Lab2部分

1. netshop.py

函数：

checkInfo(info)

功能：接收用户信息，检查是否符合规范（见lab2，P2-3），如果符合，返回1，如果有至少一项不符合，返回0。

端口：

（1）/store/request 开店申请页面对应端口，后端没有

（2）/store/request/commit

从前端接收以下信息：userName,shopName,goodsType,identificationID,shopIntro,address,initialFund

并补全开店时间为当日。

之后调用checkInfo()进行信息检查。如果信息符合要求，补全submitStatus为Success，并向前端返回message = "开店请求已受理。"，并向商户显示开店申请成功页面的html页面；如果不符合要求，向前端返回error = "用户信息不符合规范，请按说明修改后再提交。"，并返回开店申请页面。

p.s:我还没有写储存表格信息的部分，所以返回开店申请页面后所有数据会直接清空，用户体验极差。

1. checkOpenningRequest.py

端口：

（1）/admin/checkOpenningRequest

调取数据库中所有submitStatus=Success的开店申请表的所有信息，并返回result（二重元组）至前端。

（2）/admin/checkOpenningRequest/Passed

管理员点击“通过”后跳转到这个页面。

将开店申请表格中的所有信息加入商店列表这一数据表，并在开店申请的数据表内更新submitStatus='Passed'。

（3）/admin/checkOpenningRequest/Sentback

将在开店申请的数据表内更新submitStatus='sentBack'。

Lab3部分

1. lab3\_part3\_merchant.py

函数：

1. checkCommodityInfo(info)

检查提交的商品信息是否符合要求（见lab3，P5），以及是否为空。如果符合要求以及全不为空，返回1，如果至少有一项不符合或为空，返回0。

（2）getGoodsList()。从flask-caching中取出商户的userName，拼到“goodListOf”的后面，组成该商户的商品列表表名，后返回该表名。

端口：

（1）商品列表页面有关内容

（1-1）/merchant/goodsList

调取数据库中对应商户的商品列表，并向前端返回result（二重元组）。

（1-2）/merchant/goodsList/add

向商户显示增加商品申请表的页面，后端没有。

（1-3）/merchant/goodsList/add/commit

接收商户在开店申请表中填写的数据，包含以下内容：userName,shopName,name,discription,picture,price,stock；并调用checkCommodityInfo进行信息检查（规范见lab3，P5）。如果信息符合规范，就向前端返回message = "新增商品请求已受理。"，后向用户显示一个提示申请成功的html页面；如果至少一项不符合，向前端返回error = "信息不符合规范，请按说明修改后再提交。"。

（1-4）/merchant/goodsList/alter

从前端接收商品名称，从数据库的商户的商品列表当中调取对应数据并返回到前端。

（1-5）/merchant/goodsList/alter/commit

基本逻辑和（1-3）类似，只是信息符合规范，会向前端返回的信息改为"修改商品信息请求已受理。"

（1-6）/merchant/goodsList/downShelf

商户点击商品下架后使用此接口。

向前端接收商品名称后，查询该商户商品列表中对应数据条目，将该条目的上架状态一项改为down。

（1-7）/merchant/goodsList/delete

商户点击删除商品后使用此接口。

向前端接收商品名称后，删除该商户商品列表中对应数据条目。

（2）商品上架（新增商品）申请列表有关内容

（2-1）/merchant/goodsList/upShelfRequestList

从数据库的上架申请表中查询该商户的所有申请条目，返回给前端。

（2-2）/merchant/goodsList/upShelfRequestList/delete

调用储存在flask-caching模块中的商户的用户名，从前端接收商品名称，后从数据库的上架申请表中直接删除该条目。

（2-3）/merchant/goodsList/upShelfRequestList/alter

调用储存在flask-caching模块中的商户的用户名，从前端接收商品名称，后从数据库的上架申请表中调取商品信息，返回给前端。

（2-4）/merchant/goodsList/upShelfRequestList/alter/commit

基本逻辑和（1-3）类似，使用数据库中上架申请的数据表，信息符合规范，会向前端返回的信息改为"修改新增商品申请表成功。"

1. 修改商品信息申请列表有关内容

（3-1）/merchant/goodsList/MCIRequestList

基本逻辑和（2-1）类似。

（3-2）/merchant/goodsList/MCIRequestList/delete

基本逻辑和（2-2）类似。使用数据库中的修改商品信息申请这一数据表。

（3-3）/merchant/goodsList/MCIRequestList/alter

基本逻辑和（2-3）类似。使用数据表同（3-2）。

（3-4）/merchant/goodsList/MCIRequestList/alter/commit

基本逻辑和（2-4）类似.使用数据表同（3-2），信息符合规范，会向前端返回的信息改为"对修改商品信息申请表的修改成功。"

p.s:我觉得合并成一个函数还更麻烦，所以就没合并了。

1. lab3\_part3\_admin.py

函数：

1. updateSubmitStatus(formName,submitStatus)。

调用时，需要前端传入数据表名称和目标状态的名称。

该函数首先从前端接收商户的用户名和经审核的商品的名称，在数据库内名为formName的数据表中更新对应条目的submitStatus。

1. selectListwithSubmitStatus(formName,submitStatus)

在数据库内名为formName的数据表中，查询所有提交状态为submitStatus的数据条目。返回值是查询结果组成的二重元组。

端口：

1. 审核商品上架申请

（1-1）/admin/checkUpShelfRequest

调用selectListwithSubmitStatus("upShelfRequest","Submitted")，也就是查询upShelfRequest这一数据表中提交状态为Submitted的所有条目，后将查询结果返回给前端。

（1-2）/admin/checkUpShelfRequest/Passed

管理员点击审核通过按钮后，调用updateSubmitStatus("upShelfRequest","Passed")，也就是在upShelfRequest这一数据表中，将管理员选中的条目的申请状态更新为Passed。

（1-3）/admin/checkUpShelfRequest/sentBack

基本逻辑和（1-2）类似，只是第二个参数修改为Sentback。

1. 审核商品信息修改申请

（2-1）/admin/checkMCIRequest

基本逻辑和（1-1）类似，只是第一个参数修改为MCIRequest。

（2-2）/admin/checkMCIRequest/Passed

基本逻辑和（1-2）类似，只是第一个参数修改为MCIRequest。

（2-3）/admin/checkMCIRequest/sentBack

基本逻辑和（1-3）类似，只是第一个参数修改为MCIRequest。

3、lab3\_part4.py

函数：

（1）askForGoodsStatus(merchantName,goodsName)

调用时需要输入商品对应的商户名和商品的名称。

该函数会从数据库的，商户的商品列表中，查询上架状态，并返回该上架状态（up/down），如果商品不存在，返回None。

（2）getCart()

基本逻辑和lab3\_part3\_merchant.py中的函数（2）类似，需要拼贴的字符串改为“cartOf”。

端口：

（1）/user/cart

从数据库的用户购物车这一数据表中，调取所有数据后，利用购物车中每个商品条目中merchantName,goodsName这两个数据，调用askForGoodsStatus(merchantName,goodsName)，更新每个商品的上架状态，之后重新调取一次购物车中所有商品的数据，返回给前端。

（2）/addingToCart。

p.s：这里默认了商品显示页面会缓存商品信息，所以写了前端向后端传输信息。如果不是的话我之后再改。

从前端接收商品信息，并在数据库中该用户的购物车这一数据表中填入新的一条数据。

（3）/user/cart/delete。

p.s：前端要把用户要删除的商品的店主名和商品名打包成二重元组传到后端，以兼容单个删除和批量删除。

从前端接收上述二重元组，并在该用户的购物车中删除对应条目。