****

**本科学生毕业论文（设计）**

题目 基于Django框架的电商网站

系别 电子工程系

专业 电子信息工程

学生姓名 熊孝如

学号 16510501084

指导教师 张辉 职称 讲师

论文字数

完成日期 年 月 日

**基于Django框架的电商网站（改4）**

熊孝如,电子工程系

摘 要:Django是高水准的Python编程语言驱动的一个开源模型、视图、控制器风格的Web应用程序框架，现在已经成为主流的Web应用框架之一，其流行程度很高，其特点是开发的速度快，成本较低。爱鲜蜂网站是基于Django框架设计开发的电商网站，其为用户呈现了清晰明了的商品选项，用户可以在网站上自由的选择喜爱的商品，方便了用户在线选择高质量的商品节约了用户的时间。本网站的设计采用了Python程序语言、Django、JS等技术对网站各个页面和后端接口进行设计与实现

关键词:Django;Python;电商网站

1. **commerce website based on Django framework**

Xiong xiaoru, department of electronic engineering

**Abstract :**Django is an open source model, view, controller style Web application framework driven by the high-level Python programming language. It has become one of the mainstream Web application frameworks with a high popularity, characterized by fast development speed and low cost.Aixianfeng website is an e-commerce website designed and developed based on Django framework, which presents clear and clear commodity options for users. Users can freely choose their favorite commodities on the website, which is convenient for users to choose high-quality commodities online and saves users' time.The design of this website USES Python programming language, Django, JS and other technologies to design and implement each page and back-end interface of the website

**Key words:** Django;Python;E-commerce sites

**安徽师范大学皖江学院毕业论文（设计）评定意见**

|  |  |
| --- | --- |
| 指导  教师  评语 | 成绩：  签名： 年　月　日 |

评语主要内容包括：学生写作态度、科研作风，论文选题的理论意义和实践价值，论据是否充分、可靠，掌握基础理论、专门知识、研究方法和技能的水平，写作的逻辑性、技巧及其他优缺点。

**安徽师范大学皖江学院毕业论文（设计）评定意见**

|  |  |
| --- | --- |
| 答  辩  组  评  定  意  见 | 成绩：  答辩组组长签名：  年　　月　　日 |
| 系  意  见 | 成绩：  系主任签章：    　　　　　　　　　　　　　　　　年　　月　　日 |

# **目 录**

**[1 概述 6](#_Toc8039)**

[1.1项目研究背景 6](#_Toc22976)

[1.2项目研究的意义 6](#_Toc11790)

**[2 网站系统分析 6](#_Toc4038)**

[2.1网站系统功能设计 6](#_Toc1370)

[2.2电商网站的ORM对象关系映射 7](#_Toc21690)

[2.3网站设计实现所需的软硬件平台 7](#_Toc12659)

**[3 构建网站的准备工作 7](#_Toc25203)**

**[4 电商网站具体设计 8](#_Toc31100)**

[4.1首页模块的构建 6](#_Toc12672)

[4.2闪购模块的构建 10](#_Toc12672)

[4.3登陆/注册页面的构建 12](#_Toc22330)

[4.4我的模块的构建 13](#_Toc18170)

[4.5购物车模块的构建 14](#_Toc13780)

[4.6订单模块的构建 15](#_Toc22013)

**[5 总结 17](#_Toc17166)**

**[参考文献 1](#_Toc17166)8**

**[致谢 1](#_Toc17166)9**

**1 概述**

**1.1项目研究背景**

进入二十一世纪以来，互联网的飞快发展推动着各行业的进步和发展。人类进入了互联网信息时代，人们对互联网数据和信息的处理和应用已经进入自动化、网络化和社会化的阶段。在互联网信息时代的今天，越来越多的互联网用户利用手机等移动端进行相关的网上交流和娱乐活动。网络购物成为了人们的重要活动，此种情况下，传统的线下购物已经不能满足人们随时随地购物的需求了，于是，电商网站应运而生，人们可以利用互联网来随时随地购入自己喜爱的东西。

**1.2 项目研究的意义**

我国电商的快速发展，催生了许多细分业务，Axf项目就是为了解决人们在购买新鲜食材上的困扰。人们可以在网页上购买自己喜爱的食材和水果，而快速的送货服务可以解决人们担忧的食材新鲜度的问题。

**1.3 电商网站开发软件介绍**

1.3.1 Python语言介绍

Python是一种跨平台的[计算机程序设计语言](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1%E8%AF%AD%E8%A8%80/7073760" \t "https://baike.baidu.com/item/Python/_blank)。 是一个高层次的结合了解释性、编译性、互动性和面向对象的脚本语言。最初被设计用于编写自动化脚本(shell)，随着版本的不断更新和语言新功能的添加，越多被用于独立的、大型项目的开发。

1.3.2 Django框架介绍

Django是高水准的Python编程语言驱动的一个开源模型．视图，控制器风格的Web应用程序框架，它起源于开源社区。使用这种架构，程序员可以方便、快捷地创建高品质、易维护、数据库驱动的应用程序。

1.3.3 MySQL数据库介绍

[MySQL](https://baike.baidu.com/item/MySQL/471251" \t "https://baike.baidu.com/item/MySQL%E6%95%B0%E6%8D%AE%E5%BA%93/_blank)是一种[开放源代码](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%BA%90%E4%BB%A3%E7%A0%81/114160" \t "https://baike.baidu.com/item/MySQL%E6%95%B0%E6%8D%AE%E5%BA%93/_blank)的关系型[数据库管理](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86" \t "https://baike.baidu.com/item/MySQL%E6%95%B0%E6%8D%AE%E5%BA%93/_blank)系统（RDBMS），使用最常用的数据库管理语言--[结构化查询语言](https://baike.baidu.com/item/%E7%BB%93%E6%9E%84%E5%8C%96%E6%9F%A5%E8%AF%A2%E8%AF%AD%E8%A8%80/10450182" \t "https://baike.baidu.com/item/MySQL%E6%95%B0%E6%8D%AE%E5%BA%93/_blank)（SQL）进行数据库管理。MySQL是开放源代码的，因此任何人都可以在General Public License的许可下下载并根据个性化的[需要](https://baike.baidu.com/item/%E9%9C%80%E8%A6%81/9109492" \t "https://baike.baidu.com/item/MySQL%E6%95%B0%E6%8D%AE%E5%BA%93/_blank)对其进行修改。MySQL因为其速度、可靠性和适应性而备受关注。大多数人都认为在不需要[事务](https://baike.baidu.com/item/%E4%BA%8B%E5%8A%A1" \t "https://baike.baidu.com/item/MySQL%E6%95%B0%E6%8D%AE%E5%BA%93/_blank)化处理的情况下，MySQL是管理内容最好的选择。

**2 网站系统分析**

**2.1 网站系统功能设计**

主要分为两个模块，一个是展示模块，负责展示商品信息，购物车以及账号信息。另一个是注册登录模块，负责注册和登录账号，把账号储存到数据库中，并在登录时与数据库里的内容对比检验，判断是否一致。

前端页面主要有六个，展示模块一共有四个，分别是首页、闪购、购物车、我的四个页面，首页闪购负责把Django的视图函数中从数据库中提取出来的数据显示在前端页面中，并给其添加样式。购物车页面负责显示选中的商品信息，并把商品做成一个订单。最后我的页面负责显示登陆账号信息。

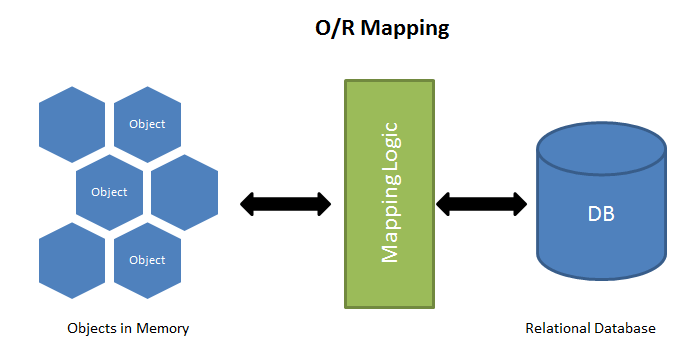
登陆注册模块主要有两个页面，一个是登录页面，一个是注册页面，注册页面把输入的账号信息储存到数据库中，登录页面把输入的信息与数据库中数据对比，判断是否一致，如果一致，在我的页面显示，如果不一致，在页面中提示输入错误。



**图1 电商网站的结构图**

**2.2电商网站的ORM对象关系映射**

Django中MVC框架中包括一个重要的部分，就是ORM，它实现了数据模型与数据库的解耦，即数据模型的设计不需要依赖于特定的数据库，通过简单的配置就可以轻松更换数据库。



**图2 ORM对象关系映射**

**2.3 网站设计实现所需的软硬件平台**

（1）硬件系统：一台搭建 Windows 10 系统的计算机。

（2）软件平台：Python3.6.8,Django2.22,Virtualenv,MySQL，Pycharm专业版

**3 构建网站的准备工作**

在构建一个电商网站之前，我们需要先构建一个Django工程，在Windows环境下首先创建虚拟环境，在命令行输入virtualenv envname创建一个名字为envname的虚拟环境，进入虚拟环境所在的Scripts文件夹输入命令 activate即可启动虚拟环境，在虚拟环境下安装Django指定版本。

上述内容完成后，输入命令django-adming startproject Axf创建一个Django工程，此Django工程名字叫Axf,这样就完成了准备工作。

**4 电商网站具体设计**

**4.1首页模块的构建**

4.1.1首页项目的创建

Django中创建项目的命令,在Pycharm上的命令行中输入python manage.py startapp HomeApp创建首页项目。

4.1.2首页部分的模型创建

此处模型创建我们使用的是MySQL数据库，MySQL数据库社区版免费且好用，可以满足我们的使用需求。

模型的创建我们可以在IDE中完成，此处我们一直使用Pycharm，在Pycharm中找到HomeApp然后打开它的models.py文件创建模型。

我们在HomeApp也就是首页中构建模型的字段主要是首页要展示的商品图片、商品名字、商品价格等等。

创建模型完成后，我们在Pycharm命令行中输入python manage.py makemigrations然后输入python manage.py migrate 完成模型的迁移，即把模型迁移到MySQL数据库的指定数据库中。

// 首页部分的模型代码

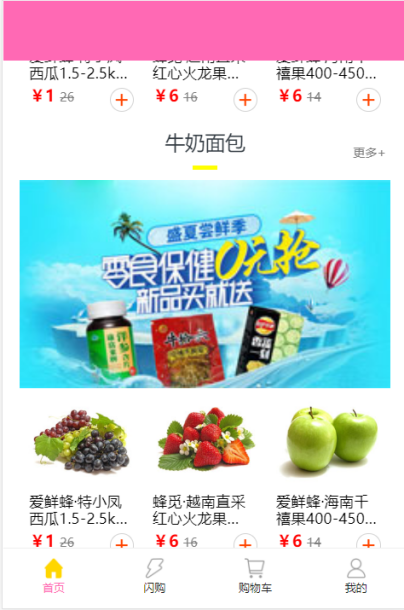
**class** AxfWheel(models.Model):  
 img = models.CharField(max\_length=256)  
 name = models.CharField(max\_length=32)  
 trackid = models.IntegerField()  
  
 **class** Meta:  
 db\_table = **'axf\_wheel'  
  
  
class** AxfNav(models.Model):  
 img = models.CharField(max\_length=256)  
 name = models.CharField(max\_length=32)  
 trackid = models.IntegerField()  
  
 **class** Meta:  
 db\_table = **'axf\_nav'  
  
  
class** AxfMustBuy(models.Model):  
 img = models.CharField(max\_length=256)  
 name = models.CharField(max\_length=32)  
 trackid = models.IntegerField()  
  
 **class** Meta:  
 db\_table = **'axf\_mustbuy'  
  
class** AxfMainShow(models.Model):  
 img = models.CharField(max\_length=256)  
 name = models.CharField(max\_length=32)  
 trackid = models.IntegerField()  
 categoryid = models.IntegerField()  
 brandname = models.CharField(max\_length=32)  
 img1 = models.CharField(max\_length=256)  
 childcid1 = models.IntegerField()  
 productid1 = models.IntegerField()  
 longname1 = models.CharField(max\_length=256)  
 price1 = models.IntegerField()  
 marketprice1 = models.IntegerField()  
 img2 = models.CharField(max\_length=256)  
 childcid2 = models.IntegerField()  
 productid2 = models.IntegerField()  
 longname2 = models.CharField(max\_length=128)  
 price2 = models.IntegerField()  
 marketprice2 = models.IntegerField()  
 img3 = models.CharField(max\_length=256)  
 childcid3 = models.IntegerField()  
 productid3 = models.IntegerField()  
 longname3 = models.CharField(max\_length=256)  
 price3 = models.IntegerField()  
 marketprice3 = models.IntegerField()  
 **class** Meta:  
 db\_table = **'axf\_mainshow'**

4.1.3 将数据库中的数据返回到HTML页面中

我们打开HomeApp，接着打开views.py文件，写上返回数据给前端的代码段,return render()即把MySQL数据库中储存的数据返回到前端页面给与显示。

4.1.4首页的轮播展示

首页的轮播展示的功能，可以为我们简介展示我们的商品 ，此处我们需要借助swiper.css以及swiper.js第三方插件。我们在home.html中引入swiper.js和swiper.css，然后激活swiper，并向其添加分页器功能和轮播功能。



**图3 首页展示图**

**4.2 闪购模块的构建**

4.2.1闪购项目的准备工作

跟我们的首页页面一样，在我们完成闪购页面之前我们需要把闪购页面搭建起来，跟首页项目一样，我们在Pycharm命令行输入python manage.py startapp MarketApp 。这样，我们就搭建好了闪购页面的项目，之后的代码书写我们就可以在MarketApp中来书写了。

4.2.2闪购项目的模型构建

因为我们闪购页面的HTML页面要显示左侧导航条，上侧导航栏还有右下角商品界面的显示，这其中包括了商品名、商品价格、商品图片、商品类型、商品id等等。因此，我们构建了商品类型表与商品表两个模型来完成闪购页面的显示。

//闪购项目的模型代码

**class** AxfFoodType(models.Model):  
 typeid = models.CharField(max\_length=256)  
 typename = models.CharField(max\_length=256)  
 childtypenames = models.CharField(max\_length=256)  
 typesort = models.IntegerField()  
  
 **class** Meta:  
 db\_table = **'axf\_foodtype'  
  
class** AxfGoods(models.Model):productid = models.CharField(max\_length=10)productimg = models.CharField(max\_length=150)productname = models.CharField(max\_length=50)productlongname = models.CharField(max\_length=100)isxf = models.NullBooleanField(default=**False**)pmdesc = models.CharField(max\_length=10)specifics = models.CharField(max\_length=20)price = models.CharField(max\_length=10)marketprice = models.CharField(max\_length=10)categoryid = models.CharField(max\_length=10)childcid = models.CharField(max\_length=10)childcidname = models.CharField(max\_length=10)dealerid = models.CharField(max\_length=10)storenums = models.IntegerField()productnum = models.IntegerField()  
  
 **class** Meta:  
 db\_table = **'axf\_goods'**

4.2.3闪购页面左侧导航栏的构建

我们在商品类型表中插入我们需要的数据，然后我们在闪购项目的views.py文件中将商品类型表，即GoodsType模型实例化，然后把其所有的数据通过return render()返回给前端页面中，前端页面接收并显示。前端的HTML页面接收数据时要根据页面的css布局将商品类型显示出来，这样闪购页面的左侧导航栏就完成了。

4.2.4闪购页面上侧导航栏的完成

此处页面的显示与左侧导航栏的完成一样，也是拿出我们模型中的数据然后将他们显示在前端的HTML页面上，不过我们要注意的是我们在后端接口的书写，我们要分离模型字段，然后在前端页面显示上侧导航栏的内容。

4.2.5闪购页面右下角商品页面的完成

在完成商品界面之前，我们需要构建商品的模型，这其中包含了商品的ID、商品名、商品价格、商品图片等等。

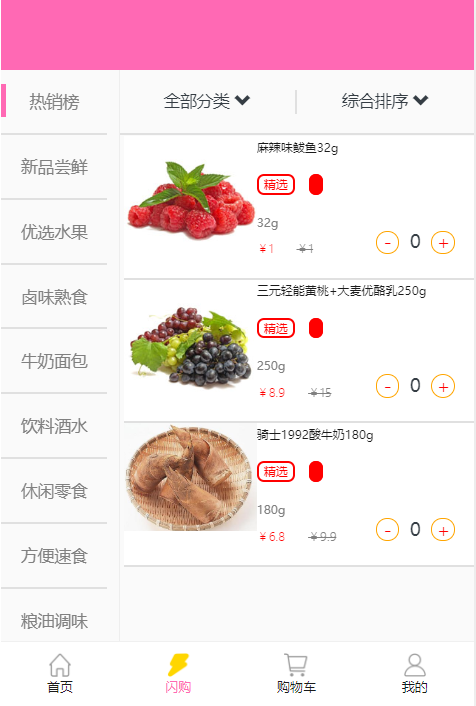
完成了商品模型之后，我们跟之前一样在后端接口中调出商品名、商品图片的外链等等，再返回给前端页面，然后让前端页面显示出来。

4.2.6商品侧边栏的高亮显示

此步骤需要一定的代码逻辑，当我们点击商品侧边栏上的分类时， 我们需要让其显示高亮，这时我们需要在前端页面中让我们的书写 相应的JQuery和css等文件。css让页面具有样式而JQuery让页面 点击具有反应。而我们让侧边栏拥有高亮显示还需要在前端页面中 添加URL反向解析，例如闪购页面左侧导航栏的高亮显示，我们需 要让typeid和类型表中的typeid相等，此时显示高亮。

4.2.7根据价格或销量排序

此功能我们依旧需要调用商品表即Goods表中的数据，将Goods表 实例化后，在后端接口用order\_by函数将Goods表中的按price字 段或num字段排序，我们既能完成用价格或销量来对商品进行排序。 之后我们再把排序后的数据在前端页面显示出来。



**图4 闪购模块展示图**

**4.3 登陆/注册页面的构建**

4.3.1 注册页面的准备工作

注册页面的构建是为了后面我的页面的搭建做了一个准备工作, 我

们构建完成注册页面之后，我们就可以在用户模型中，拿出用户的头像，姓名等等。

跟之前一样，我们要先构建用户项目，输入命令python manage.py startapp UserApp 这样我们就构建好了用户项目。

4.3.2注册页面需要的用户资料的模型

用户资料的模型，我们需要的是用户的姓名、头像图片、密码和电子邮件等等，因此我们在UserApp的models.py文件中构建出用户模型，然后在命令行输入python manage.py makemigrations和python manage.py migrate迁移模型至数据库。

//用户资料模型代码

**class** AxfUser(models.Model):  
 u\_name = models.CharField(max\_length=128)  
 u\_password = models.CharField(max\_length=256)  
 u\_email = models.CharField(max\_length=64)  
 u\_icon = models.ImageField(upload\_to=**'icons'**)  
 u\_token = models.CharField(max\_length=256)  
  
 u\_active = models.BooleanField(default=**False**)  
  
 **class** Meta:  
 db\_table = **'axf\_user'**

4.3.3 注册功能的初步完成

我们在后端的用户接口判断是GET还是POST请求，GET请求我们暂且不管，当请求时POST时，我们把页面POST请求的数据通过request.POST.get()来获取，然后再用实例化的用户模型来保存数据，即用User.sava()来保存数据至数据库。

4.3.4注册页面的完善

现在我们的注册页面已经初步构建好了，但是还是有一些BUG。例如当我们注册时如果用户名重复的话就不好了，因此我们要在用JS来判断用户名是否重复以及两次输入的密码是否一致。

我们先在后端接口写出检查姓名的接口，然后在JS中利用getJson来先分析接口URL，然后再把数据在前端接口显示出来，密码也是如此，我们在后端接口写出检验密码的接口，然后利用JS显示出来，这样，我们就完成了注册页面的完善。

4.3.5登录界面的HTML页面构建

登录界面的构建不需要多少的逻辑，我们直接在静态文件夹下写出login.html即登录页面，出来必须的姓名和密码验证之外，我们还可以添加验证码验证，在百度搜索验证码的代码即可。

4.3.6 登录界面的后端接口的完成

在前面注册界面的书写中，我们暂时没有写出GET请求的内容，这

时，我们可以写成GET请求的内容，如果是GET请求我们返回页面至login.html即登录页面。

登录页面的后端接口书写，我们主要验证用户是否在数据库中存在、用户输入的密码是否正确，我们利用Python中的IF语句，检查用户是否存在是，我们把用户在表单中输入的用户名拿去与数据库中用户名相检验，而密码的验证我们利用Django中的check\_password()函数判断用户书写的密码和数据库中的密码是否一致，如果两项都一致，那么我们就把页面重定向和反向解析到我的界面上。

**4.4 我的模块的构建**

4.4.1 我的页面的初步完成

我们首先完成我的静态页面，此时我们借助bootstrap来完成一些静态的图标，例如头像的图标还有付款的图标等等。在静态页面中写入bootstrap中对应图标的代码，即可完成此页面。

4.4.2 已登陆时我的页面的完成

我们在登陆时把登陆者的ID存储在Django的session中，在我们的我的界面的后端接口中，我们先把登陆者的ID从session中提取出来，然后把数据返回到静态页面。之后，我们再在静态页面中进行if判断，如果有登陆者的ID我们就显示登陆者的头像和用户名，如果没有我们就显示默认头像和默认的登录字段。



**图5 我的页面展示图**

**4.5 购物车模块的构建**

4.5.1购物车模块的准备工作

跟之前一样，我们在开始购物车模块的构建之前，我们需要将购物

车项目搭建起来，命令行输入python manage.py startapp CartApp 创建成功购物车模块。

4.5.2 购物车模块模型的构建

因为购物车表与用户表、商品表属于Django中的多对多的关系，因

此，我们设计表结构时需要往模型中添加外键ForeignKey。

我们添加外键c\_goods,c\_user使它们分别对应商品表还有用户表，

然后我们再添加一些商品数目字段等。命令行输入python manage.py makemigrations和python manage.py migrate迁移数据至MySQL数据库中。

//购物车模块模型代码

**class** AxfCart(models.Model):  
c\_goods = models.ForeignKey(AxfGoods)  
 c\_user = models.ForeignKey(AxfUser)

c\_goods\_num = models.IntegerField(default=1)  
 c\_is\_select = models.BooleanField(default=**True**)  
  
 **class** Meta:  
 db\_table = **'cart'**

4.5.3 购物车模块的静态页面搭建

我们写出cart.html静态页面，写出购物车的样式，例如电话、地

址等等，然后我们还要写出商品的图片、价格等等样式。但是商品图片、价格的显示我们需要与后端接口相对应，所以我们要在静态页面上留出位置显示这些图片和数据。

4.5.4 购物车模块的后端接口书写

购物车的显示需要在用户登陆之后，才能够显示出来。因此，我们在书写后端接口时，要先验证用户是否已经登陆，因为之前我们已经将用户的id放入了session中，因此我们要先在session中提取用户id看看是否能提取出来，如果没有提取出来，那么我们就将页面重定向至注册界面，重定向利用redirect函数。

验证用户id之后，我们将购物车模型实例化，然后将数据通过

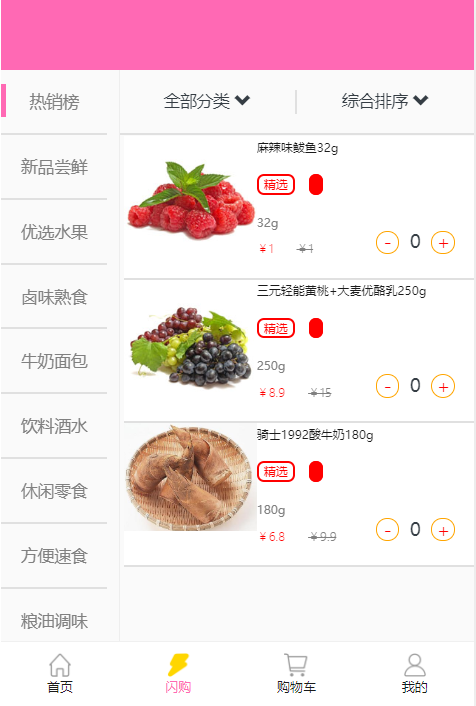
Render（）函数返回给前端数据显示。因为我们购物车页面需要的价格是总价，因此我们需要在views.py文件中写一个计算总价的函数get\_total\_price()将商品价格和数量传进去计算总价，再把数据返回前端。

4.5.5购物车页面的完善

我们的购物车页面必须要有对于商品的选择功能，因此，此处我们可以使用ajax异步刷新请求来做。ajax可以动态的刷新页面的内容，从而实现我们前端页面动态的显示价格、数量等数据。

我们通过编写cart.js在其中确定增加商品、减少商品URL和传输类型GET。在后端接口中我们通过将购物车模型实例化，然后如果我们点击了增加或减少按钮，我们就把数据表中的商品数量+1或者-1，需要注意的是，如果是减少的话，我们不能让数据减少为比0还要小，因为实际中，是不会存在负数数量的商品的，因此在商品数量小于0时，我们将购物车的实例化模型删除，就不会出错了。

按照上述内容，我们完成了后端接口的逻辑问题，现在我们要完成前端的增加和删除了，当我们通过URL解析，发现数据库中的商品数量是0时，因为后端发送给前端的状态码是不同的，所以此时我们的逻辑是，状态码正确时，我们显示样式，否则，我们将样式移除。



**图6 购物车模块展示**

**4.6订单模块的构建**

4.6.1 订单模块项目的完成

因为我们的商品放在购物车中，我们可以通过点击购物车页面的下单按钮，然后跳转到订单页面。在Pycharm命令行中输入python manage.py startapp Order 创建订单项目。

4.6.2 订单模块模型的完成

在订单模块的模型设计中，我们要设计两张表，第一张是order表，此表中的字段有o\_user,此字段是用来关联用户的，即是用户表的外键，还有time字段，主要是记录订单的生成时间。第二张表是order\_goods表，此表中的字段主要有og\_order用来关联order表的外键，og\_goods用以关联goods表的外键，还用价格、数目等等。

把所有的模型都建立好了之后，Pycharm命令行输入python manage.py makemigrations和python manage.py migrate迁移至数据库。

//订单模块模型代码

**class** AxfOrder(models.Model):

o\_user = models.ForeignKey(AxfUser, on\_delete=**None**)  
 o\_time = models.DateTimeField(auto\_now\_add=**True**)  
  
 **class** Meta:  
 db\_table = **'order'  
  
class** AxfOrderGoods(models.Model):

og\_order = models.ForeignKey(AxfOrder)  
 og\_goods = models.ForeignKey(AxfGoods)  
 og\_goods\_num = models.IntegerField()  
 og\_total\_price = models.FloatField()  
  
 **class** Meta:  
 db\_table = **'ordergoods'**

## 4.6.3 订单模块的接口完成

订单模块的接口需求是，我们需要把订单的ID通过外键关联值用户的ID然后把它存入数据库，还有就是利用外键关联，把商品表中的商品取出来，再计算出总价然后返回给前端页面。

这些步骤完成之后，我们的把购物车的数据删除，因为已经下过了单，购物车中当然不 能再有数据了，我们直接实例化购物车，然后利用delete（）函数把数据删除。

****

**图7 订单页面展示**

**5 总结**

在构建整个电商网站时，模型的建立和表与表的关系是最困难的点，因为我们必须要把网站中所有数据用字段的方式建立然后保存起来，还要理清每一张表的关系，分辨它们是一对一的关系、一对多的关系还是多对多的关系。例如，购物车模块模型的构建，我们要先明确cart表与goods表和user表构成多对多关系，然后还要用相关的外键把他们联系在一起，需要我们仔细思考。

**参考文献:**

[1]Adam Freeman.HTML5权威指南[M].人民邮电出版社:北京,2014-1:1.

[2][英] Ben Forta.MySQL必知必会[M].人民邮电出版社:北京,2009-1:1.

[3]胡阳.Django企业开发实战[M].人民邮电出版社:北京,2019-2:1.

[4][巴西] Luciano Ramalho.流畅的Python[M].人民邮电出版社:北京,2017-5:1.

[5]Randy Jay Yarger.MySQL权威指南[M].中国电力出版社:北京,2003-11-1:1.

[6]Baron,王小东.高性能MySQL[M].电子工业出版社:北京,2013:0.

[7]Eric Matthes,袁国忠译.Python编程从入门到实践[M].人民邮电出版社:北京,2016:1.

[8][美] 尼古拉斯·泽卡斯.JavaScript高级程序设计（第3版）[M].人民邮电出版社:北京,:2012-3:1.

[9][美] Erich Gamma / Richard Helm / Ralph Johnson / John Vlissides.设计模式[M].机械工业出版社:北京,2000-9:1.

[10][美] Elizabeth Castro / [美] Bruce Hyslop.HTML5与CSS3基础教程（第8版）[M].人民邮电出版社:北京,2014-5:1.

致 谢

光阴似箭，日月如棱。四年的时间，在我们漫长的人生旅途中是那么的短暂，但是，这短短的四年是最美好的青春，是最纯真的岁月，是最美丽的大学生活……我们的自学能力在这里得提升，我要感谢我们所有的老师，正是由于他们，我才能在各方面取得显著的进步，在此向他们表示我由衷的谢意，并祝所有的老师培养出越来越多的优秀人才，桃李满天下！

在这次的毕业论文中，我发挥了在学校学到的文化知识和技能的应用，也算是我最后一次做学校的作业了，我要感谢我的指导老师，还有我的辅导员老师，以及任课老师，感谢他们的教诲，让我知道在社会上懂得怎样去做好自己，端正自己的位置，为社会贡献出我自己的力量。写作毕业论文是一次系统再学习的过程，毕业论文的完成，同样也意味着新的学习生活的开始。我将铭记我曾是一名皖江学子，在今后的工作中把皖江的优良传统发扬光大。