顺德职业技术学院

毕 业 论 文

题 目  基于Django的Python Web开发

系 别 智能制造学院

年级专业 17计算机应用

学生姓名 刘俊杰

指导教师 余 方

专业负责人 张志强

答辩日期

**一、目录**

**目录**

[二、摘要 2](#_Toc12793)

[基于Django的Python Web开发 2](#_Toc7808)

[三、正文 3](#_Toc7242)

[第1章 Python简介 3](#_Toc27391)

[背景 3](#_Toc12310)

[1.1 简单 3](#_Toc28896)

[1.2 开源 3](#_Toc21312)

[1.3 高级语言 3](#_Toc32532)

[1.4 可移植性 3](#_Toc32555)

[1.5 解释执行 4](#_Toc16094)

[1.6 可扩展性 4](#_Toc8143)

[1.7 可嵌入性 4](#_Toc15629)

[1.8 丰富的类库 4](#_Toc29182)

[第2章 Django简介 5](#_Toc14521)

[第3章 电子商务系统的开发 8](#_Toc26289)

[第4章 电子商务系统的后端开发 32](#_Toc18291)

[4.1 Mysql数据库设计 32](#_Toc25270)

[4.2 用户认证模型设计 33](#_Toc30955)

[4.3 用户认证模型开发 35](#_Toc29633)

[4.3.1 用户注册 35](#_Toc10897)

[4.3.2 用户ID信息加密 42](#_Toc12566)

[4.3.3 用户信息保存 44](#_Toc31191)

[4.3.4 用户登出 45](#_Toc13077)

[4.4 其他安全处理与其他功能 46](#_Toc4997)

[4.4.1 跨站请求伪造保护 46](#_Toc27933)

[4.4.2 地址管理功能 47](#_Toc11416)

[第5章总结 48](#_Toc15940)

[python包版本 48](#_Toc16345)

**二、摘要**

基于Django的Python Web开发

针对在疫情期间不能外出，外出风险大、安全没保障的问题，利用计算机和Python语言进行在线购物送到家，极大程度减少了风险性。系统框架使用Python语言中的Django框架，利用HTML技术进行内容展示，MYSQL数据库进行表单存储。系统模块包括商品预览、账号注册、账号登录、下单、地址管理。该系统功能齐全、操作简便、实用性好，在电子商务领域具有很好的应用前景，使用Python语言开发的Django框架可以作为快速Web开发的应用框架，引起越来越多的开发者重视。

关键词：Django、PythonWeb、电子商务网站系统

Abstrac：Aiming at the problem of not being able to go out during the epidemic, the risk of going out and the lack of security, the use of computers and Python language for online shopping and delivery to the home has greatly reduced the risk. The system framework uses the Django framework in the Python language, uses HTML technology for content display, and the MYSQL database for form storage. The system module includes product preview, account registration, account login, order placement, and address management. The system has complete functions, simple operation, good practicability, and has a good application prospect in the field of e-commerce. The Django framework developed using Python language can be used as an application framework for rapid web development, which has attracted more and more developers.

**Key Words：**Django、PythonWeb、E-commerce website system

**三、正文**

**第1章 Python简介**

背景

Python语言是由荷兰人Guido van Rossum创建的。Python是一种面向对象的程序设计语言，它作为通

能强大的编程语言广受好评。Python语言的语法简洁而清晰，具有丰富和强大的类库，常被称为胶水语言

n是解释型语言，适用于多个平台，可以运行在Windows、MAC和各种Linux/UNIX系统上。

1.1 简单

Python是一种代表简单思想的语言。一个良好的Python程序像一篇英文文档，非常接近于人的自然语言 发Python程序时，研发人员更专注于解决实际问题，而不必考虑计算机语言的细节。

1.2 开源

Python是Floss(自由/开放源码软件)之一，用户可以查看它的源代码，对它做改动或进行二次开发。用户 为使用Python支付费用，也不涉及版权问题。

1.3 高级语言

当用户使用Python编写程序时，无需要再考虑如何管理程序使用的内存一类的底层

1.4 可移植性

Python的标准实现是由可移植的ANSIC编写的，可以在目前所有的主流平台上编译和运行。尽管有一些 同平台开发的特有模块，但是在任何一个平台上用Python开发通用软件都可以稍作修改或者原封不动地在其 上运行。这种可移植性既适用于不同的架构，也适用于不同的操作系统。

1.5 解释执行

Python是一种解释性语言，使用这种语言编写的程序，不需要编译成计算机就可执行二进制代码，同时 接从源代码运行程序。Python程序是通过Python解释器解释并执行，Python解释器首先把程序的源代码转换 字节码的中间形式，然后再把它翻译成计算机语言并执行。这使Python更简单，也使Python程序更易于移植

1.6 可扩展性

Python的可扩展性使程序员能够灵活地附加程序，缩短了开发周期，因为Python是基于C语言开发的，所 用C/C++来编写Python的扩展性。

# 1.7 可嵌入性

# Python的嵌入性是指它可以作为一种成熟的脚本语言，并且以一种很方便的方式嵌入其他的程序中，比 C/C++中。

1.8 丰富的类库

Python的标准库很大。基于庞大的标准库，可以编写程序来处理各种工作，包括正则表达式、文档生成 测试、线程的功能。除了标准库，它还有大量高质量的第三方库，如NumPy、Twisted、Django等。

**第2章 Django简介**

背景

Django是用Python开发的一套开放源代码的Web框架。Django是一套组件，能够快速高效地开发Web应 架的作用是把程序的整体架构搭建好，在其基础上开发者填写业务逻辑。框架在需要使用逻辑功能时调用业务 而不需要开发者自己去调用。使用Django框架，可以减少很多开发中的繁琐的工作，让开发者将更多的精力 编写业务逻辑，而不是重复生产基础组件的工作。

2.1 Django的特点

2.1.1丰富的组件

Django框架有丰富的用于开发Web应用的组件，这些组件都是用Python开发的，并为 所修改和使用。设计Django框架的组件的目的是实现重用性，并使之具有易于性。

2.1.2对象关系映射和多数据库支持

Django框架的数据库组件——对象关系映射提供了数据模块和数据 间的接口。支持的数据库包括SQLite、MySQL和PostgreSQL等。这种设计使数据库的切换只需要修改配置 可，使开发者得数据库设计具有很好的灵活性。

2.1.3简洁的URL设计

Django框架中URL系统设计非常强大且灵活，可以在Web应用中为URL设计匹配 并用Python函数处理。这种设计使Web应用开发者可以创建友好的URL，使之更适合于搜索引擎的搜索。

2.1.4自动化的管理界面

Django框架已经提供了一个易用的管理界面，这个界面可以方便地管理用户数 有高度的灵活性和可配置性。

2.1.5强大的开发环境

Django框架提供了强大的Web开发环境，有一个可用于开发和测试的轻量级W 器。启用调试模式后，Django会显示大量的调试信息，使开发者很容易清除bug。

2.1.6使用MTV框架

Django框架遵循Web开发的MVC(Model模型、View视图、Controller控制器）架构 框架也可以是一种MVC框架，而Django更关注的是模型（Model）、模板（Template）和视图（View）

2.1.7强大的可扩展性

Django允许使用第三方库来扩展Web应用程序。

2.2 Django框架的工作原理

用户在浏览器输入URL后回车，浏览器会对URL进行检查，首先判断使用的是什么协议。如果是HTTP协 按照Web来处理，再调用DNS域名解析服务器解析，将域名转换成IP地址，然后经过网络传输到对应Web服务器接收到用户浏览器发来的Web请求后，会对URL进行解析，然后调用View中的业务逻辑（MTV 以及Model(MTV中的M）,与数据库进行交互，再将数据发送到Templates(MTV中的T）进行渲染，最后发送浏览器以合适的方式呈现给用户。

2.3 Django的模块及功能

|  |  |
| --- | --- |
| 配置文件 | 作用 |
| urls.py | 配置文件，主要是将URL映射到应用程序中的相应函数。 |
| views.py | 处理用户发来的请求，从urls.py中对应过来，通过渲染templates中的网页可以将显示内容 网页。 |
| models.py | 与数据库操作相关，存入或读取数据时使用。 |
| templates | 文件夹：views.py中的函数渲染templates中的HTML模板，得到动态内容的网页。 |
| admin.py | 网站后台管理相关操作，可以用很少的代码就拥有强大的后台管理系统 |
| settings.py | Django的主配置文件，用于设置参数连接数据库等 |

2.3表

**第3章 电子商务系统的开发**

**背景**

为了简化网站操作难度，适合广大群众，特别是年长的长辈一切以简洁为主，让人一眼看去就知道自己想要什么，只需要填写地址就能送货到家，不用为出门而烦恼，减少人与人的接触。

。

* 1. 草稿设计（静态网页部分）

（草稿原图放在文件夹的Design内，但不代表最终效果）



图1-1首页上部分



图1-2首页中部分



图1-3首页下部分

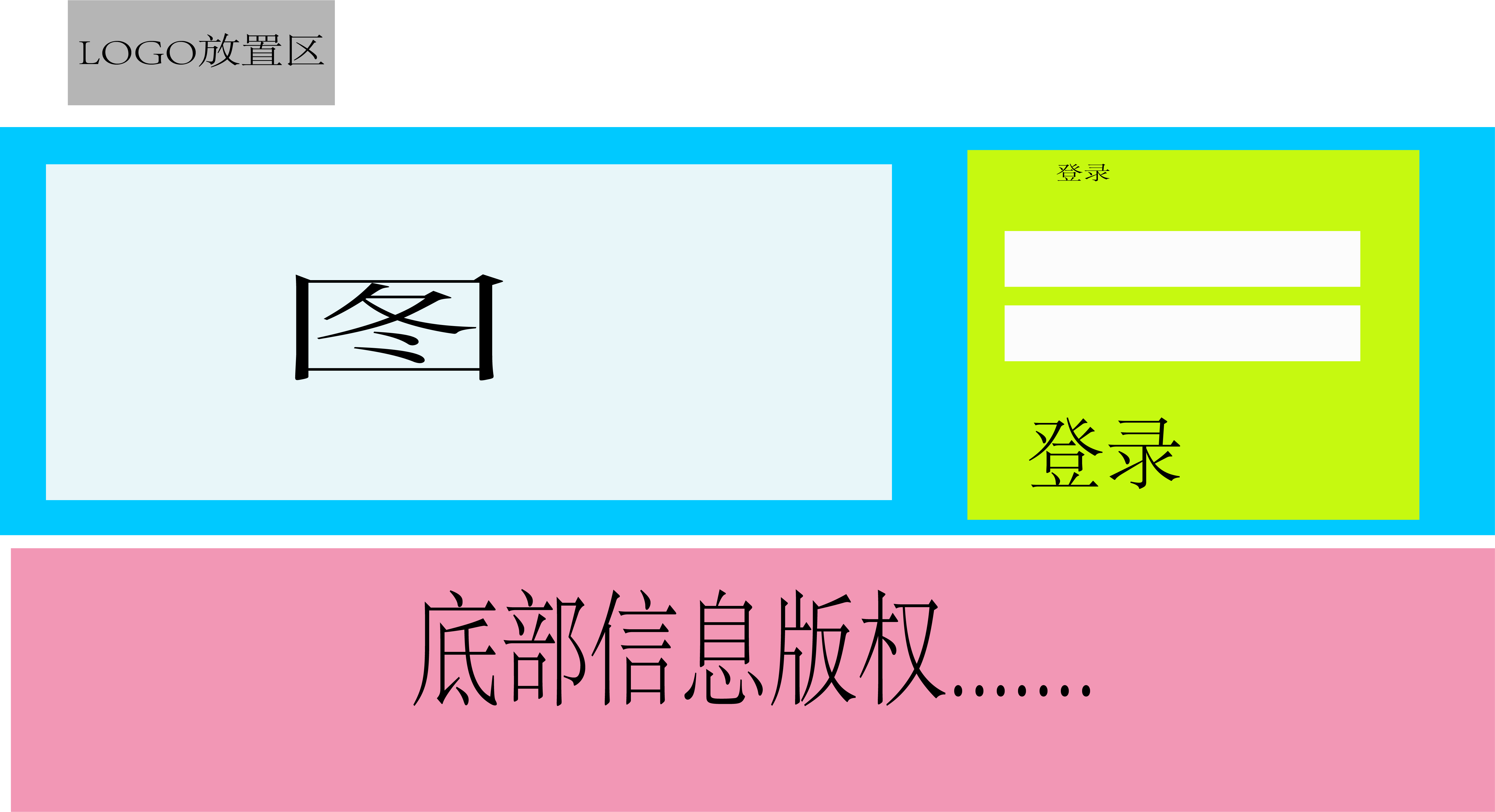


图1-4登录页部分

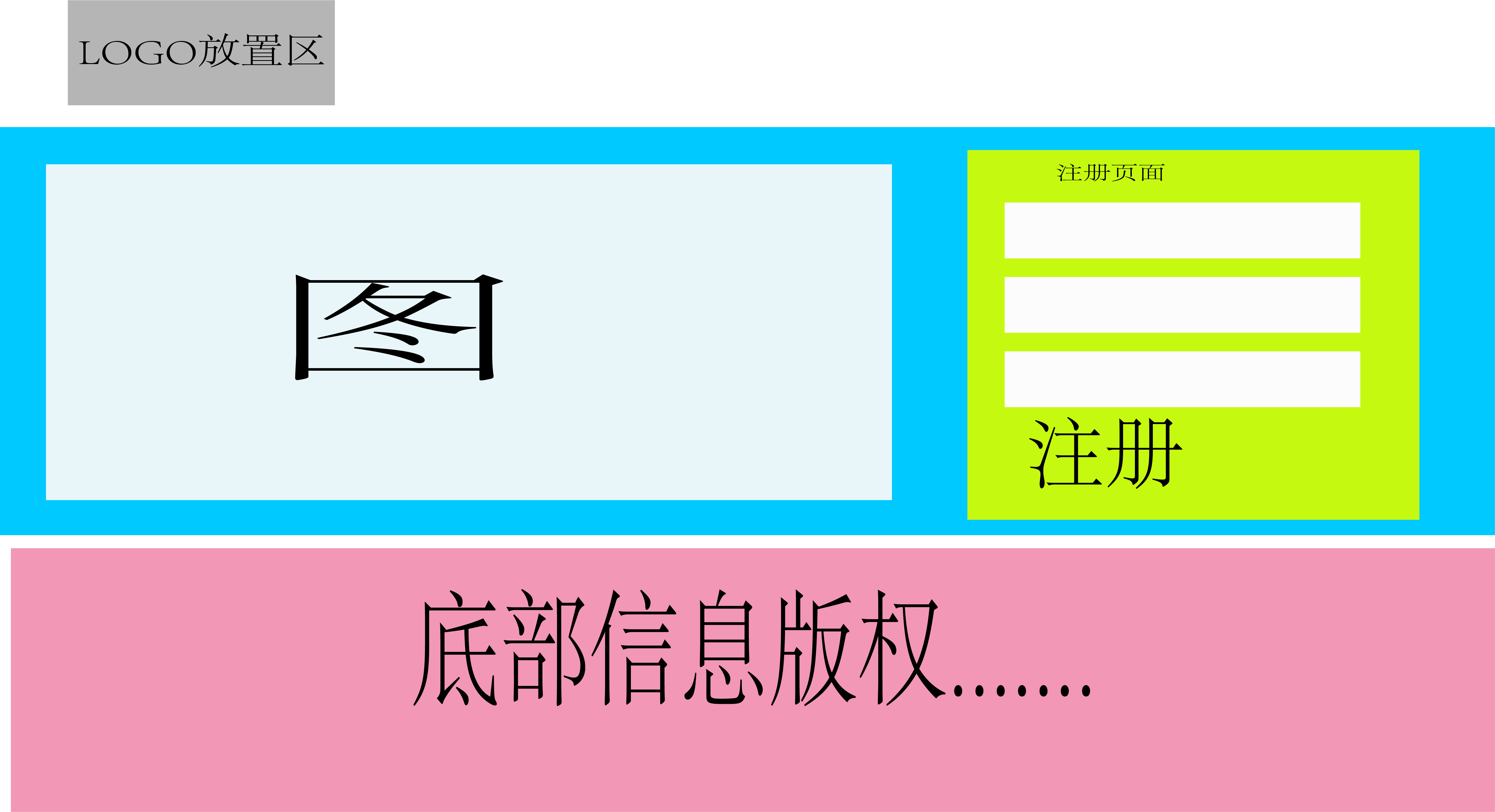


图1-5注册页面部分



图1-6购物车部分

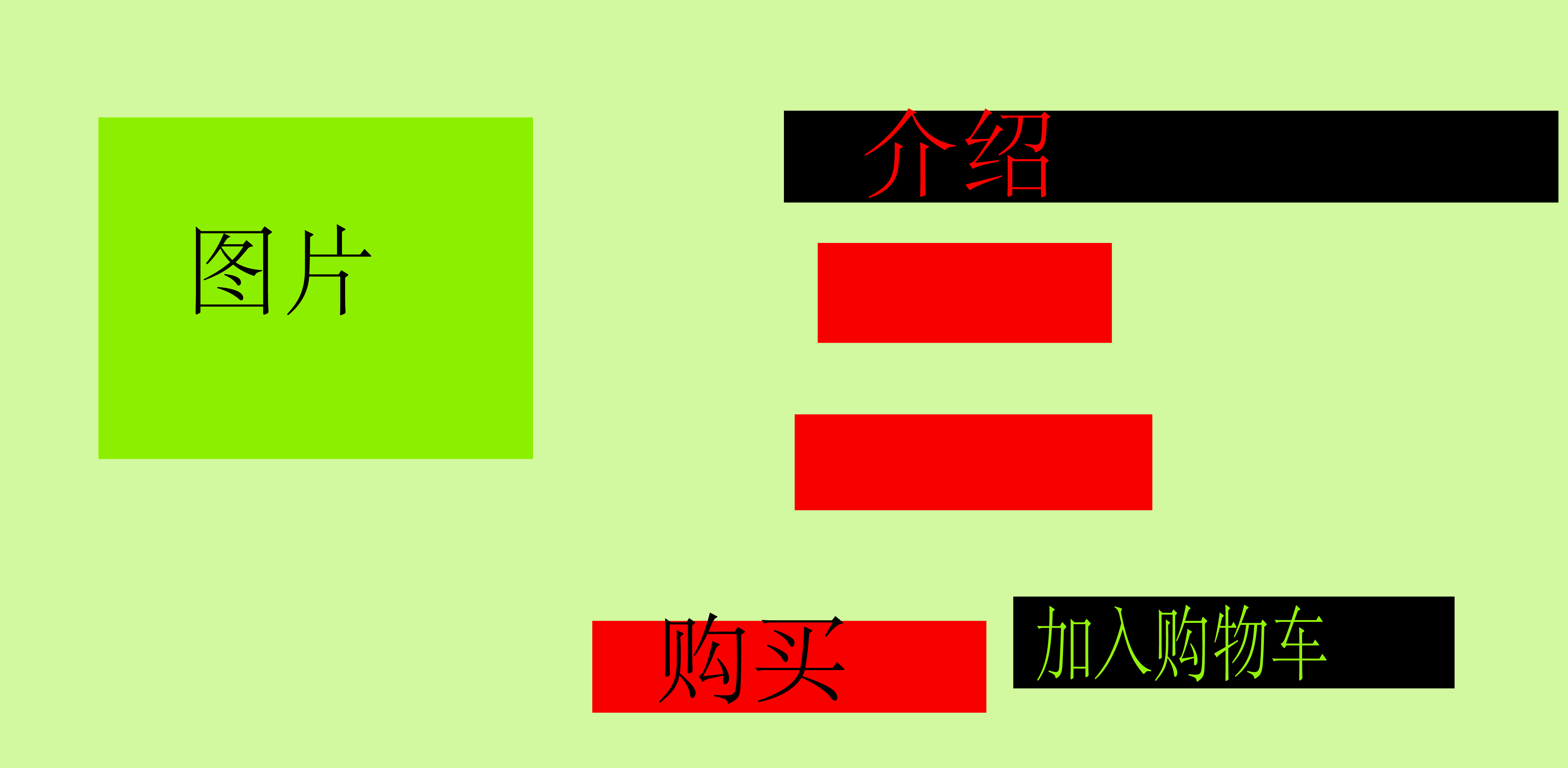


图1-7商品详细页部分

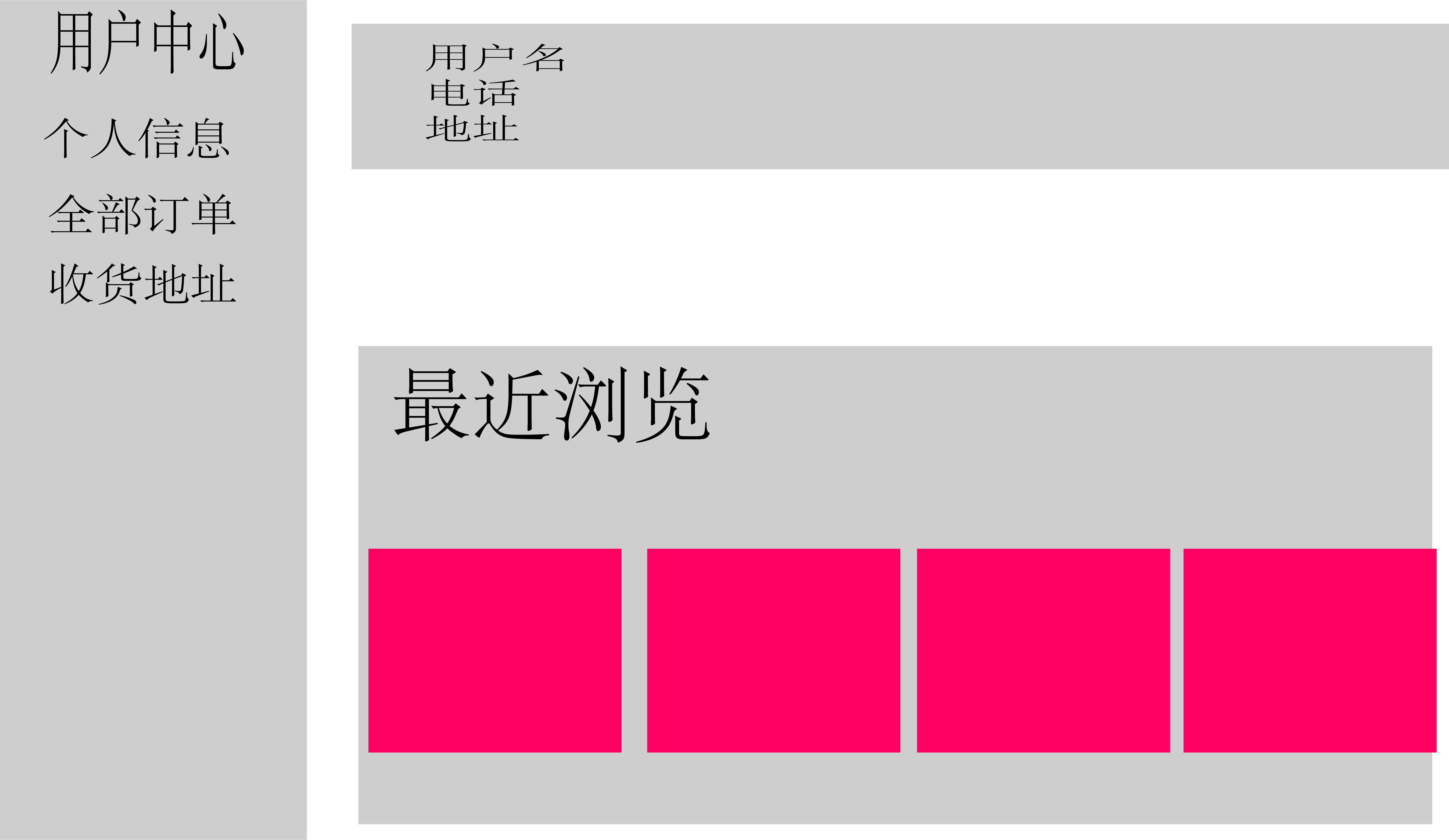


图1-8个人中心用户信息部分

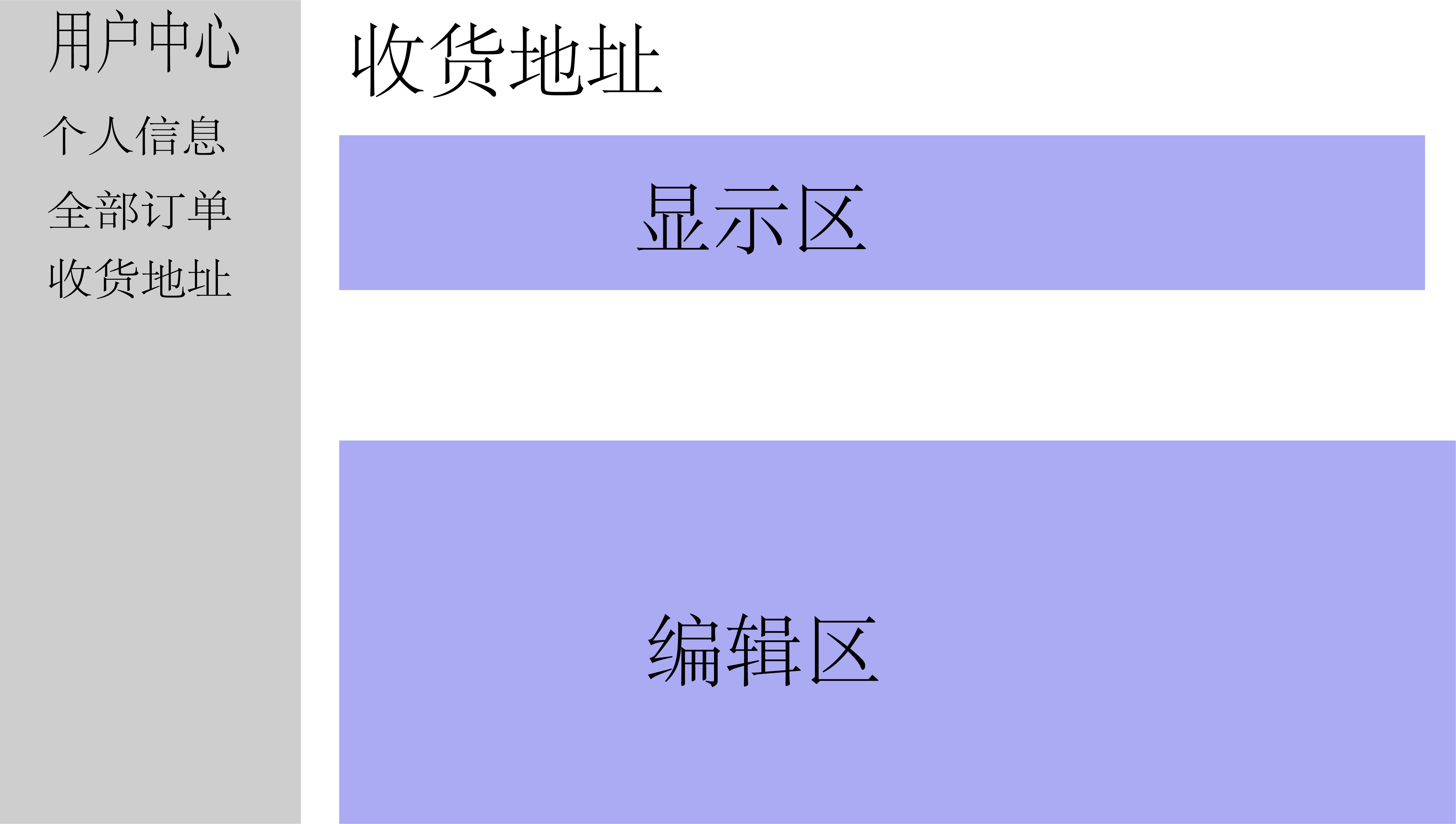


图1-9个人中心收货地址部分

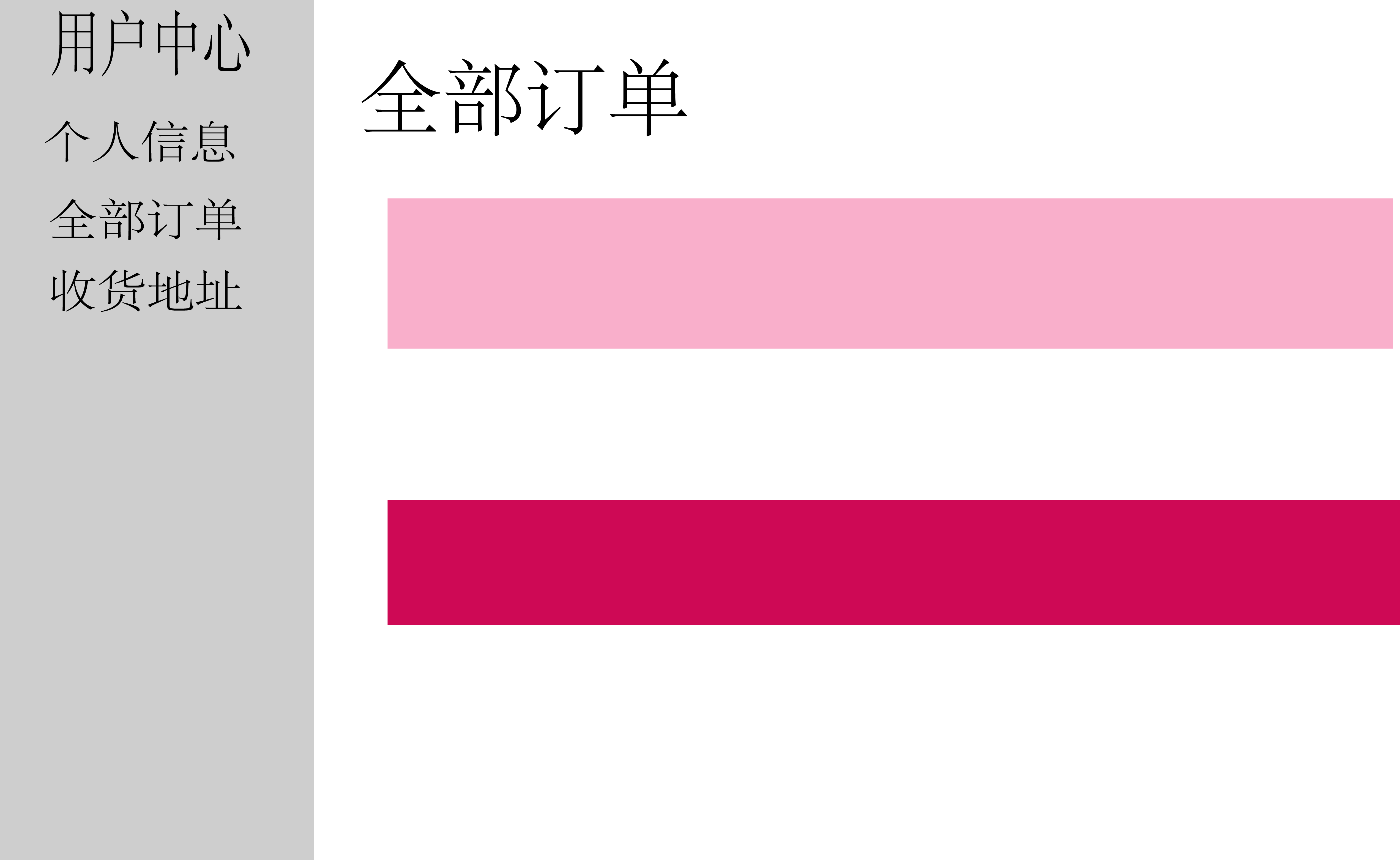


图1-10个人中心全部订单部分

2.1网页实现部分

2.1.1 本次使用的开发软件有WebStorm, Adobe Photoshop 2020, Adobe Illustrator CC 2019，使用的语言有HTML+CSS，

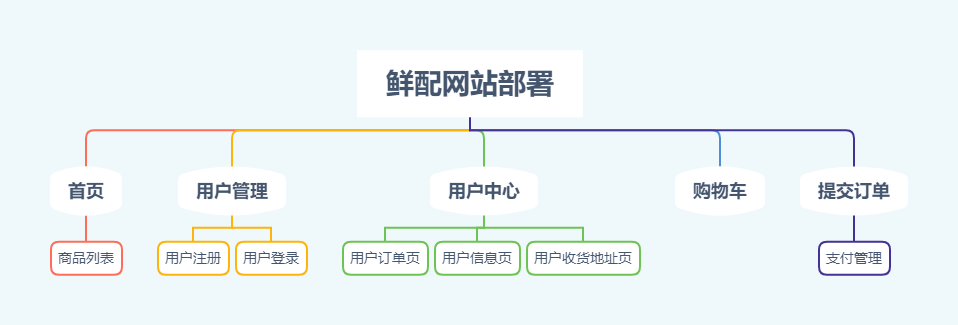
|  |  |
| --- | --- |
| 使用的软件 | 作用 |
| WebStorm | 代码设计 |
| Adobe Photoshop 2020 | 插图设计 |
| Adobe Illustrator CC 2019 | 草稿设计 |
| XMind ZEN | 逻辑图设计 |

JavaScript（JQuery）。

|  |  |
| --- | --- |
| 使用的语言 | 作用 |
| HTML | 布局设计 |
| CSS | 样式设计 |
| JavaScript（JQuery） | 特效设计 |

2.1.2 代码设计规划

本系统设计共五大功能模块



2.1.3 效果实现

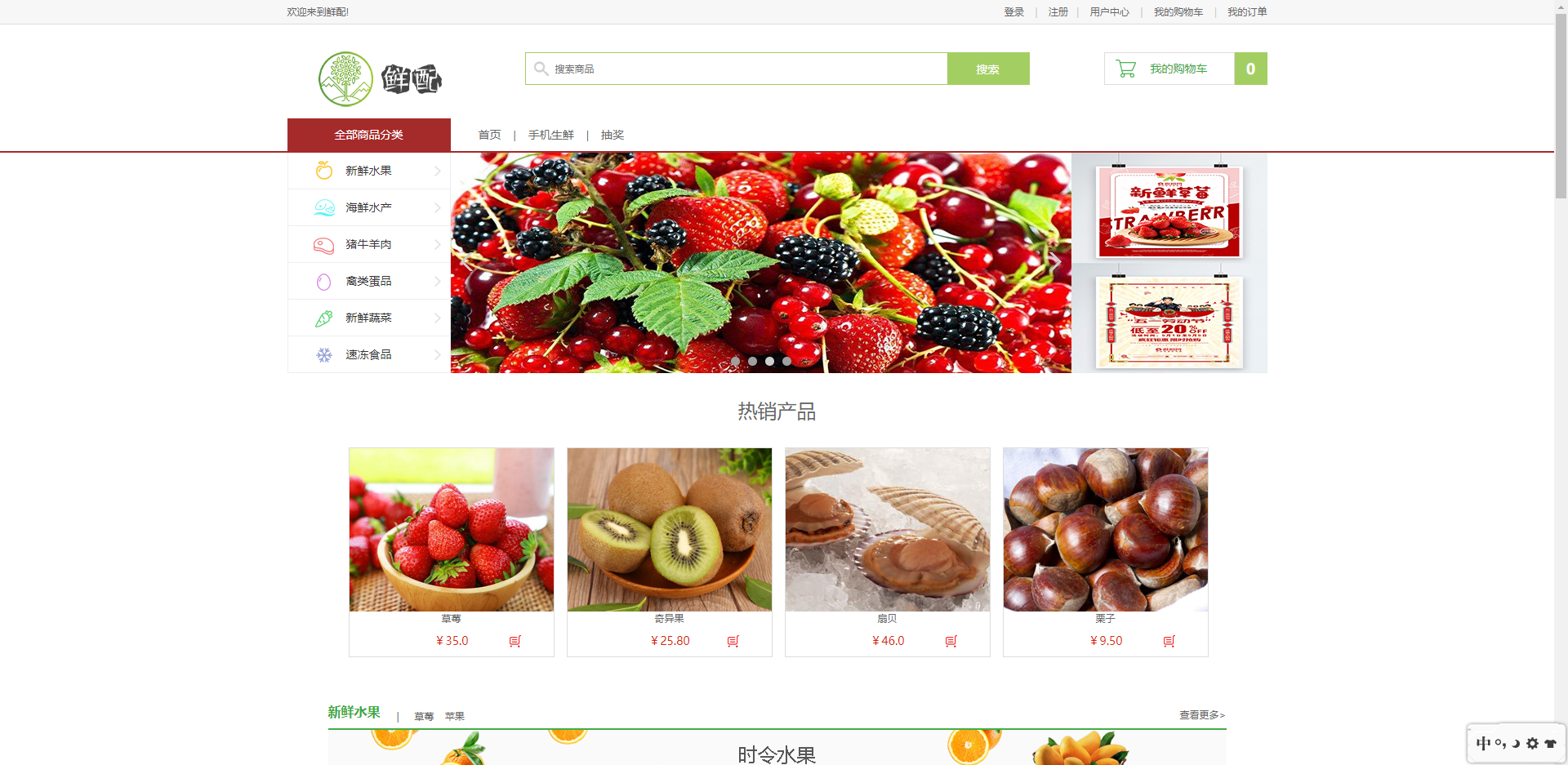
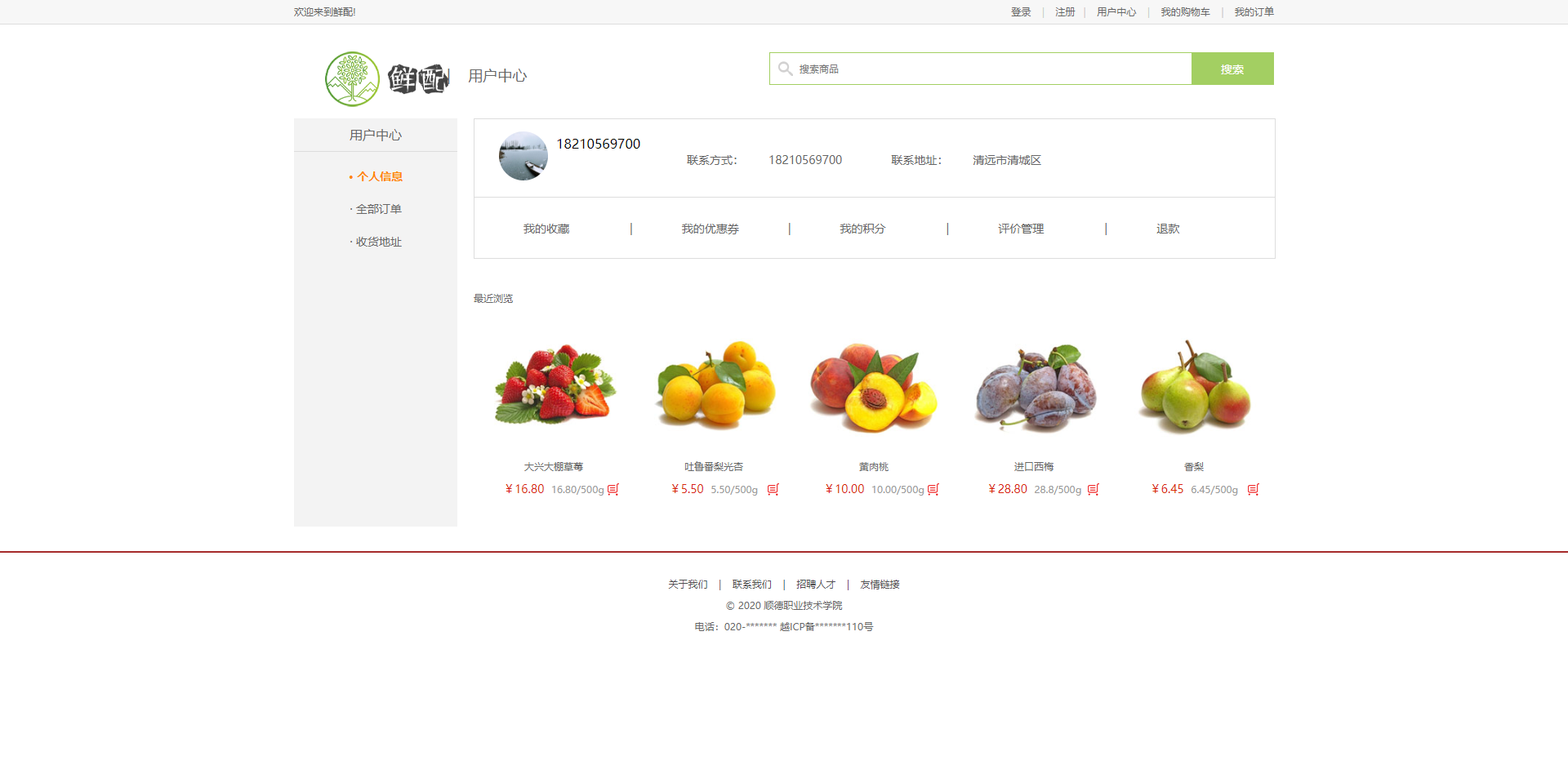


图1-1 Index.html



图1-2 Login.html

图1-3 register.html

图1-4 user\_center\_info.html

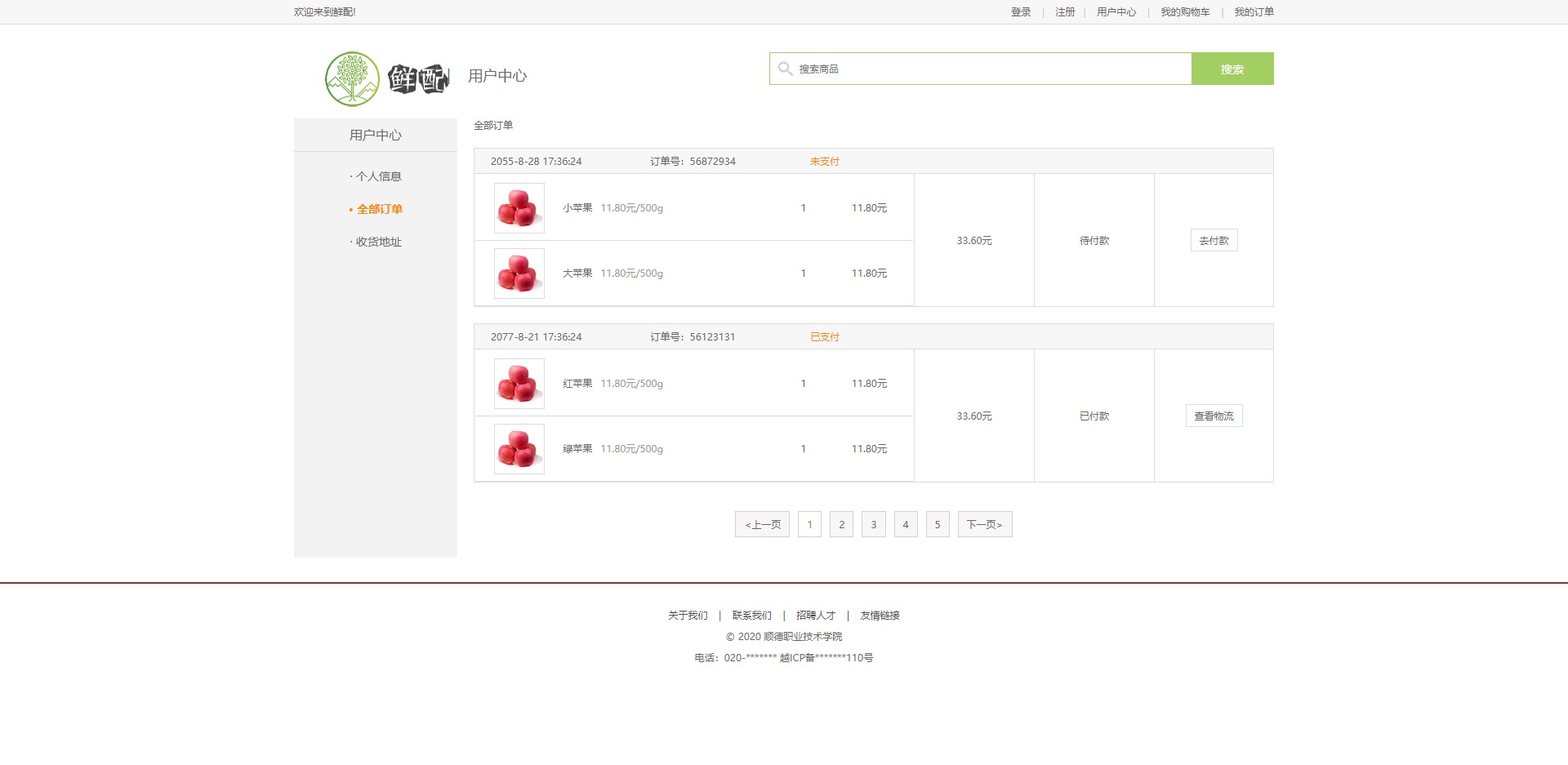


图1-5 user\_center\_order.html

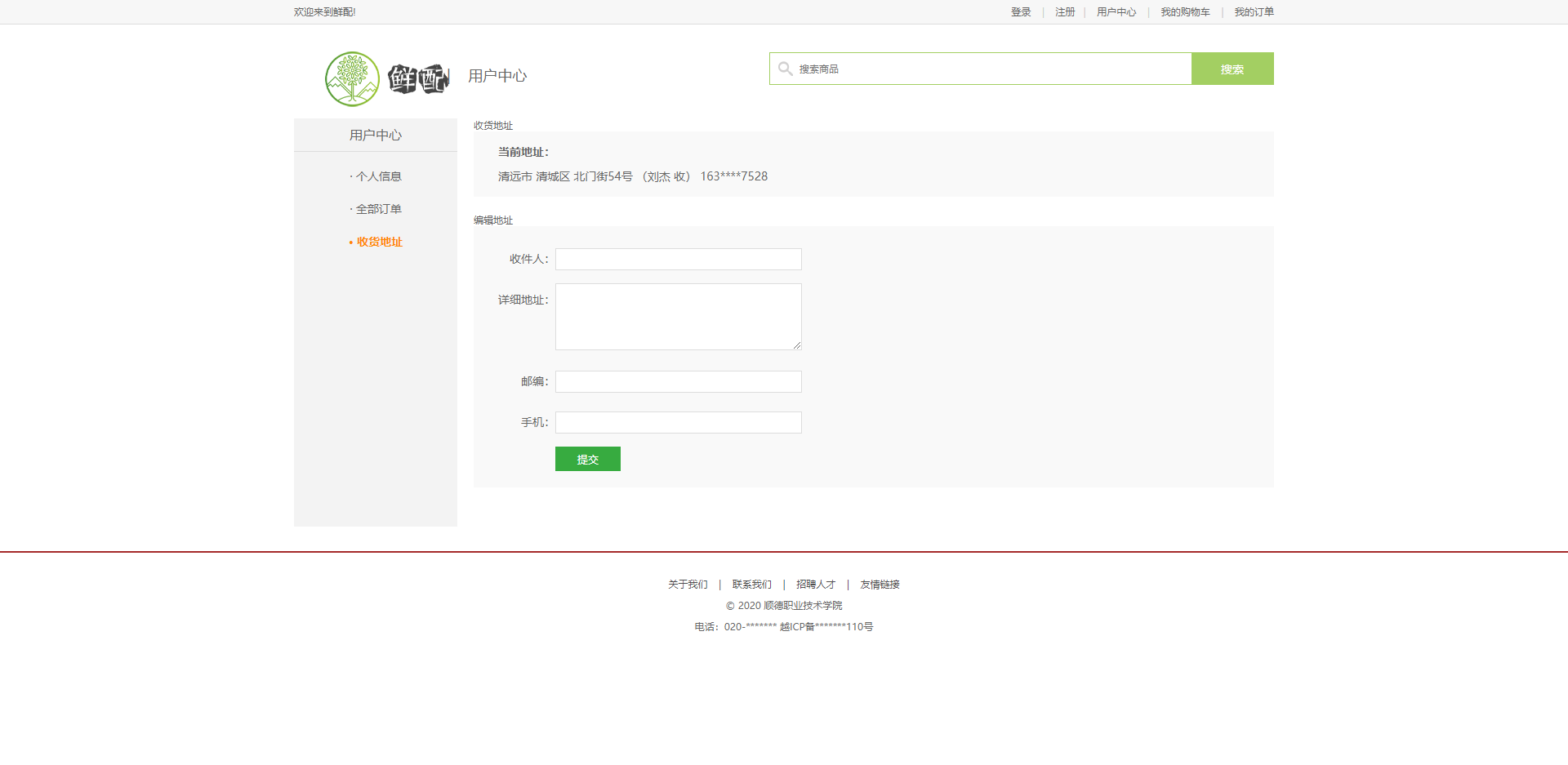


图1-6 user\_center\_site.html

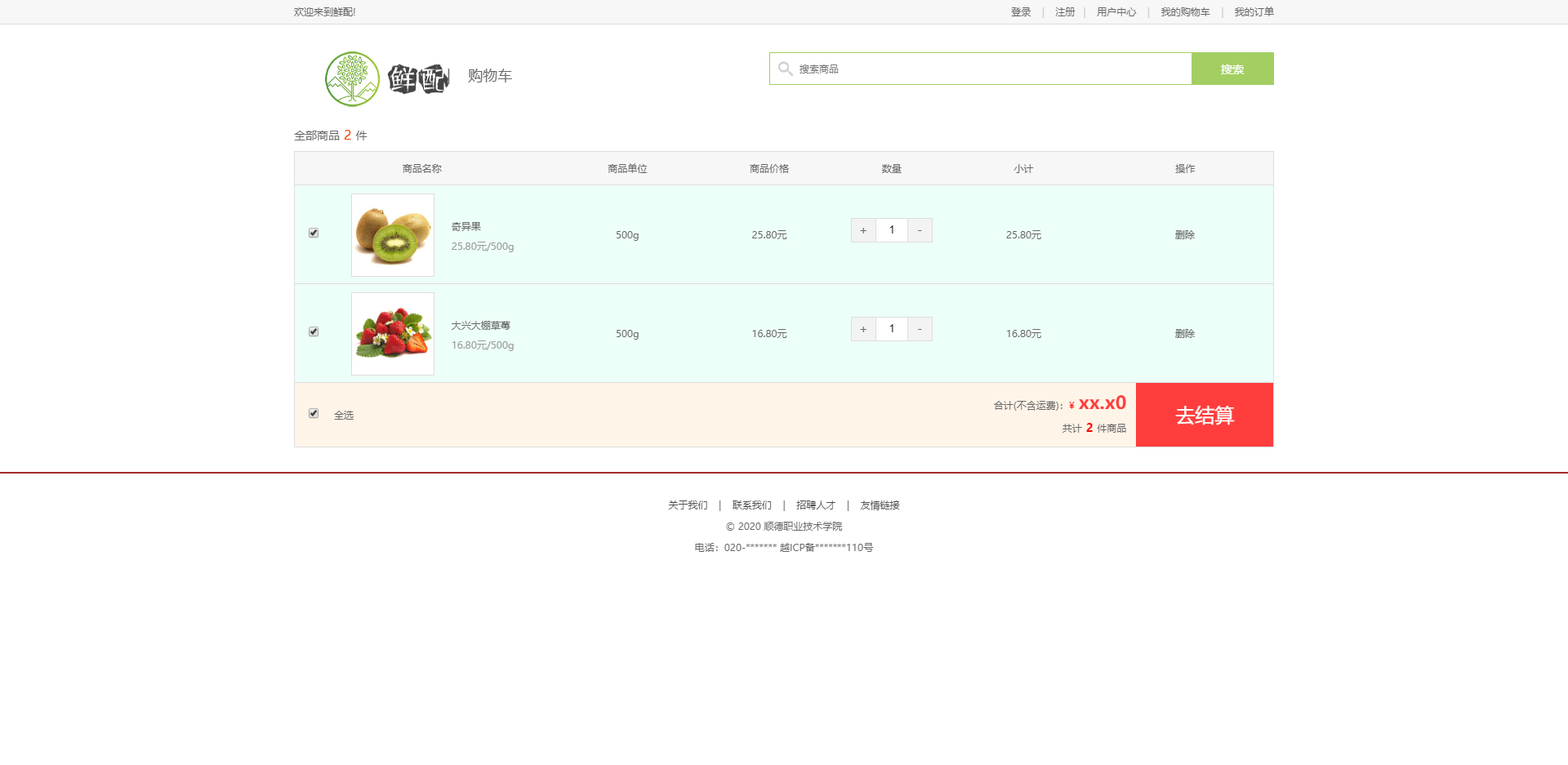
图1-7 cart.html



图1-8 list.html

2.1.4 代码简说

CSS

|  |  |
| --- | --- |
| CSS文件 | 作用 |
| main.css | 网页整体样式 |
| reset.css | 全局默认样式（兼容性协调） |

表1-1

HTML

|  |  |
| --- | --- |
| 网页文件 | 说明 |
| index.html | 网站首页，顶部“注册|登录”和用户信息是切换显示的，商品分类菜单点击直接链接滚动到本页面商品模块。首页已加入幻灯片效果。 |
| list.html | 商品列表页，商品分类菜单鼠标悬停时切换显示和隐藏，点击菜单后链接到对应商品的列表页。 |
| detail.html | 商品详情页，某一件商品的详细信息 |
| cart.html | 我的购物车页，列出已放入购物车上的商品 |
| place\_order.html | 提交订单页 |
| login.html | 登录页面 |
| register.html | 注册页面，已加入了初步的表单验证效果， |
| user\_center\_info.html | 用户中心-用户信息页 用户中心功能一，查看用户的基本信息 |
| user\_center\_order.html | 用户中心-用户订单页 用户中心功能二，查看用户的全部订单。 |
| user\_center\_site.html | 用户中心-用户收货地址页 用户中心功能三，查看和设置用户的收货地址 |

表1-2

javascript

|  |  |  |
| --- | --- | --- |
| 文件 | 作用 | 备注 |
| slide.js | 幻灯片样式 | 自编写 |
| register.js | 注册逻辑判定 | 自编写 |
| jquery-1.12.4.min.js | JavaScript代码库 | 基于javascript类库的框架 |
| jquery-ui.min.js | 策划的用户界面交互 | 需配合jquery-1.12.4.min使用 |
| jquery.cookie.js | 用户允许的情况下存储用户的登录信息等 | 需配合jquery-1.12.4.min使用 |

2.1.5 主要核心代码解析

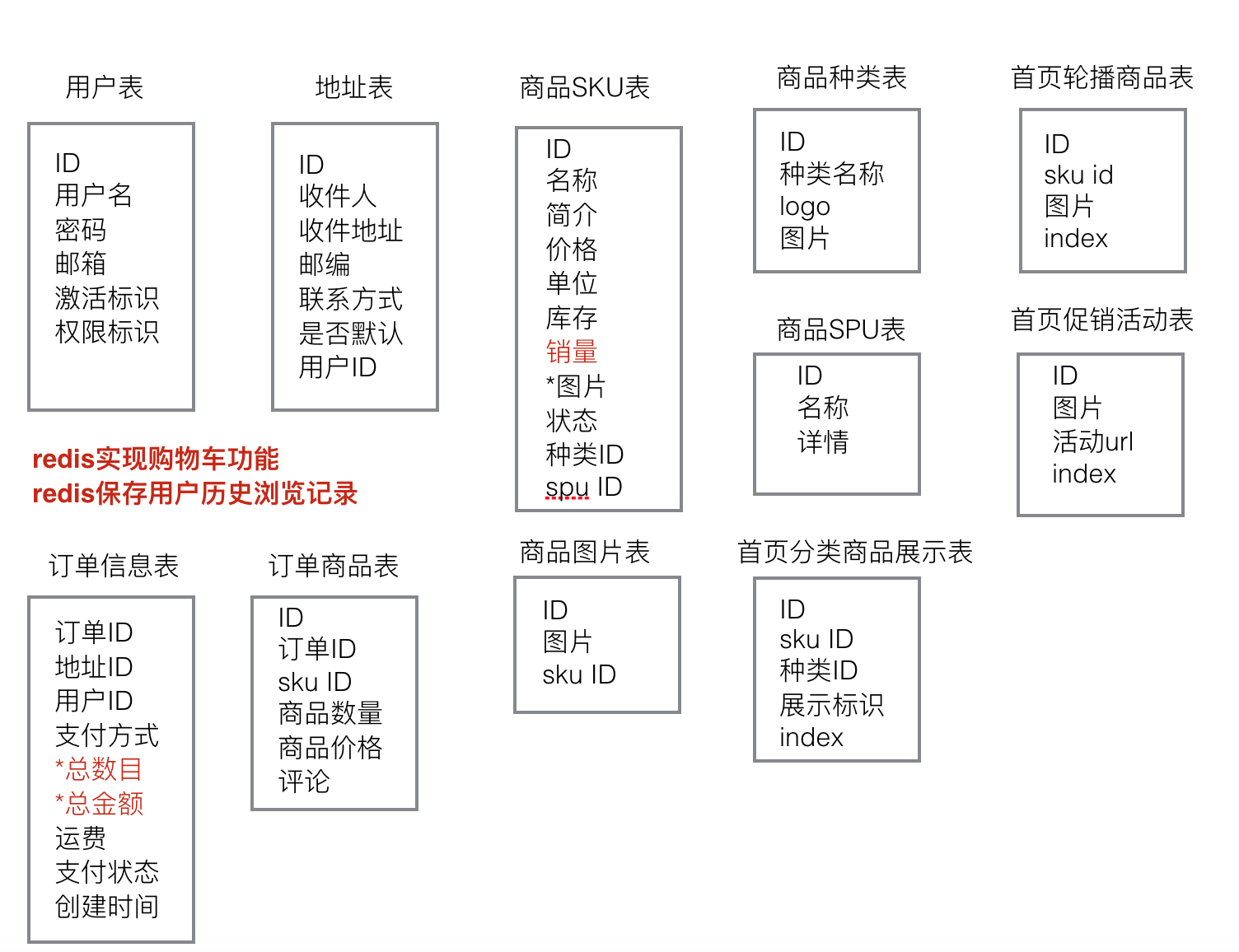
本次设计用jqery做注册逻辑参考网上例子，并结合网页姜老师的正则表达式小案例结合做出的效果。

|  |
| --- |
| $(function() {  var error\_name = false;  var error\_password = false;  var error\_check\_password = false;  var error\_email = false;  var error\_check = false;  $('#user\_name').blur(function() {  check\_user\_name();  });  $('#pwd').blur(function() {  check\_pwd();  });  $('#cpwd').blur(function() {  check\_cpwd();  });  $('#email').blur(function() {  check\_email();  });  $('#allow').click(function() {  if ($(this).is(':checked')) {  error\_check = false;  $(this).siblings('span').hide();  } else {  error\_check = true;  $(this).siblings('span').html('请勾选同意');  $(this).siblings('span').show();  }  });  function check\_user\_name() {  var len = $('#user\_name').val().length;  if (len < 5 || len > 20) {  $('#user\_name').next().html('请输入5-20个字符的用户名') $('#user\_name').next().show();  error\_name = true;  } else {  $('#user\_name').next().hide();  error\_name = false;  }  }  function check\_pwd() {  var len = $('#pwd').val().length;  if (len < 8 || len > 20) {  $('#pwd').next().html('密码最少8位，最长20位') $('#pwd').next().show();  error\_password = true;  } else {  $('#pwd').next().hide();  error\_password = false;  }  }  function check\_cpwd() {  var pass = $('#pwd').val();  var cpass = $('#cpwd').val();  if (pass != cpass) {  $('#cpwd').next().html('两次输入的密码不一致') $('#cpwd').next().show();  error\_check\_password = true;  } else {  $('#cpwd').next().hide();  error\_check\_password = false;  }  }  function check\_email() {  var re = /^[a-z0-9][\w\.\-]\*@[a-z0-9\-]+(\.[a-z]{2,5}){1,2}$/;  if (re.test($('#email').val())) {  $('#email').next().hide();  error\_email = false;  } else {  $('#email').next().html('你输入的邮箱格式不正确') $('#email').next().show();  error\_check\_password = true;  }  }  $('#reg\_form').submit(function() {  check\_user\_name();  check\_pwd();  check\_cpwd();  check\_email();  if (error\_name == false && error\_password == false && error\_check\_password == false && error\_email == false && error\_check == false) {  return true;  } else {  return false;  }  });  }) |
| register.js |

|  |
| --- |
| $(function(){  var $slides = $('.slide\_pics li');  var len = $slides.length;  var nowli = 0;  var prevli = 0;  var $prev = $('.prev');  var $next = $('.next');  var ismove = false;  var timer = null;  $slides.not(':first').css({left:760});  $slides.each(function(index, el) {  var $li = $('<li>');  if(index==0)  {  $li.addClass('active');  }  $li.appendTo($('.points'));  });  $points = $('.points li');  timer = setInterval(autoplay,4000);  $('.slide').mouseenter(function() {  clearInterval(timer);  });  $('.slide').mouseleave(function() {  timer = setInterval(autoplay,4000);  });  function autoplay(){  nowli++;  move();  $points.eq(nowli).addClass('active').siblings().removeClass('active');  }  $points.click(function(event) {  if(ismove)  {  return;  }  nowli = $(this).index();  if(nowli==prevli)  {  return;  }    $(this).addClass('active').siblings().removeClass('active');  move();  });  $prev.click(function() {  if(ismove)  {  return;  }  nowli--;  move();  $points.eq(nowli).addClass('active').siblings().removeClass('active');  });    $next.click(function() {  if(ismove)  {  return;  }  nowli++;  move();  $points.eq(nowli).addClass('active').siblings().removeClass('active');  });  function move(){  ismove = true;  if(nowli<0)  {  nowli=len-1;  prevli = 0  $slides.eq(nowli).css({left:-760});  $slides.eq(nowli).animate({left:0},800,'easeOutExpo');  $slides.eq(prevli).animate({left:760},800,'easeOutExpo',function(){  ismove = false;  });  prevli=nowli;  return;  }  if(nowli>len-1)  {  nowli = 0;  prevli = len-1;  $slides.eq(nowli).css({left:760});  $slides.eq(nowli).animate({left:0},800,'easeOutExpo');  $slides.eq(prevli).animate({left:-760},800,'easeOutExpo',function(){  ismove = false;  });  prevli=nowli;  return;  }  if(prevli<nowli)  {  $slides.eq(nowli).css({left:760});  $slides.eq(prevli).animate({left:-760},800,'easeOutExpo');  $slides.eq(nowli).animate({left:0},800,'easeOutExpo',function(){  ismove = false;  });  prevli=nowli;    }  else  {  $slides.eq(nowli).css({left:-760});  $slides.eq(prevli).animate({left:760},800,'easeOutExpo');  $slides.eq(nowli).animate({left:0},800,'easeOutExpo',function(){  ismove = false;  });  prevli=nowli;  }  }  }) |
| slide.js |

**第4章 电子商务系统的后端开发**

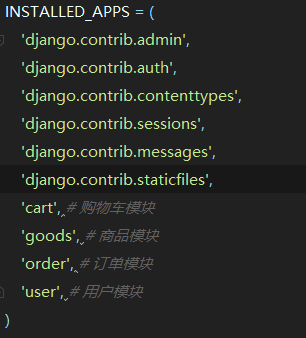
**4.1 Mysql数据库设计**

****

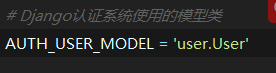
**思路源自此文章并做一部分简化：**<https://wenku.baidu.com/view/50830a104935eefdc8d376eeaeaad1f3469311d6.html>

因为DJango里面内置了用户认证系统所以我只需填写除去用户表与地址表之外的数据库就可以了！！！

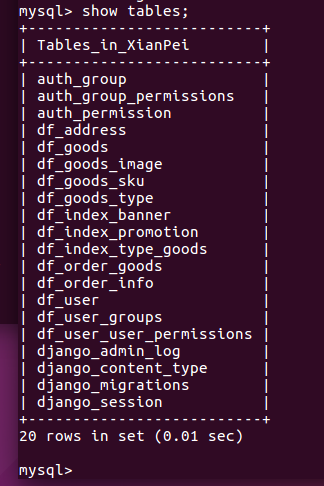
**4.2 用户认证模型设计**



使用Django内置认证系统



生成后如图：



**4.3 用户认证模型开发**

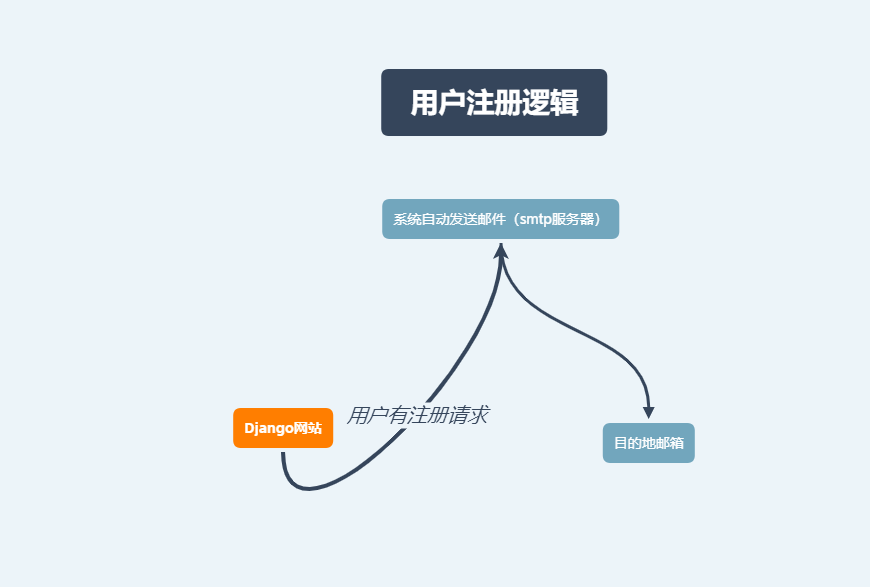
|  |  |
| --- | --- |
| **方法名** | **备注** |
| create\_user | **创建用户** |
| authenticate | **登录验证** |
| login | **记录登录状态** |
| logout | **退出用户登录** |
| is\_authenticated | **判断用户是否登录** |
| login\_required装饰器 | **进行登录判断** |

表4.2

参见<https://docs.djangoproject.com/en/3.0/topics/auth/default/> 之认证用户和《Python新手使用Django架站的16堂课》的275页-318页。

4.3.1 用户注册

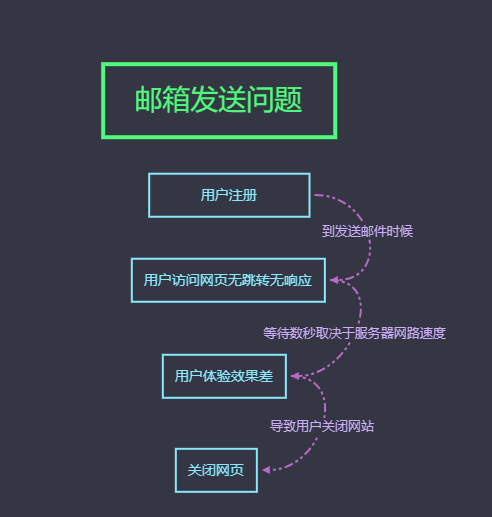
**（1）项目想通过用户注册之后系统会自动发送邮件到用户邮箱用户点击链接后系统允许登录**



1. **注册逻辑核心代码**

|  |
| --- |
| class RegisterView(View):  def get(*self*, request):  return render(request, 'register.html')  def post(*self*, request):  username = request.POST.get('user\_name')  password = request.POST.get('pwd')  allow = request.POST.get('allow')  email = request.POST.get('email')  *# 1.进行数据校验*  if not *all*([username, password, email]):  return render(request, 'register.html', {'errmsg': '数据不完整'})  if not re.match(r'^[a-z0-9][\w.\-]\*@[a-z0-9\-]+(\.[a-z]{2,5}){1,2}$', email):  return render(request, 'register.html', {'errmsg': '邮箱格式不正确'})  if allow != 'on':  render(request, 'register.html', {'errmsg': '请同意用户协议'})  *# 2.进行业务处理，进行用户注册*  *# 校验用户名是否重复*  try:  User.objects.get(username=username)  except User.DoesNotExist:  user = None  *# 用户已存在*  if user:  return render(request, 'register.html', {'errmsg:用户已存在'})  user = User.objects.create\_user(username, email, password)  user.is\_active = 0  user.save()  *# 生成用户加密信息，加密token*  serializer = Serializer(settings.SECRET\_KEY, 3600)  info = {'confirm': user.id}  token = serializer.dumps(info)  token = token.decode()  *# 发邮件*  send\_register\_active\_email.delay(email, username, token)  *# 3.返回应答*  return redirect(reverse('goods:index')) |
| **路径apps/user/view.py** |

**奈何如果我如果在注册时候同时发送邮件会导致网页无响应，需要等待，为了解决这个问题而做出二次修改，原因图示如下**



**于是找到新方法去二次制作邮件发送系统**

**参考文章：**<https://www.jianshu.com/p/620052aadbff>

1. **邮件发送系统二次修改异步 Celery(使用celery异步发送邮件）.**

**解决方案图示**



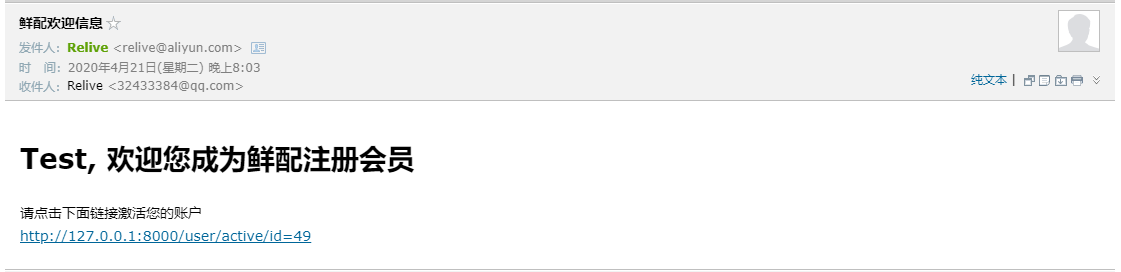
**核心代码块**

|  |
| --- |
| *# 使用celery*  from django.core.mail import send\_mail  from django.conf import settings  from django.template import loader, RequestContext  from celery import Celery  import time  *# 在任务处理者一端加这几句*  import os  import django  os.environ.setdefault("DJANGO\_SETTINGS\_MODULE", "XianPei.settings")  django.setup()  from goods.models import GoodsType,IndexGoodsBanner,IndexPromotionBanner,IndexTypeGoodsBanner  from django\_redis import get\_redis\_connection  *# 创建一个Celery类的实例对象*  app = Celery('celery\_tasks.tasks', broker='redis://192.168.199.250:6379/8')  *# 定义任务函数*  @app.task  def send\_register\_active\_email(to\_email, username, token):  *'''发送激活邮件'''*  *# 组织邮件信息*  subject = '鲜配欢迎信息'  message = ''  sender = settings.EMAIL\_FROM  receiver = [to\_email]  html\_message = '<h1>%s, 欢迎您成为鲜配注册会员</h1>请点击下面链接激活您的账户<br/><a href="http://127.0.0.1:8000/user/active/%s">http://127.0.0.1:8000/user/active/%s</a>' % (username, token, token)  send\_mail(subject, message, sender, receiver, html\_message=html\_message)  time.sleep(5) |
| **路径celery\_tasks/tasks.py** |

**配置如图：**

|  |
| --- |
|  |

**测试结果如图：**



**如果我使用用户的信息直接发送出去很容易暴露自己数据区内信息，所以需要做安全处理。**

4.3.2 用户ID信息加密

**本次毕业设计加密使用的是itsdangerous 官方网站：**<https://itsdangerous.palletsprojects.com/en/1.1.x/>

**参考文章：**

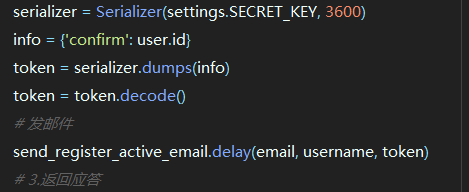
<https://www.jianshu.com/p/f1dc0cb67a71>

**一般的，邮件激活时，激活链接需要一个携带用户信息的url，来判断是激活的哪一个用户，且用户信息必须加密。python中可以使用 itsdangerous 来进行加解密。**

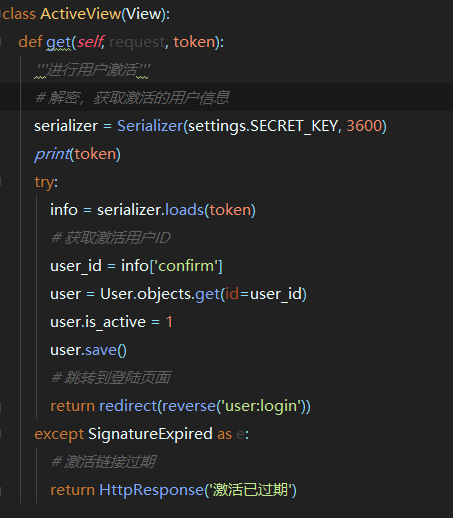
from itsdangerous import TimedJSONWebSignatureSerializer as Serializer

from itsdangerous import SignatureExpired

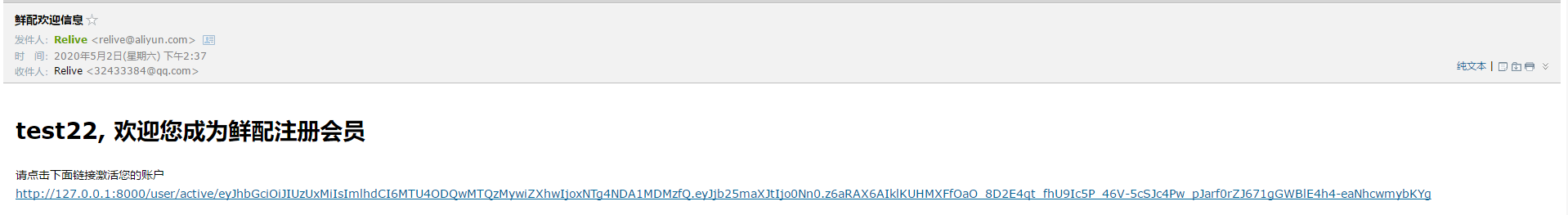
引入**itsdangerous到项目**



加密用户信息ID



**解密用户信息id**



**加密后效果图**

4.3.3 用户信息保存

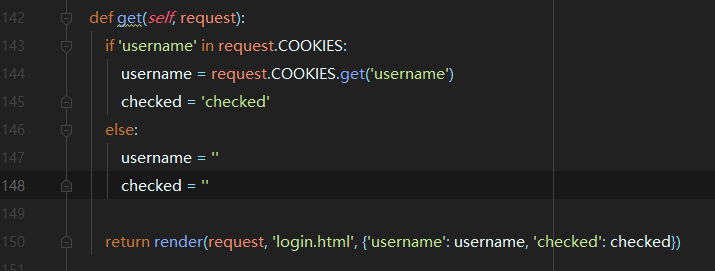
如果用户想要记住自己的用户信息就需要把相关信息存储在浏览器上本设计配置了redis作为Django缓存和session后端



S**ettings中的reids配置项**



判断是否点击了记住用户名



**自动填写账号密码**

4.3.4 用户登出

既然用户能登入自己的账号，那必须要登出这个账号这才是一个完整的用户系统。



**用户登出处理**

现在用户只需要按下退出按钮，就可以把账号退出去了。

**4.4 其他安全处理与其他功能**

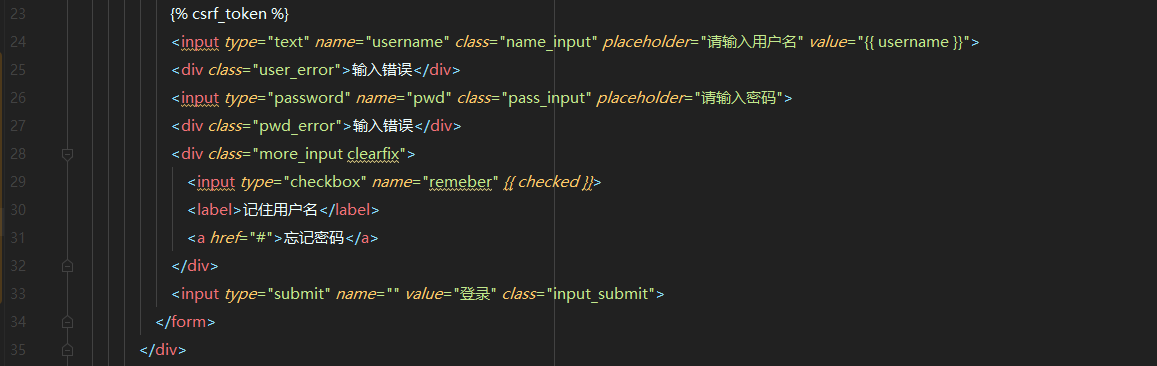
### **Django内置了安全设置功能，使开发者很轻松就可以得到基础的保护其中我使用跨站请求伪造保护。**

4.4.1 跨站请求伪造保护

**官方说明：**<https://docs.djangoproject.com/en/3.0/ref/csrf/>



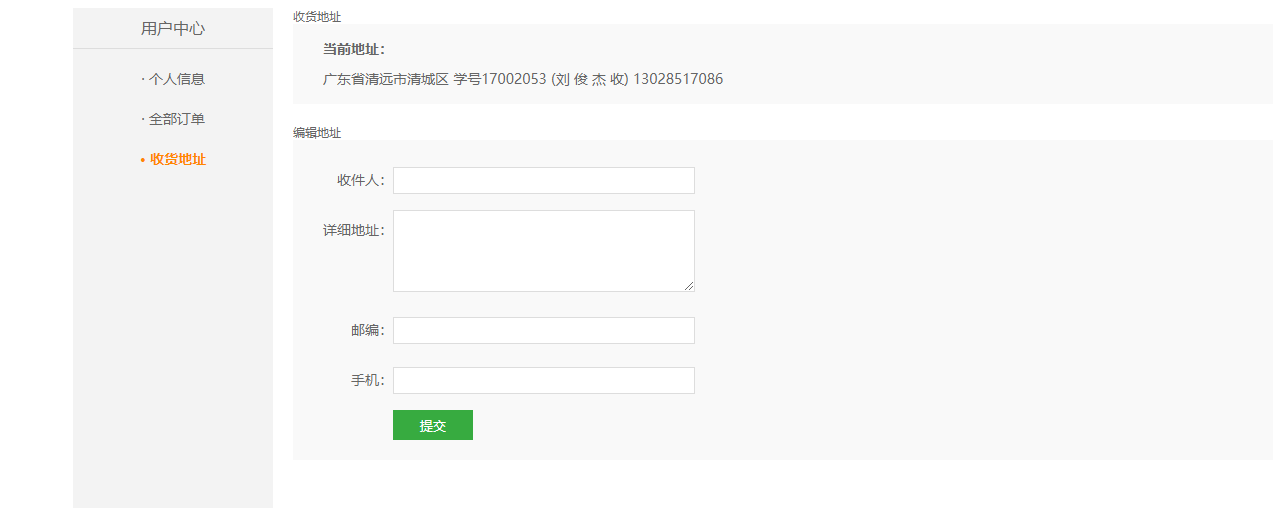
按照说明我只需要在防止跨站攻击的页面设置csrf防护就可以了



**图4.4.1**

### 4.4.2 地址管理功能

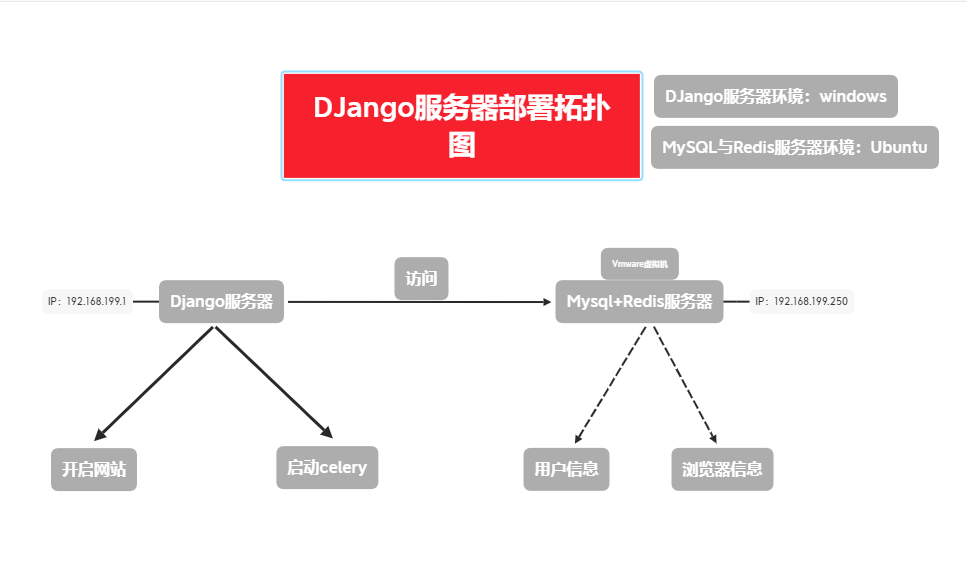
为了方便用户下单网站设置了用户新建地址功能，用户可以自主填写地址自主下单。



**地址管理效果图**

**第5章总结**

**本电子商务系统部署如下：**



环境部署

包版本：

|  |
| --- |
| **python包版本** amqp==2.5.2  billiard==3.5.0.5  celery==4.1.1  certifi==2019.11.28  chardet==3.0.4  Django==1.8.2  django-celery==3.1.17  django-redis==4.8.0  django-tinymce===2.6.0  dnspython==1.16.0  eventlet==0.25.1  fdfs-client-py===1.2.6  greenlet==0.4.15  idna==2.9  importlib-metadata==1.5.0  itsdangerous==1.1.0  kombu==4.6.8  monotonic==1.5  mutagen==1.44.0  Pillow==7.0.0  PyMySQL==0.9.3  pytz==2019.3  redis==3.4.1  requests==2.23.0  setuptools==45.2.0  six==1.14.0  urllib3==1.25.8  vine==1.3.0  wheel==0.34.2  zipp==3.1.0 |

|  |
| --- |
| 运行方法：  Windows：   1. 执行 python manage.py runserver 开启Django服务器运行网站 2. 执行celery -A celery\_tasks.tasks worker -l info -P eventlet 开启异步处理邮件   Ubuntu：   1. 、sudo redis-server /etc/redis/redis.conf 运行redis服务器 我的配置文件在/etc/redis/redis.con 2. Service mysql start |

**5.1遇到问题总结：**

1. 因为电子商务类型网站存在大量图片，如果放上服务器上面加载很慢很慢而且还会重复，本作品准备使用FDFS服务器作为图片存储服务器在寻觅了网站很多教程之后始终存在一些问题，而且很是复杂又是因为Django开发时版本与FdFs版本不兼容所致，所以中途暂停该开发。
2. 本应该制作的支付系统因为开通支付渠道的付款跳转需要商家/个体户营业执照才可以申请，所以被迫中止该开发。
3. GitHub项目链接：<https://github.com/Removechroot/Graduation-Project>

**5.2结论：**本文利用Django框架，结合Mysql数据库设计了简易的电子商务网站系统。使用Django框架，可以在短时间内创建高品质、易维护和数据库驱动的Web应用程序，提高了开发效率，降低了系统开发的复杂度，便于系统的更新和维护。

**5.3参考文献：**

[1] 何敏煌，Python新手使用Django架站的16堂课，北京：清华大学出版社，2017-5

[2] 顾翔，基于Django的电子商务网站设计，北京：清华大学出版社，2018

[3] 齐伟，跟老齐学pythonDjango实战，北京：电子工业出版社，2017

**5.4致谢：**

在全电子商务建设过程中，老师和同学给了我很大的帮助。在设计过程中老师和同学给了我很多好的建议，很多问题都是在老师和同学的帮助下解决的。所以在这里特别地感谢我的毕业设计指导老师余方老师，她在百忙中给我细心的技术指导和很多好的建议。还有在大学三年里教给我知识的其他老师与及给我帮助的同学在这里一并表示感谢。

最后，再一次真诚地祝愿曾经给与我帮助和支持的老师和同学幸福开心！谢谢！