期末大作业报告

刘振宇 元培学院 1700017853

**作业概况**

期末大作业要求完成一个股票在线交易系统，并对系统的功能提出了一些基本的要求，同时鼓励自己添加创新功能。首先需要在此承认，虽然未与同班同学进行组队，但是这次大作业并不是由我一个人完成的，大作业的基本要求中很多的函数定义、网页搭建均是在另一位计算机专业的同学帮助下完成的。但是大作业中所有的拓展功能均是由我自己一人完成的。

我们先按照CS50课程网站（https://docs.cs50.net/2018/x/psets/7/finance/finance.html）上的作业要求，定义函数、创建网页，完成了一个基本满足课程要求的基本交易系统，之后，在此系统的基础上，我又自行添加了一些其他自己设计的元素。

**源代码结构**

程序储存在文件夹finance中，其中各个不同的文件以及其作用，内部代码结构如下：

finance/

application.py：

整个系统的核心应用程序。在application.py中，首先引入了一系列程序包，设置

了API KEY，配置了数据库等，这些工作在作业提供的基础代码中已经帮我们完成，我们的工作主要在于自己定义了完成各项具体功能的函数（按代码出现的先后顺序）：

* index(): 从数据库中调取用户当前所有拥有的所有股票的名称，数量，股价，总价等数据。
* buy(): 购买特定数量的特定股票，并且更新购买后用户的财产信息，并储存交易记录。
* history(): 从数据库中调取用户的交易历史信息。
* login(): （作业原本已经定义好了）进行用户登录操作
* logout(): （作业原本已经定义好了）退出登录操作。
* quote(): 查询对应股票的价格。
* register(): 注册用户，并把信息储存在数据库中。
* sell(): 出售一定数量的股票，并记录交易后的财产信息和交易记录。

helpers.py：

定义了一些重要的辅助函数，协助程序完成各项功能。helpers.py中的所有函数在作业提供的基础代码中均已经定义完成，可以使用。因此完成作业过程中未对helpers.py进行改动。

* apology(message, code=400): 当出现错误时报错，message为错误信息，code反应错误类型，默认值400。
* login\_required(f): 确保application.py中的各个功能只能在登陆后使用。
* lookup(symbol): 从网上查询对应股票的信息，形参symbol为对应股票的代号。
* usd(value): 将数值转化成美元，形参value为价格。

finance.db.:

数据库文件，其中包含了3个不同的表：

* users: 储存已经注册过的用户的信息
* transactions: 储存用户的交易记录
* portfolio: 储存用户当前拥有的资产信息。

finance/static/

style.css: 对网页排版的控制文件，作业提供的原始代码中已经完成，未进行修改。

finance/templates/

此文件夹下储存着一系列html文件，分别对应系统在执行对应功能时的网页排版，除layout.html为完整定义之外，其他均直接套用layout.html进行修改。在此不再一一罗列。这里需要指出，网页中的表格，按钮设计均由Bootstrap（https://getbootstrap.com）直接复制代码而来，并不是本人的工作。

**作业中遇到的问题及解决**

1. 作业中遇到最大的问题（困难）来自于语法不熟悉，无论是对于python本身还是flask框架，HTML语言，SQL数据库操作，都只是在课堂上听老师讲解而从未进行过实践操作，所以在完成作业时往往是思路很明确，但是无法用程序语言写出。整个作业过程中最大的障碍就是来源于此。

对这个问题，主要的解决方法是在网络上查询相关的说明，查看样例，以及请教其他同学。好在作业的主要部分，即不同功能函数的定义和对应网页的搭建，均为重复性较高的工作。在完成了register，quote功能函数定义以及对应网页设计后，对于语法逐渐熟悉，后续工作变得顺利很多。

1. 在index函数和history函数的定义过程中，需要遍历数据库中用户拥有的所有股票信息，以及所有的交易历史，并通过表的形式展示在网页上。由于二者的数量均为不确定，无法知晓具体的表的行数。在设计index和history这两个网页时苦恼了很久。

起初想到的方法是开一个行数足够大的表，把这些信息逐行填充，未填充的多余的行就空置。但是这种方法不够美观，也不够实用（何为足够大？），之后在同学的提醒下学习了html语言中的for循环，把数据库中的表当做一个迭代器，利用for循环进行表格的创建，从而保证了表格的行数和数据库中的数据条数匹配。

对应部分源代码如下（以history.html为例）：

<tbody>

{% for stock in stocks %}

<tr>

<th scope="row">{{stock.stock}}</th>

<td>{{stock.shares}}</td>

<td>{{stock.price}}</td>

<td>{{stock.total}}</td>

</tr>

{% endfor %}

在sell界面中，选择出售的股票时也遇到了类似的问题，即用户拥有的股票数量不确定，同样通过for循环，表做迭代器实现了功能。

1. 在计算一支股票的总价时，用每股价格乘以股数，由于浮点数的计算误差，结果中总会出现很小的误差，如20\*100.79= 2015.8000000000002，而单位为美元时，最多保留两位小数。因此需要对所得数字进行修正。

修正方法：调用round()函数，并把形参n设置为2（2位小数）。虽然round()函数在处理浮点数时误差较大，但已经满足在计算股价时的精度需求。

源代码：

value=round(lookedup["price"] \* int(request.form.get("shares")),2)

1. 在输出表时，由于在html中直接把数据库中的表当做迭代器，输出的均为数据库中的数据，而表中对price和value的类型限制为REAL，所以输出price和value时只能输出浮点数而没有单位，同时由于usd(value)定义于helpers.py中，无法再html中使用usd()来带上单位。

我想到的第一个解决方法是在表中新建一列，类型限制为TEXT，专门储存加上单位后的price和value，后来觉得实行起来有些麻烦，于是就退而求其次，在表头上指示该列的单位为$，而不再每个数值后面带单位。

**新功能：**

1. 股票检索能力的增强

大作业本身要求从AlphaVantage上检索对应股票的价格，但是这种检索只能返回得到股票的价格，而对于其走势，成交量，价格区间，等等很多重要信息都没有反应。为了让用户得到更为详细的股票信息，我在quote界面上另外添加了一个搜索框，用于搜索股票信息。搜索直接利用的是新浪股票（http://finance.sina.com.cn/stock/）的搜索引擎，得到的数据也均来自于新浪股票，点击搜索后，会自动进入新浪股票中对应股票的信息界面，而该界面可以提供较为详尽的股票信息资料。

实现功能的方法老师上课有讲授过（以百度和Google为例），但是由于新浪股票的首页较为复杂，而且搜索页面和首页的url差异较大，搜索时动用的变量较多，因此具体的实现过程比上课讲授过的要复杂一些。

我先在新浪股票的主页上（http://finance.sina.com.cn/stock/）搜索msft，得到对应搜索结果网页的url为：

http://stock.finance.sina.com.cn/usstock/quotes/MSFT.html

从这个url中我们无法看到对应的搜索函数名，变量名，故无法再自己的网站中添加链接调用这个搜索功能。考虑到MSFT为正确的股票名，搜索结果网页可能有简化，于是我改在首页中搜索“哈哈哈哈”（随意输入一个不为股票名的东西），得到搜索结果的url为：

http://biz.finance.sina.com.cn/suggest/lookup\_n.php?country=&q=哈哈哈哈&name=哈哈哈哈&t=keyword&c=all&k=哈哈哈哈&range=all&col=1\_7&from=channel

从这个url中我们可以看到，以“哈哈哈哈”为值的变量一共有3个，变量名分别为q，name和k，我们并不知道那一个为对应的搜索关键词，因此我采用了逐个尝试的办法。当我在地址栏中输入

http://biz.finance.sina.com.cn/suggest/lookup\_n.php?q=哈哈哈哈

的时候，出现了搜索“哈哈哈哈”的界面，但是当我把其中的q换成name和k时，均不能正确搜索，据此，我确定了新浪股票的搜索函数对应url为：

http://biz.finance.sina.com.cn/suggest/lookup\_n.php

搜索关键字变量名为q，据此，我就可以在quote的HTML文件中添加对应的搜索链接，对应的源代码为：

<form action="http://biz.finance.sina.com.cn/suggest/lookup\_n.php" method="get">

<p><h4>Quote more information about the stock</h4></p>

<input autocomplete="off" autofocus class="form-control" name="q" placeholder="Symbol" type="text"/>

<button class="btn btn-info" type="submit">Quote</button>

</form>

从而实现了检索功能。

1. 实用性超链接的添加：

在每一次完成查询、购买、售卖操作之后的quoted，bought和sold界面中，除了显示此次的交易信息之外，我在下方加入了几个用户可能要进行的操作链接：再次购买（售卖），查看我的交易记录，查看我现在的资产情况，这样可以方便用户的操作，优化用户的使用体验。

实现方法也很简单，直接在对应的网站html文件最后添加对应的超链接即可。对应功能虽然可以通过点击上方菜单栏中的buy，history，sell等链接完成，但是在交易信息下方添加对应链接更适应于用户的使用习惯，更利于用户使用。

1. 股价信息来源的说明：

在查询股价信息时，我们说明了股票价格信息的来源：Alphavantage，查询股票的更多信息时，指出了信息来源：新浪股票，并提供了对应的链接。

**未能实现的新想法：**

在实际完成作业的过程中，由于个人能力有限，还有一些想要实现的功能最终没能完成，在此列举如下。

1. 在quote界面上提供关于股票的更多信息。

在初始的quote界面中，我们只能查询到对应股票的当前售价，对于其他的信息如价格走势、成交量、价格区间等等都没有涉及。为了解决这个问题，我在quote界面上增加了搜索链接，允许用户在网页上检索更多信息，但是检索的结果是跳转到新浪股票的对应网页上，离开了原始的股票交易系统。

我构想的理想情况应该是在系统中查询对应的股票信息后，可以在系统内的页面上反应股票的具体信息。即从外部网站上抓取对应信息后，返回到自己的系统中自己设计的网页上去显示给用户。这涉及到从网站中定向抓取数据，并实现网页间的数据传输。由于自己定义的quote网页本身不带参数，无法通过修改url的方式完成，只能利用form表单传输，而我对相关的知识并不了解，所以未能实现此功能。

1. 快捷的购买、售卖操作。

在查询股票信息的quoted界面，我添加了购买的链接，指向对应的购买网页。理想情况下，在quoted页面中点击“购买”的链接，可以直接购买刚刚查询过的股票，即不需要再重新输入要购买的股票信息，系统会自动填充在上个界面刚刚查询的股票。但是这同样涉及到在不同的网页之间传递参数问题，并且同时涉及到input框的自动填充，程序未能实现。