|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **系统设计报告**   1. 整体框架设计   首先说明一下本网站所使用的开发环境   1. 操作系统： Windows 11 2. Web服务器：Tomcat 8.5.73 3. Java开发包：java 17 4. 数据库：MySQL 8.0.29 5. 开发工具：IDEA集成开发工具 6. 浏览器：Edge/Google Chrome   在IDEA开发环境中，我选择使用maven框架来组织自己的代码结构，并在项目中添加Web模块，这样才可以使用tomcat等进行后续功能的实现，整体上，在处理数据库方面，使用mybatis来进行持久层的处理。  在后端代码部分，具体分包如下，mapper包保存对数据库进行处理的接口代码，pojo包中包含着项目中涉及到的javaBean，service包提供各种服务，将数据库的mapper操作封装成函数方便调用，util包中包含各种工具，比如验证码生成，激活码生成，邮件发送，sqlSessionFactory的创建和调用等，web包中包含servlet包，里面包含着各种servlet代码，这里我改变了原始的一个后端功能对应一个servlet的实现方式，这种方式下的servlet文件会显得非常的多，而且不利于管理，我的做法是重写一个BaseServlet继承HttpServlet，在这里实现根据请求的最后一段路径分发，然后让其他servlet继承即可。    在resources文件中，包括mybatis的配置文件，以及mapper所对应的xml文件    对于所有的html文件统一放在webapp文件夹中，而如果需要导入jar包，直接在pom.xml中书写该jar包的名称和版本即可，通过maven来管理，既简便又高效。  在前端的处理方面，我选用ajax技术来进行异步请求，用vue+element来实现前端页面的设计与交互功能的实现。  项目的整体结构如下：    总体的设计的功能框架如下：         1. 数据库设计   这里先放出数据库表的模型      这个项目数据库表一共是五个，分别是user用户表，product商品表，orders订单表，orderItem订单项表，cartItem购物车项表。  User表   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 字段名 | 数据类型 | 是否主键 | 是否外键 | 默认值 | 描述 | | Id | int(11) | √ |  | auto\_increment | 主键id自增长 | | Username | varchar(50) |  |  | Not null | 用户名 | | Password | varchar(50) |  |  | Not null | 密码 | | Name | varchar(50) |  |  | Not null | 用户昵称/销售人员店铺名 | | Gender | Varchar(2) |  |  | null | 性别 | | Email | varchar(50) |  |  | Null | 邮箱地址 | | Telephone | varchar(20) |  |  | Null | 电话号码 | | Address | varchar(100) |  |  | Null | 地址 | | Introduce | varchar(150) |  |  | Null | 简介 | | activeCode | varchar(50) |  |  | Null | 注册激活码 | | state | Int(11) |  |  | 0 | 用户状态 | | role | varchar(6) |  |  | “user” | 用户类型 | | registerTime | Timestamp |  |  | Not null | 注册时间 |   Product表   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 字段名 | 数据类型 | 是否主键 | 是否外键 | 默认值 | 描述 | | Id | int(11) | √ |  | auto\_increment | 主键id自增长 | | name | varchar(255) |  |  | null | 商品名 | | description | varchar(511) |  |  | null | 描述 | | price | double |  |  | null | 价格 | | category | varchar(10) |  |  | null | 种类 | | categoryName | varchar(10) |  |  | Null | 种类名称 | | pnum | int(11) |  |  | Null | 商品数量 | | imgurl | varchar(100) |  |  | Null | 商品图片 | | saleId | int(11) |  | √ | 2 | 销售人员id | | ordered | Int |  |  | 1 | 商品优先级 | | state | Int |  |  | 0 | 商品状态 | | Time | timestamp |  |  | Not null | 商品上一次被修改或添加的时间 |   Orders表   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 字段名 | 数据类型 | 是否主键 | 是否外键 | 默认值 | 描述 | | Id | Varchar(50) | √ |  | Not null | 主键id | | money | double |  |  | null | 商品价格 | | receiverAddress | varchar(255) |  |  | null | 收货地址 | | receiverName | varchar(20) |  |  | null | 收货人姓名 | | receiverPhone | varchar(20) |  |  | null | 收货人电话 | | payState | Int(11) |  |  | 0 | 支付状态 | | orderTime | timestamp |  |  | Not null | 订单生成时间 | | userId | Int(11) |  | √ | Not null | 订单用户id |   OrderItem表   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 字段名 | 数据类型 | 是否主键 | 是否外键 | 默认值 | 描述 | | orderId | Varchar(50) |  | √ | Not null | 关联订单id | | productId | int(11) |  |  | null | 商品id | | buyNumber | int(11) |  |  | null | 购买数量 |   CartItem表   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 字段名 | 数据类型 | 是否主键 | 是否外键 | 默认值 | 描述 | | userId | int(11) | √ | √ | Not null | 主键id | | productId | int(11) | √ | √ | Not null | 商品id | | num | int |  |  | null | 商品数量 | | time | timestamp |  |  | Not null | 购物车商品生成时间 |  1. 各个模块的具体实施方案   下面以实现全部商品的展示为例子，简要说明各个模块的实现方法   1. 首先是mybatis中的mapper文件，写出查询全部商品的接口和sql语句，如图所示      1. 然后在service层，编写使用mapper的服务，来查询数据库   首先是service接口    然后是该service实现类     1. Servlet层书写后端代码      1. 写前端ajax代码      1. 用products接收后，显示在页面上     这样，就完成了一个查询所有的功能，其他功能同理。 |