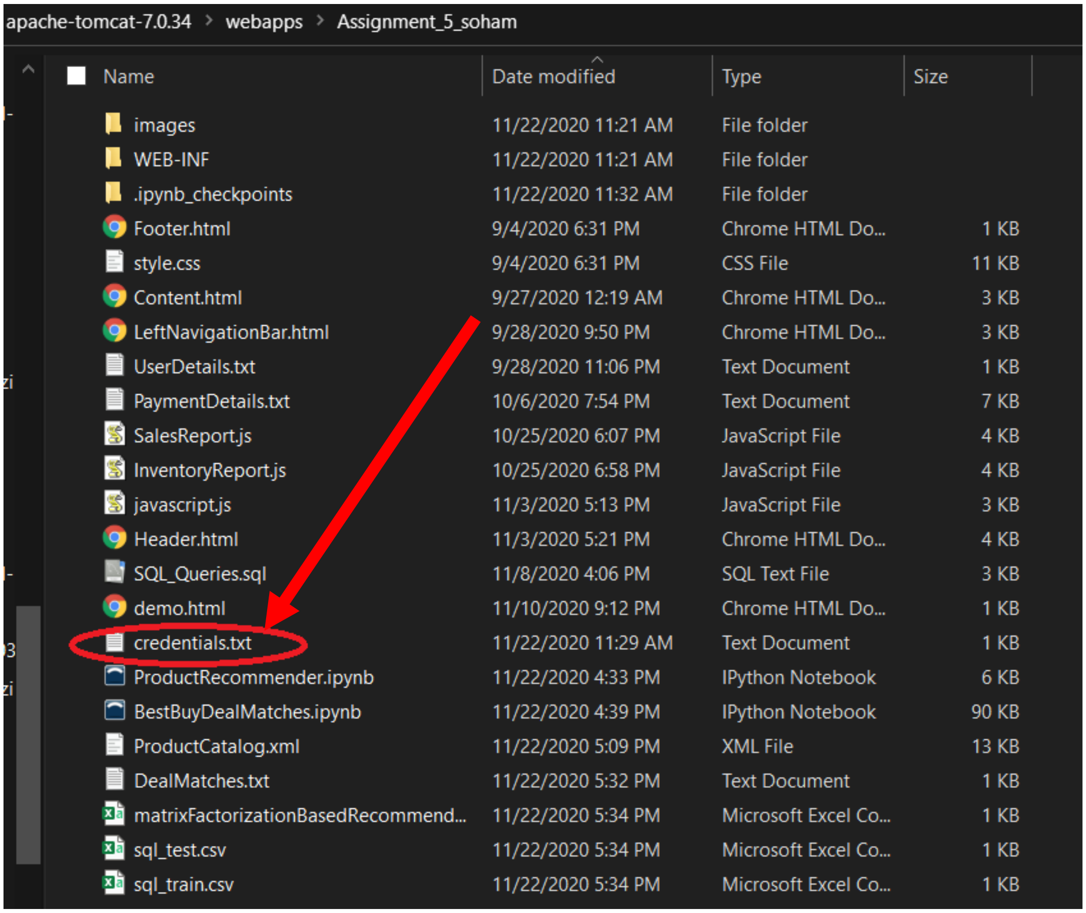
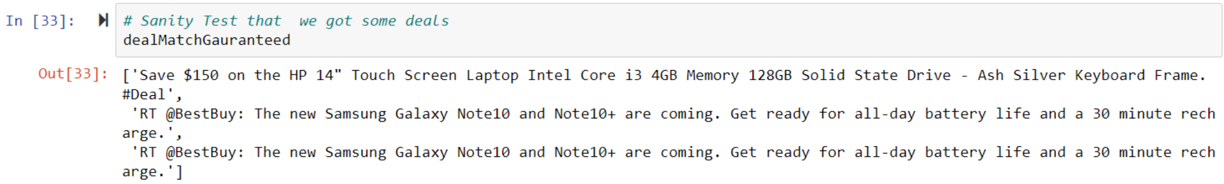
credentials.txt in the Home directory



apache-tomcat-7.0.34 > webapps > Assignment_5_soham

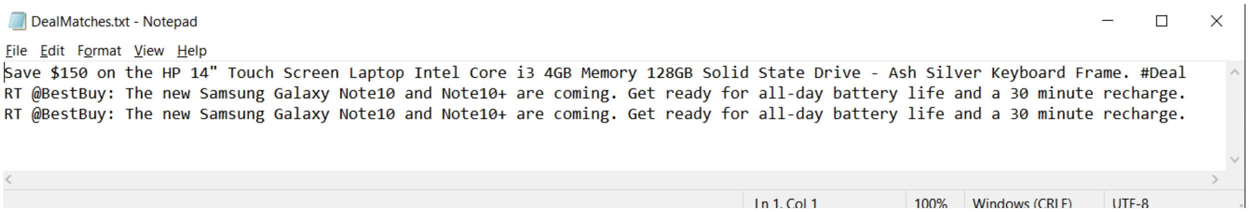
Name	Date modified	Type	Size
images	11/22/2020 11:21 AM	File folder	
WEB-INF	11/22/2020 11:21 AM	File folder	
.ipynb_checkpoints	11/22/2020 11:32 AM	File folder	
Footer.html	9/4/2020 6:31 PM	Chrome HTML Do...	1 KB
style.css	9/4/2020 6:31 PM	CSS File	11 KB
Content.html	9/27/2020 12:19 AM	Chrome HTML Do...	3 KB
LeftNavigationBar.html	9/28/2020 9:50 PM	Chrome HTML Do...	3 KB
UserDetails.txt	9/28/2020 11:06 PM	Text Document	1 KB
PaymentDetails.txt	10/6/2020 7:54 PM	Text Document	7 KB
SalesReport.js	10/25/2020 6:07 PM	JavaScript File	4 KB
InventoryReport.js	10/25/2020 6:58 PM	JavaScript File	4 KB
javascript.js	11/3/2020 5:13 PM	JavaScript File	3 KB
Header.html	11/3/2020 5:21 PM	Chrome HTML Do...	4 KB
SQL_Queries.sql	11/8/2020 4:06 PM	SQL Text File	3 KB
demo.html	11/10/2020 9:12 PM	Chrome HTML Do...	1 KB
credentials.txt	11/22/2020 11:29 AM	Text Document	1 KB
ProductRecommender.ipynb	11/22/2020 4:33 PM	IPython Notebook	6 KB
BestBuyDealMatches.ipynb	11/22/2020 4:39 PM	IPython Notebook	90 KB
ProductCatalog.xml	11/22/2020 5:09 PM	XML File	13 KB
DealMatches.txt	11/22/2020 5:32 PM	Text Document	1 KB
matrixFactorizationBasedRecommend...	11/22/2020 5:34 PM	Microsoft Excel Co...	1 KB
sql_test.csv	11/22/2020 5:34 PM	Microsoft Excel Co...	1 KB
sql_train.csv	11/22/2020 5:34 PM	Microsoft Excel Co...	1 KB

Matched tweets



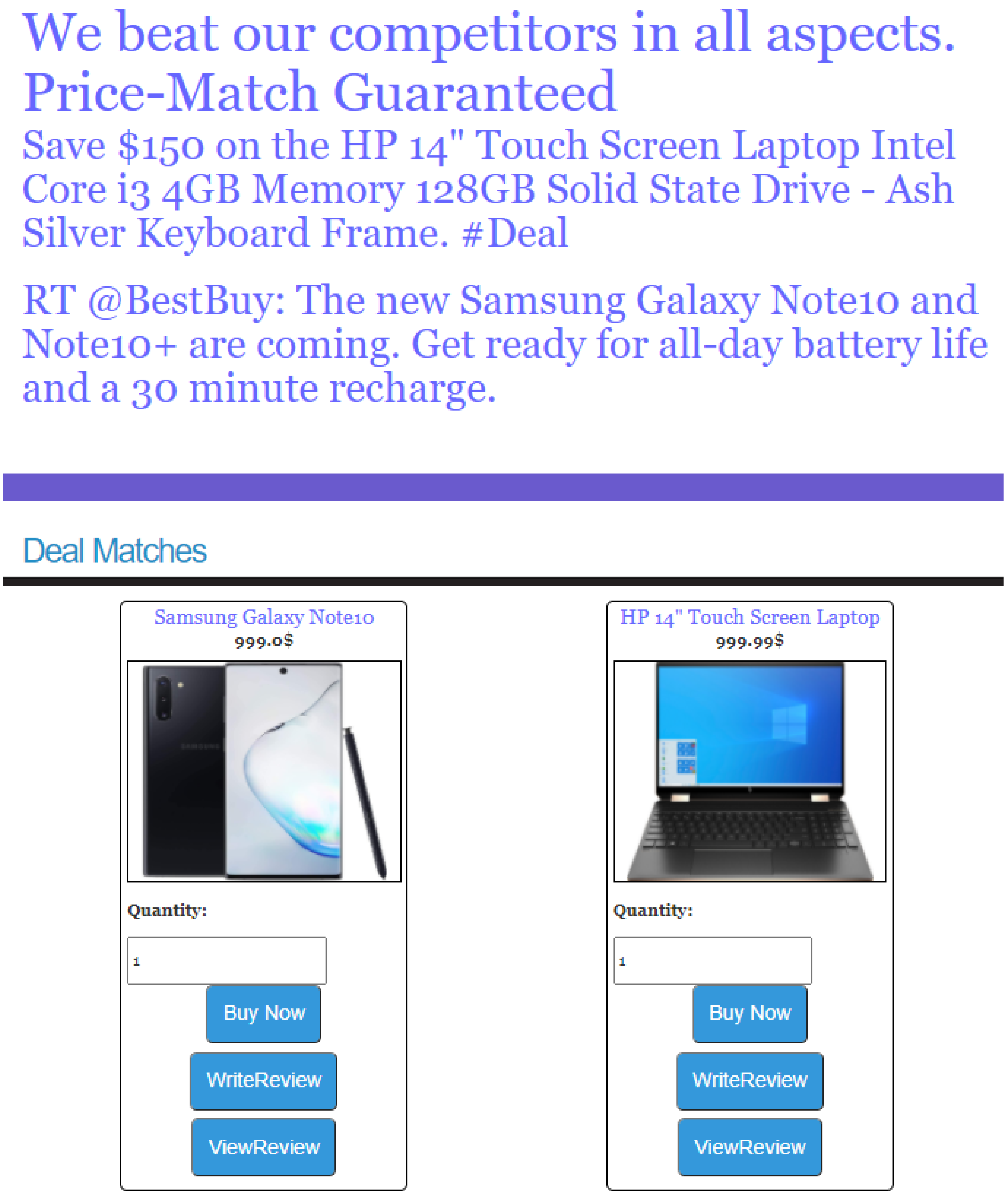
```
In [33]:  # Sanity Test that  we got some deals
          dealMatchGauranteed

Out[33]: ['Save $150 on the HP 14" Touch Screen Laptop Intel Core i3 4GB Memory 128GB Solid State Drive - Ash Silver Keyboard Frame.
         #Deal',
          'RT @BestBuy: The new Samsung Galaxy Note10 and Note10+ are coming. Get ready for all-day battery life and a 30 minute rech
         arge.',
          'RT @BestBuy: The new Samsung Galaxy Note10 and Note10+ are coming. Get ready for all-day battery life and a 30 minute rech
         arge.']
```

DealMatches.txt



```
Save $150 on the HP 14" Touch Screen Laptop Intel Core i3 4GB Memory 128GB Solid State Drive - Ash Silver Keyboard Frame. #Deal
RT @BestBuy: The new Samsung Galaxy Note10 and Note10+ are coming. Get ready for all-day battery life and a 30 minute recharge.
RT @BestBuy: The new Samsung Galaxy Note10 and Note10+ are coming. Get ready for all-day battery life and a 30 minute recharge.
```

In application



We beat our competitors in all aspects. Price-Match Guaranteed

Save $150 on the HP 14" Touch Screen Laptop Intel Core i3 4GB Memory 128GB Solid State Drive - Ash Silver Keyboard Frame. #Deal

RT @BestBuy: The new Samsung Galaxy Note10 and Note10+ are coming. Get ready for all-day battery life and a 30 minute recharge.

Deal Matches

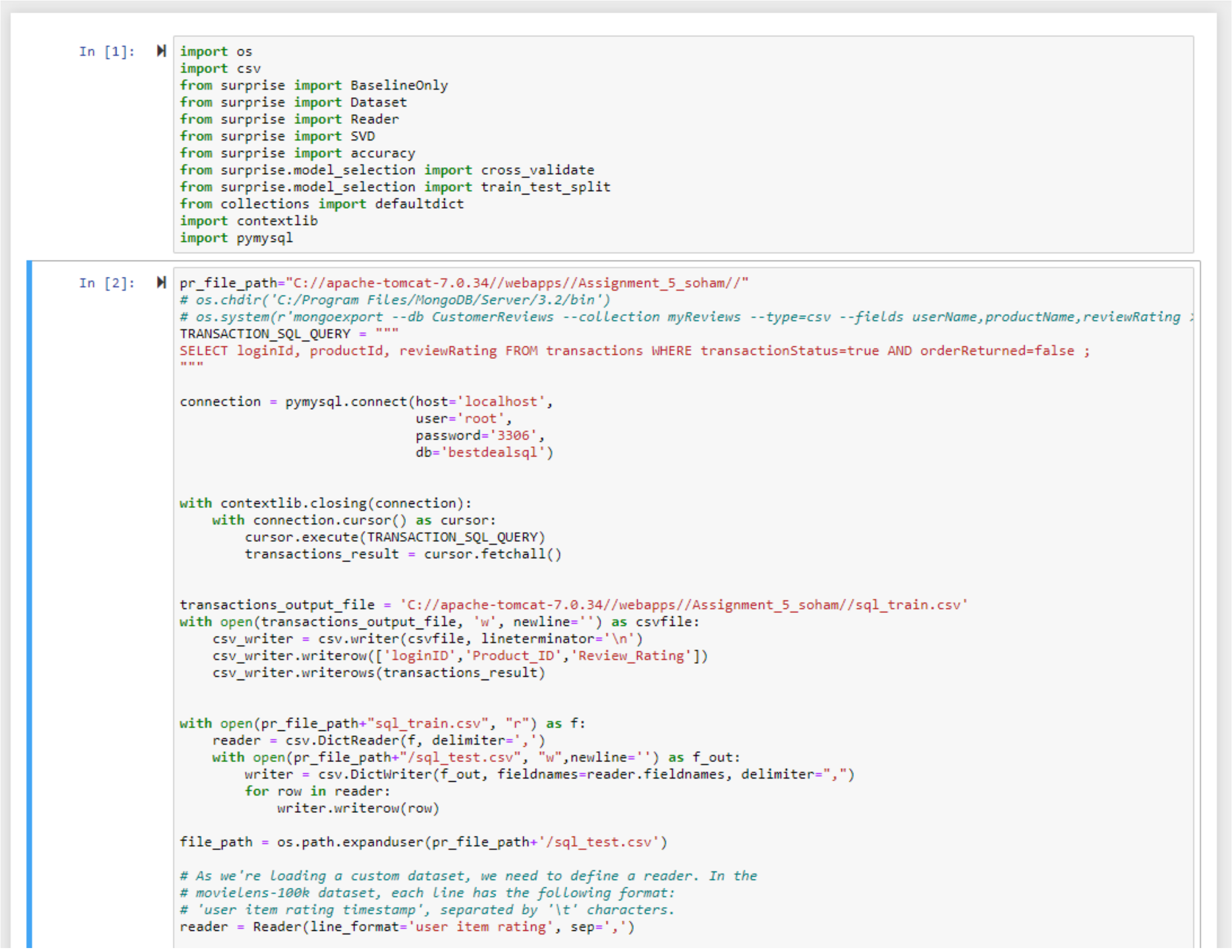
Samsung Galaxy Note10
999.0$

Quantity: 1

HP 14" Touch Screen Laptop
999.99$

Quantity: 1

Recommender script

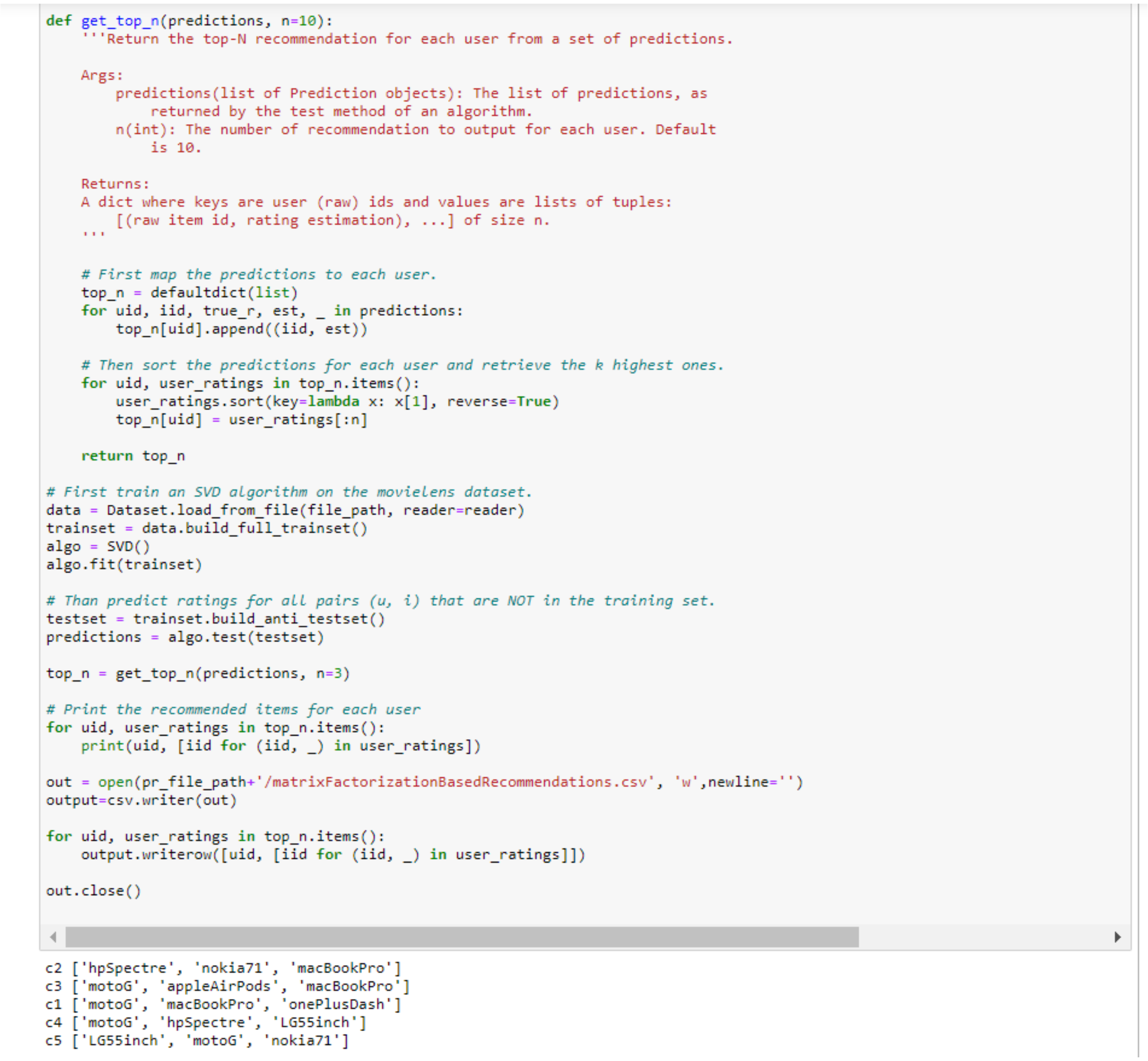


In [1]:

```
import os
import csv
from surprise import BaselineOnly
from surprise import Dataset
from surprise import Reader
from surprise import SVD
from surprise import accuracy
from surprise.model_selection import cross_validate
from surprise.model_selection import train_test_split
from collections import defaultdict
import contextlib
import pymysql
```

In [2]:

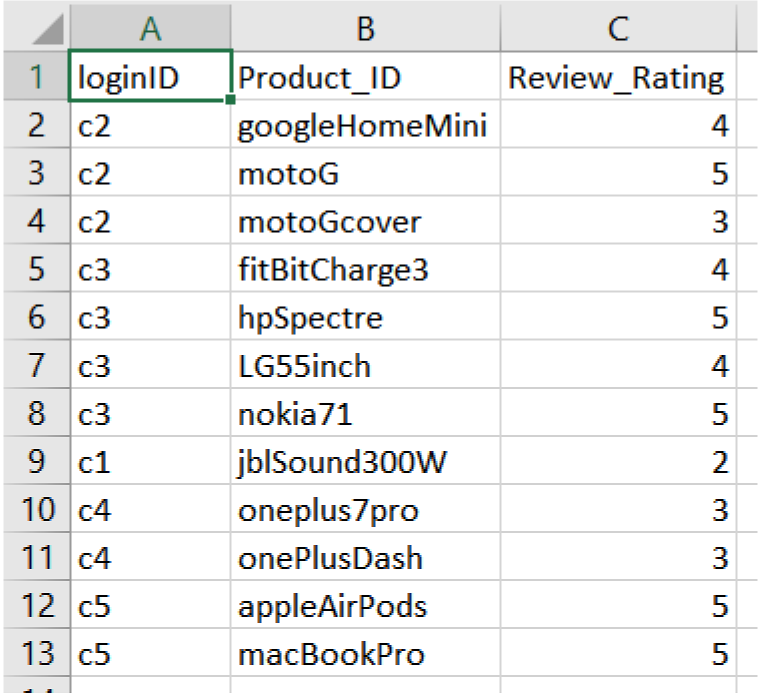
```
pr_file_path="C://apache-tomcat-7.0.34//webapps//Assignment_5_soham//"
# os.chdir('C:/Program Files/MongoDB/Server/3.2/bin')
# os.system(r'mongoexport --db CustomerReviews --collection myReviews --type=csv --fields userName,productName,reviewRating
TRANSACTION_SQL_QUERY = """
SELECT loginId, productId, reviewRating FROM transactions WHERE transactionStatus=true AND orderReturned=false ;
"""

connection = pymysql.connect(host='localhost',
                             user='root',
                             password='3306',
                             db='bestdealsql')


with contextlib.closing(connection):
    with connection.cursor() as cursor:
        cursor.execute(TRANSACTION_SQL_QUERY)
        transactions_result = cursor.fetchall()


transactions_output_file = 'C://apache-tomcat-7.0.34//webapps//Assignment_5_soham//sql_train.csv'
with open(transactions_output_file, 'w', newline='') as csvfile:
    csv_writer = csv.writer(csvfile, lineterminator='\n')
    csv_writer.writerow(['loginID','Product_ID','Review_Rating'])
    csv_writer.writerows(transactions_result)


with open(pr_file_path+"sql_train.csv", "r") as f:
    reader = csv.DictReader(f, delimiter=',')
    with open(pr_file_path+"/sql_test.csv", "w",newline='') as f_out:
        writer = csv.DictWriter(f_out, fieldnames=reader.fieldnames, delimiter=",")
        for row in reader:
            writer.writerow(row)

file_path = os.path.expanduser(pr_file_path+'/sql_test.csv')

# As we're loading a custom dataset, we need to define a reader. In the
# movielens-100k dataset, each line has the following format:
# 'user item rating timestamp', separated by '\t' characters.
reader = Reader(line_format='user item rating', sep=',')
```



```
def get_top_n(predictions, n=10):
    '''Return the top-N recommendation for each user from a set of predictions.

    Args:
        predictions(list of Prediction objects): The list of predictions, as
            returned by the test method of an algorithm.
        n(int): The number of recommendation to output for each user. Default
            is 10.

    Returns:
    A dict where keys are user (raw) ids and values are lists of tuples:
        [(raw item id, rating estimation), ...] of size n.
    '''

    # First map the predictions to each user.
    top_n = defaultdict(list)
    for uid, iid, true_r, est, _ in predictions:
        top_n[uid].append((iid, est))

    # Then sort the predictions for each user and retrieve the k highest ones.
    for uid, user_ratings in top_n.items():
        user_ratings.sort(key=lambda x: x[1], reverse=True)
        top_n[uid] = user_ratings[:n]

    return top_n

# First train an SVD algorithm on the movielens dataset.
data = Dataset.load_from_file(file_path, reader=reader)
trainset = data.build_full_trainset()
algo = SVD()
algo.fit(trainset)

# Than predict ratings for all pairs (u, i) that are NOT in the training set.
testset = trainset.build_anti_testset()
predictions = algo.test(testset)

top_n = get_top_n(predictions, n=3)

# Print the recommended items for each user
for uid, user_ratings in top_n.items():
    print(uid, [iid for (iid, _) in user_ratings])

out = open(pr_file_path+'/matrixFactorizationBasedRecommendations.csv', 'w',newline='')
output=csv.writer(out)

for uid, user_ratings in top_n.items():
    output.writerow([uid, [iid for (iid, _) in user_ratings]])

out.close()
```

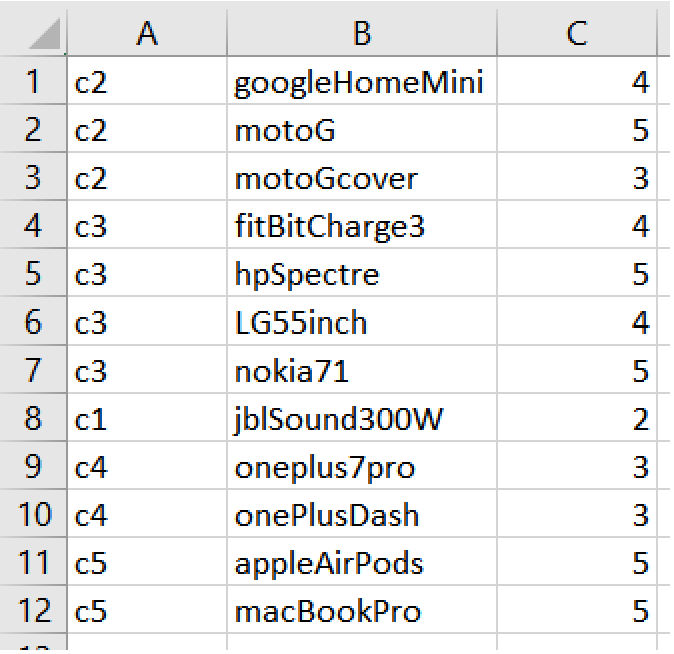
```
c2 ['hpSpectre', 'nokia71', 'macBookPro']
c3 ['motoG', 'appleAirPods', 'macBookPro']
c1 ['motoG', 'macBookPro', 'onePlusDash']
c4 ['motoG', 'hpSpectre', 'LG55inch']
c5 ['LG55inch', 'motoG', 'nokia71']
```

sql_train.csv



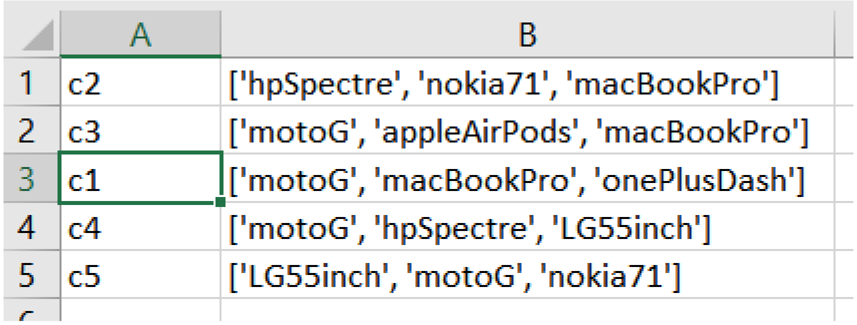
	A	B	C
1	loginID	Product_ID	Review_Rating
2	c2	googleHomeMini	4
3	c2	motoG	5
4	c2	motoGcover	3
5	c3	fitBitCharge3	4
6	c3	hpSpectre	5
7	c3	LG55inch	4
8	c3	nokia71	5
9	c1	jblSound300W	2
10	c4	oneplus7pro	3
11	c4	onePlusDash	3
12	c5	appleAirPods	5
13	c5	macBookPro	5

sql_test.csv



	A	B	C
1	c2	googleHomeMini	4
2	c2	motoG	5
3	c2	motoGcover	3
4	c3	fitBitCharge3	4
5	c3	hpSpectre	5
6	c3	LG55inch	4
7	c3	nokia71	5
8	c1	jblSound300W	2
9	c4	oneplus7pro	3
10	c4	onePlusDash	3
11	c5	appleAirPods	5
12	c5	macBookPro	5

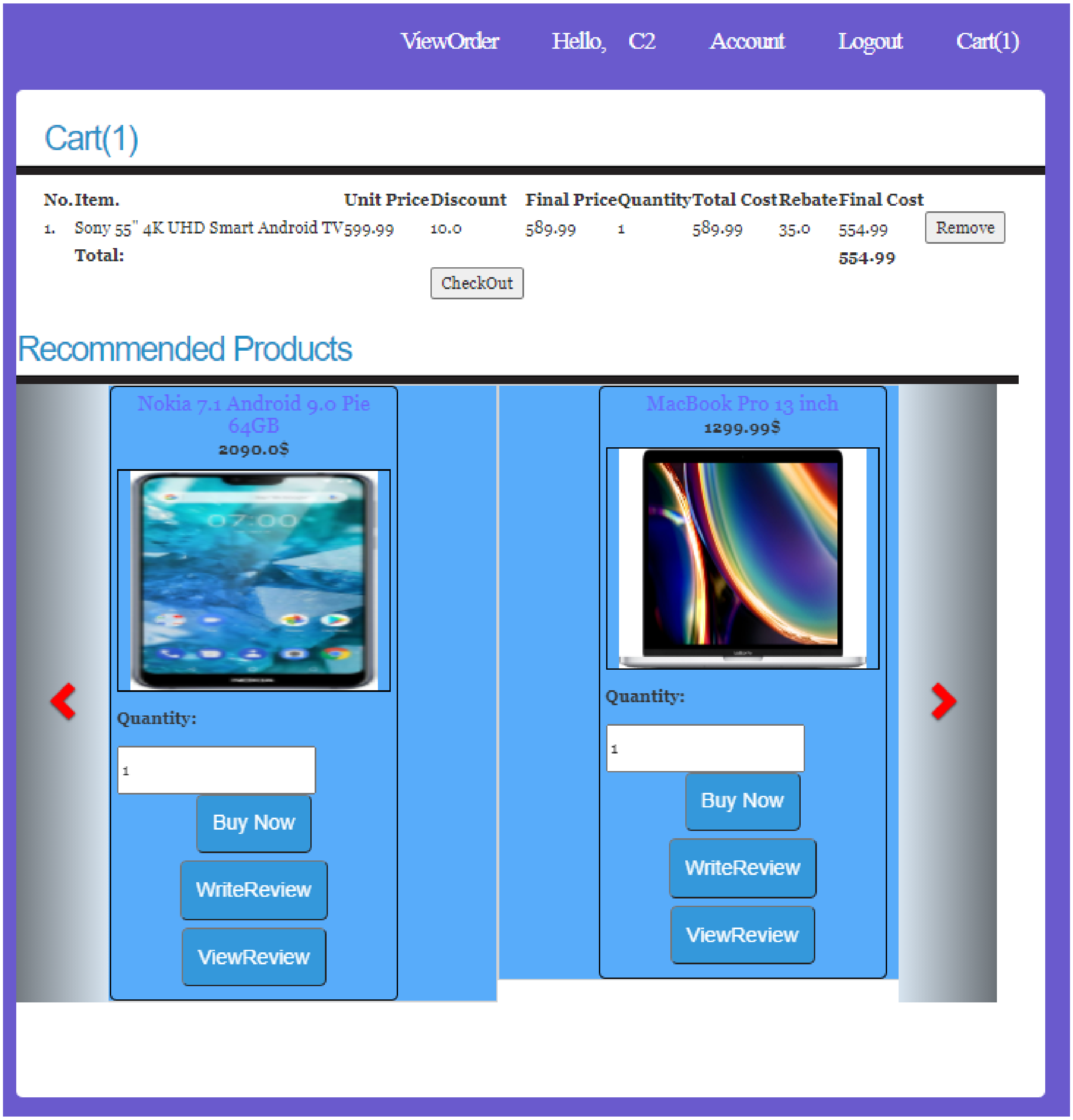
matrixFactorizationBasedRecommendations.csv



	A	B
1	c2	['hpSpectre', 'nokia71', 'macBookPro']
2	c3	['motoG', 'appleAirPods', 'macBookPro']
3	c1	['motoG', 'macBookPro', 'onePlusDash']
4	c4	['motoG', 'hpSpectre', 'LG55inch']
5	c5	['LG55inch', 'motoG', 'nokia71']

In application

For user c2



Cart(1)

No.	Item.	Unit Price	Discount	Final Price	Quantity	Total Cost	Rebate	Final Cost	
1.	Sony 55" 4K UHD Smart Android TV	599.99	10.0	589.99	1	589.99	35.0	554.99	Remove
	Total:							554.99	

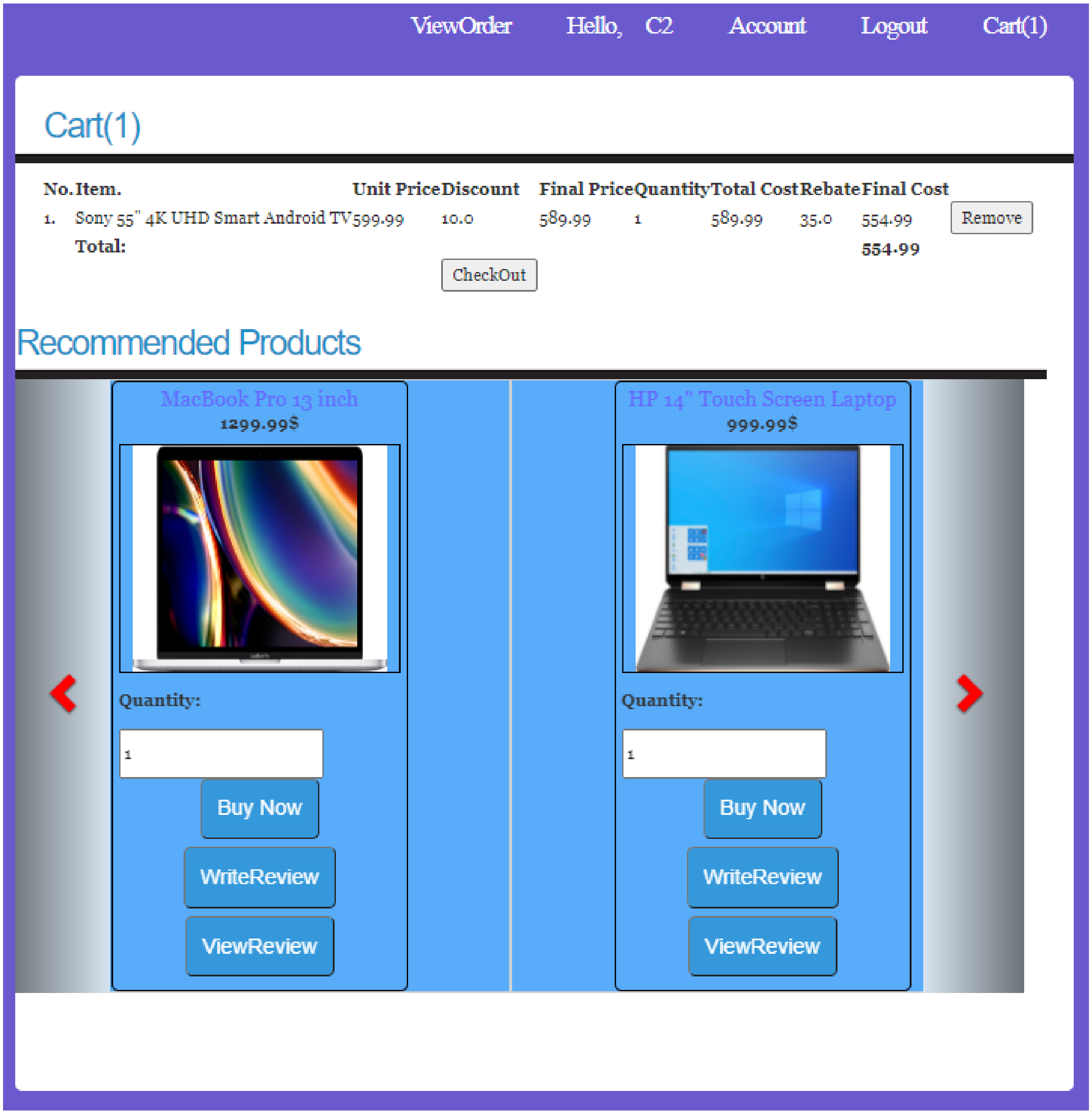
Recommended Products

Nokia 7.1 Android 9.0 Pie 64GB
2090.0$

Quantity: 1

MacBook Pro 13 inch
1299.99$

Quantity: 1



Cart(1)

No.	Item.	Unit Price	Discount	Final Price	Quantity	Total Cost	Rebate	Final Cost	
1.	Sony 55" 4K UHD Smart Android TV	599.99	10.0	589.99	1	589.99	35.0	554.99	Remove
	Total:							**554.99**	

Recommended Products

MacBook Pro 13 inch
1299.99$

Quantity: 1

HP 14" Touch Screen Laptop
999.99$

Quantity: 1