 1. **应用部署到云端**  1. 开启linux上的tomcat服务器     2. 配置tomcat   |  | | --- | | # 进入tomcat安装路径中的bin文件夹下  cd /usr/local/apache-tomcat-9.0.65/bin  # vim打开catalina.sh进行编辑  vim ./catalina.sh  在334行左右有个# Execute The Requested Command，加入  CATALINA\_OPTS="-Dcom.sun.management.jmxremote -Dcom.sun.management.jmxremote.port=1099 -Dcom.sun.management.jmxremote.rmi.port=1099 -Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false -Djava.rmi.server.hostname=你的服务器的IP地址"  export CATALINA\_OPTS |   3. 配置远程tomcat（idea）    在这里插入图片描述在这里插入图片描述  在这里插入图片描述在这里插入图片描述  点击部署，即可部署成功：     1. **相关问题解决** 2. MySQL的大小写区分   Linux版本下的MySQL区分大小写，导致在可运行的SQL语句无效。这需要我们严格根据数据库中属性区分大小写重新构造SQL语句   1. 数据库被黑   数据库出现RECOVER\_YOUR\_DATA，原本数据库消失即为被黑，可以尝试重置，设置一个强的root密码，重置权限（除了root其他用户没有权限），将root作为本地可访问，设置个别的ip能连接的用户，用该用户进行登录。同时记得关闭3306端口的安全组，设置为只对指定ip（自己的ip）开放。   1. 远程部署后不更新   可以考虑在本地更新的target/out的webapp直接复制到远程服务器的webapp对应项目文件中（通过FileZilla）   1. 前面的部署出现问题   如果在前面的部署过程中出现问题，可能是因为教程比较简陋，缺少了部分文件的设置，或者是因为版本不同出现问题。可以搜索查看相关问题的解决方案。   1. **参考文献**   [在阿里云服务器上部署Tomcat详细图文详解-阿里云开发者社区](https://developer.aliyun.com/article/1485817)[⑩① 详解Linux安装 MySQL 8.0【保姆级教程】-阿里云开发者社区](https://developer.aliyun.com/article/1397961)  [Maven远程部署tomcat-阿里云开发者社区](https://developer.aliyun.com/article/1323629)  [最详细完整，使用idea远程部署Tomcat（包括一些注意事项和不容易注意到的坑）\_idea部署远程tomcat-CSDN博客](https://blog.csdn.net/qq_46690168/article/details/127097281) |
| **小结** |
| 通过设计一个网站从搭建到本地测试、完善、部署的整个过程，我进一步加深了对于网络应用开发的认识。这个任务对曾经不熟悉java与网络编程的我来说难度颇高，在整个开发的过程中我学会了网络编程servlet与jsp、js、css等技术的许多基础知识，问题调试debug的过程中也是让我对servlet与数据库有了进一步的了解，在这个过程中我的数据库遭到攻击，成功恢复后让我认识到了网络安全防范的重要新，而云服务器的环境搭建让我对linux系统和阿里云服务器有了更全面的认识，操作变熟悉了。得益于人工智能技术的发展，同学们的互相帮助和老师的指导，我最终如期完成了任务。 |
| **指导教师评语及成绩** |
| 评语：  成绩：    指导教师签名：                                                  批阅日期： |