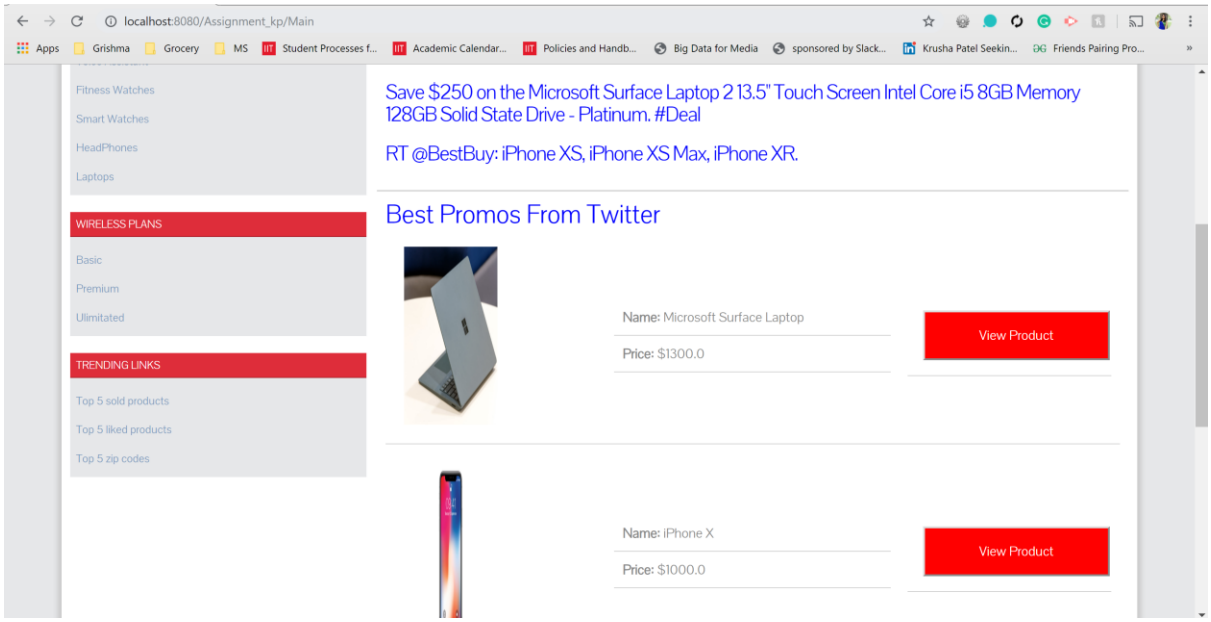
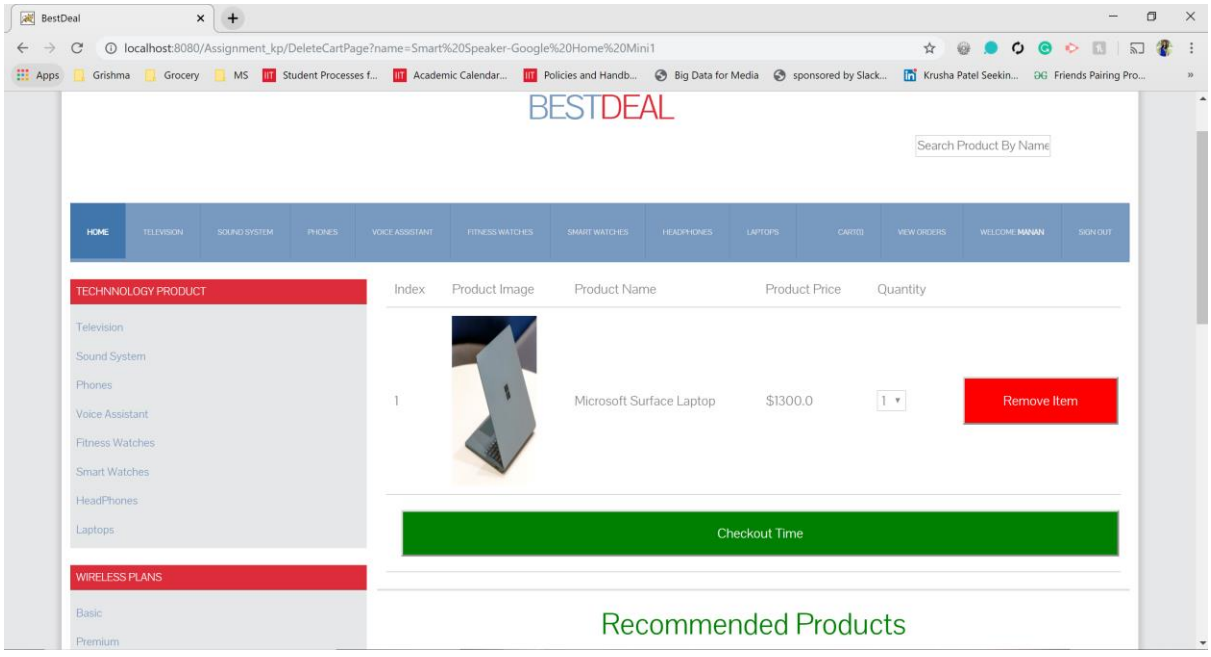
Assignment 5

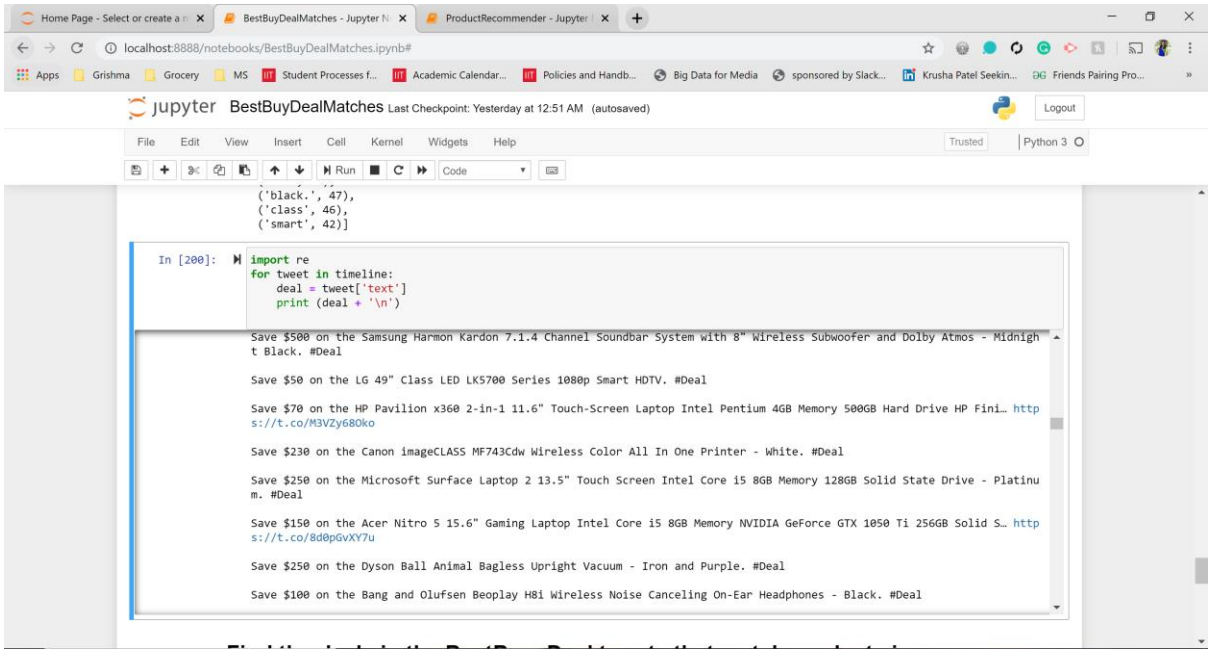
- **Displaying products from DealMatches.txt**



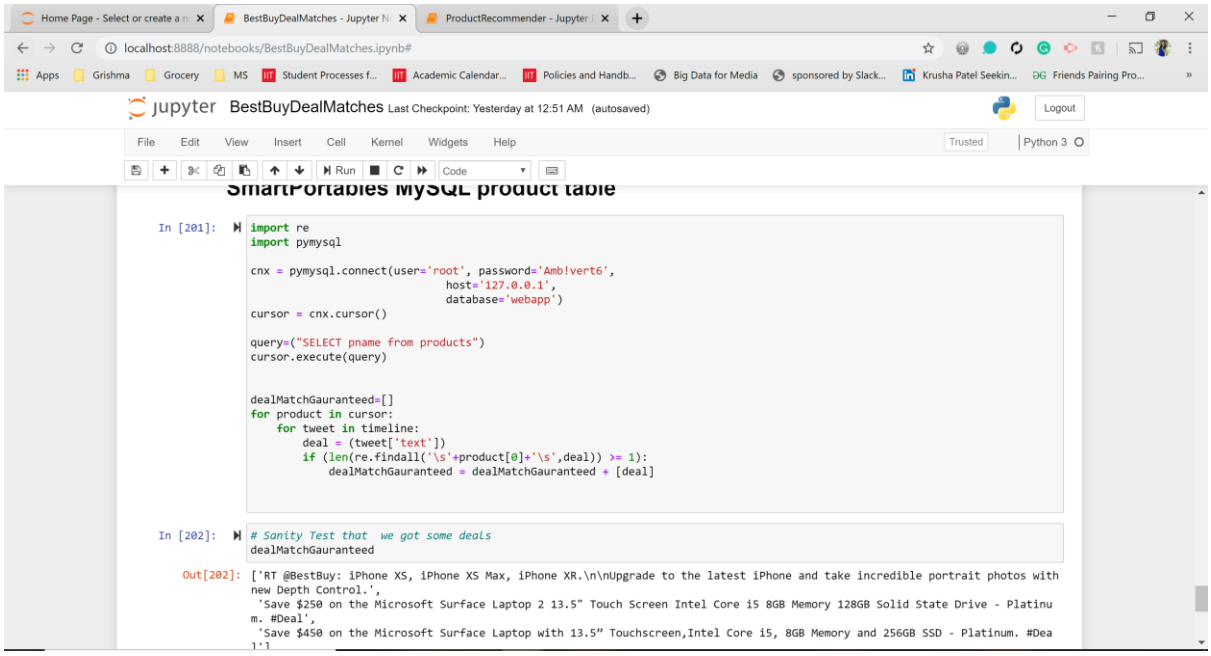
- **It can also be added to cart.**



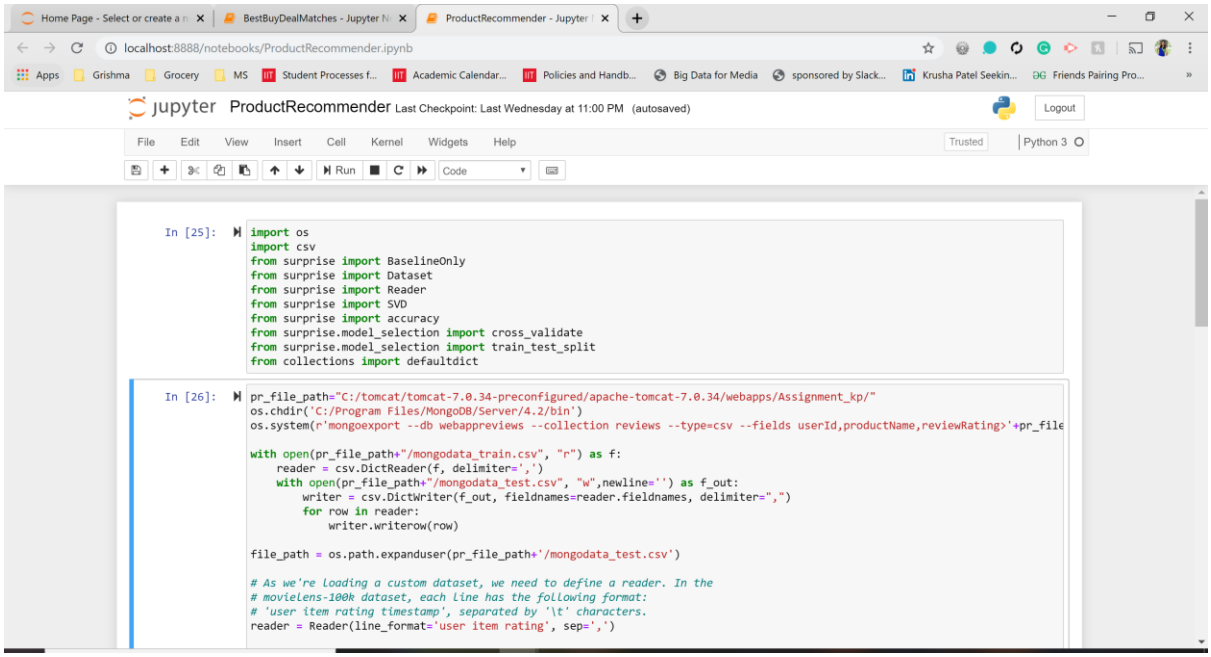
- **Get all the deals from tweeter using tweeter API.**



- **Match the products from tweeter to products table and save it to DealMatches.txt**

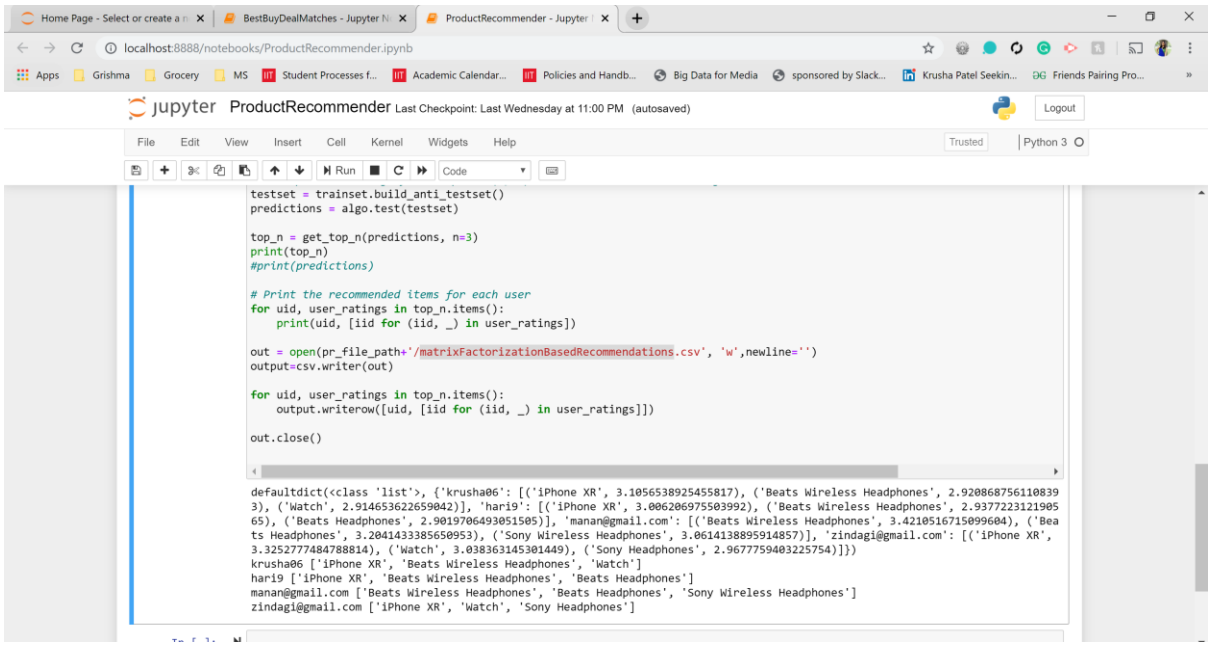


➢ ProductRecommender.ipnb



```
In [25]:
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

In [26]:
pr_file_path="C:/tomcat/tomcat-7.0.34-preconfigured/apache-tomcat-7.0.34/webapps/Assignment_kp/"
os.chdir('C:/Program Files/MongoDB/Server/4.2/bin')
os.system(r'mongoexport --db webappreviews --collection reviews --type=csv --fields userId,productName,reviewRating>'+pr_file

with open(pr_file_path+"/mongodata_train.csv", "r") as f:
    reader = csv.DictReader(f, delimiter=',')
    with open(pr_file_path+"/mongodata_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+'/mongodata_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=',')
```

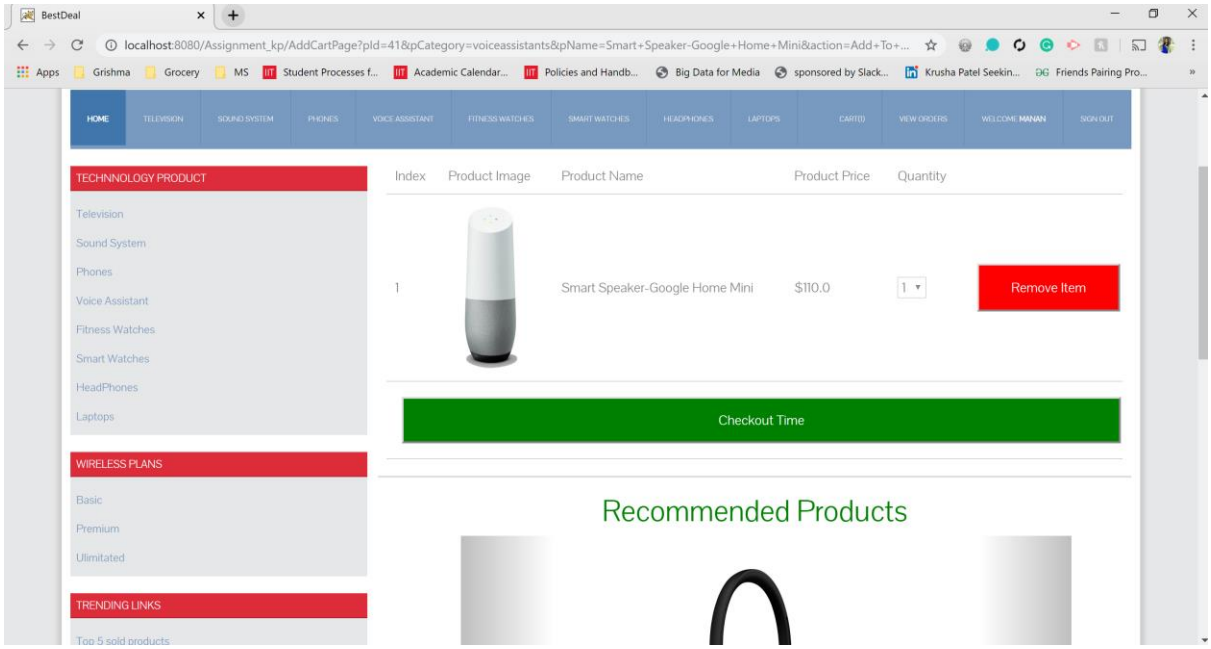
➢ Store the recommendation to matrixFactoriazatiobBasedRecommendation.csv

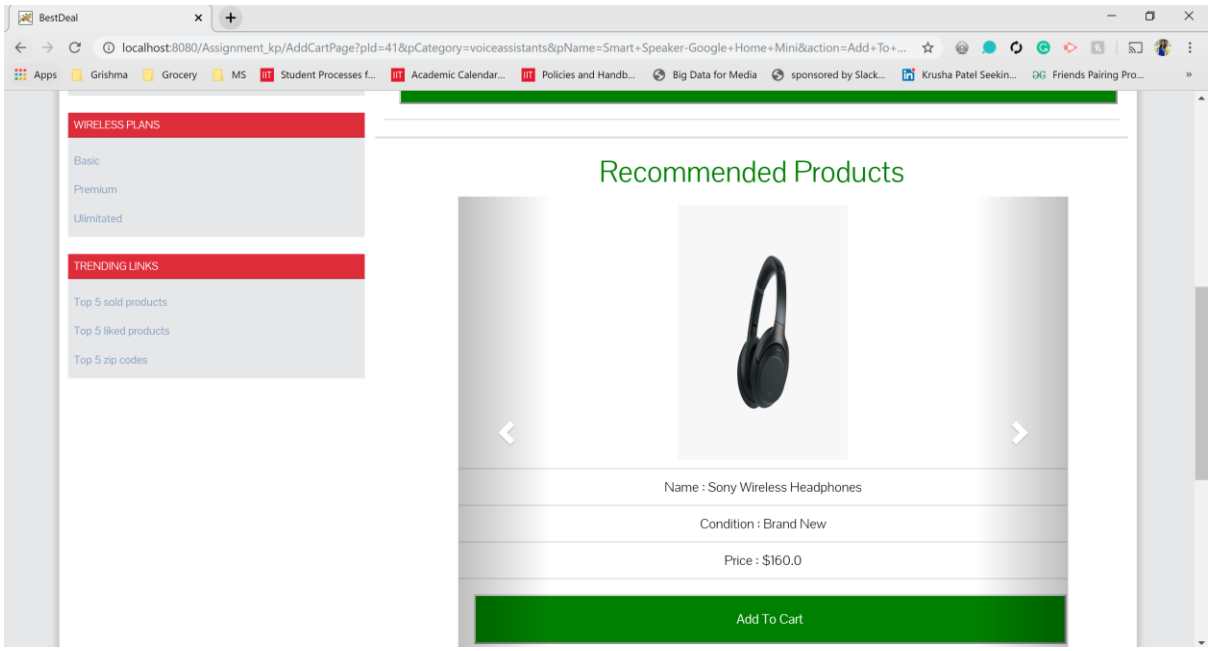


```
testset = trainset.build_anti_testset()
predictions = algo.test(testset)

top_n = get_top_n(predictions, n=3)
print(top_n)
#print(predictions)

# 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()
```

```
defaultdict(<class 'list'>, {'krusha06': [('iPhone XR', 3.1056538925455817), ('Beats Wireless Headphones', 2.920868756110839
3), ('Watch', 2.914653622659042)], 'hari9': [('iPhone XR', 3.006206975503992), ('Beats Wireless Headphones', 2.9377223121905
65), ('Beats Headphones', 2.9019706493051505)], 'manan@gmail.com': [('Beats Wireless Headphones', 3.4210516715099604), ('Bea
ts Headphones', 3.2041433385650953), ('Sony Wireless Headphones', 3.0614138895914857)], 'zindagi@gmail.com': [('iPhone XR',
3.3252777484788814), ('Watch', 3.038363145301449), ('Sony Headphones', 2.9677759403225754)]})
krusha06 ['iPhone XR', 'Beats Wireless Headphones', 'Watch']
hari9 ['iPhone XR', 'Beats Wireless Headphones', 'Beats Headphones']
manan@gmail.com ['Beats Wireless Headphones', 'Beats Headphones', 'Sony Wireless Headphones']
zindagi@gmail.com ['iPhone XR', 'Watch', 'Sony Headphones']
```

- **Showing recommendation for one User from matrixFactoriazatiobBasedRecommendation.csv in carosel.**





- **Showing recommendation for another User from matrixFactoriazatiobBasedRecommendation.csv in carosel.**