项目使用python3.6+flask 1.0.2的环境，关于flask的ORM不像Django那么好操作，所有使用原生的数据库语句进行操作

注册

注册输入对应的账号信息后，鼠标移开输入框会自动到后台查找数据库中是否已经存的在对应相同的信息，如果存在则不能进行注册，不存在才能注册，账号信息中一定要有一个用户唯一标识的信息(比如电话，邮箱)，在后台对用户的密码进行加密后存到数据使用werkzeug.security 中的generate\_password\_hash和check\_password\_hash对密码进行加密和解密。

登录

用户输入用户名和密码就可以进行登录，为了更好的用户体验，这里和注册时一样，在用户输入用户名后，鼠标移开输入框会自动到数据库中查找是否有用户名，没有则提醒用户输入错误，用户点击登录后会把账号密码提交到后台进行数据库查找，匹配密码时用check\_password\_hash方法进行输入密码和数据库中的密码散列值进行匹配。

(登录时要输入验证码)，登录后把用户数据存到session中，设置session的过期时间

忘记密码功能，如果用户忘记了密码，可以通过注册的电话或者邮箱来找回密码，通过电话或邮箱发送验证码，用户输入获取的验证码则可以进行密码的更改，

CSRF保护

给所有的视图开启CSRF保护，关于使用CSRF保护后的Ajax需要添加csrf保护请求头，如果使用的是from表单，则在表单中添加<**meta name="csrf-token" content="**{{ **csrf\_token**() }}**"**> 或者<**input type="hidden" name="csrf\_token" value="**{{ **csrf\_token**() }}**"**>，

首页

关于商城的首页，就是路由进入到首页的视图函数，然后到数据库中查找对应的数据，返回到首页中，这里需要注意的是flask不像Django的render函数可以返回多个字典，只能返回一个字典参数，所有建议创建一个大的字典返回到首页中，因为首页有许多不同的数据

搜索

关于搜索，商城一定有一个搜索功能，点击搜索后返回模糊查询匹配的数据，需要一个商品搜索后的页面，该页面应该有一些大商品的分类(点击不同的分类 页面不会整个刷新，而是更新下面div的分类内容)，商品的排序，价格或者销量，页面进行一个分页，具体根据搜索结果的展示页面所定。

关于搜索框要实现有搜索记录，和单条记录的删除(js或jQuery实现)，这里仿照京东的网页来写，没有登录的话点击搜索框则不会显示搜索记录，而登录后则会显示用户的搜索记录，

点击搜索框则会用Ajax的方式到后台去查找用户的搜索记录，(关于Ajax可以需要更改url的链接，不然浏览器可能会缓存js而不能进行动态的更新),然后显示在搜索输入框下面，

(如果可以的话，尝试实现搜索的匹配，输入关键字会出现匹配类似的信息)

用户点击搜索后，后台会自动把这条搜索记录添加到该用户的搜索记录数据库中，如果用户没有登录则不会添加

搜索后的商品展示页面，看不到的商品不加载商品图片，只有滚动到可视范围内才加载图片，可以使用jQuery的懒惰加载插件

商品详情

点击对应商品会进入到商品详情页面（根据商品id去查找），商品详情页面会有具体的商品信息，商品详情图片，

商品可以进行收藏和添加到购物车(需要进行判断用户是否登录，如果没有登录会提醒进入到登录页面)，

商品详情下面有该商品的用户评论，（尝试实现用户的图片评论），评论有对应的评论表？

收货地址(用户中心)

收货地址用一个单独的表存储，里面有用户的一些基本信息，电话，邮箱，地址，收货人姓名，为了用户信息的保密，展示用户信息时，如电话: 188\*\*\*\*5853,按照这样的方式展示，邮箱也是如此。每个用户最多只能存10个收货地址

后台管理

只有管理员才能进入到后台管理界面，对后台所有数据库的操作，不是管理员进入到后台管理的登录页面进行登录会提示非管理员(或者跳转到商城的首页)

用户中心

用户的头像展示，存在一个用户头像字段，用户注册时给用户一个默认头像，用户可以上传自己的图片作为头像，上传的图片转为二进制数据流存到数据库中？

商品图片，用户上传的头像，其他方面的图片。目前业界存储图片有两种做法：   
1、 把图片直接以二进制形式存储在数据库中   
一般数据库提供一个二进制字段来存储二进制数据。比如MySQL中有个blob字段。Oracle数据库中是blob或bfile类型

2、 图片存储在磁盘上，数据库字段中保存的是图片的路径。

笔记：关于所有div的显示和隐藏都是用style中的display: block, display: none，比如广告的展示和点击去除，都是用display把广告对应的div显示或者隐藏起来，代码其实都还在网页中，而关于搜索历史的