

Enterprise Web Applications Assignment 5

Akash Kumar Shetty(A20447720)

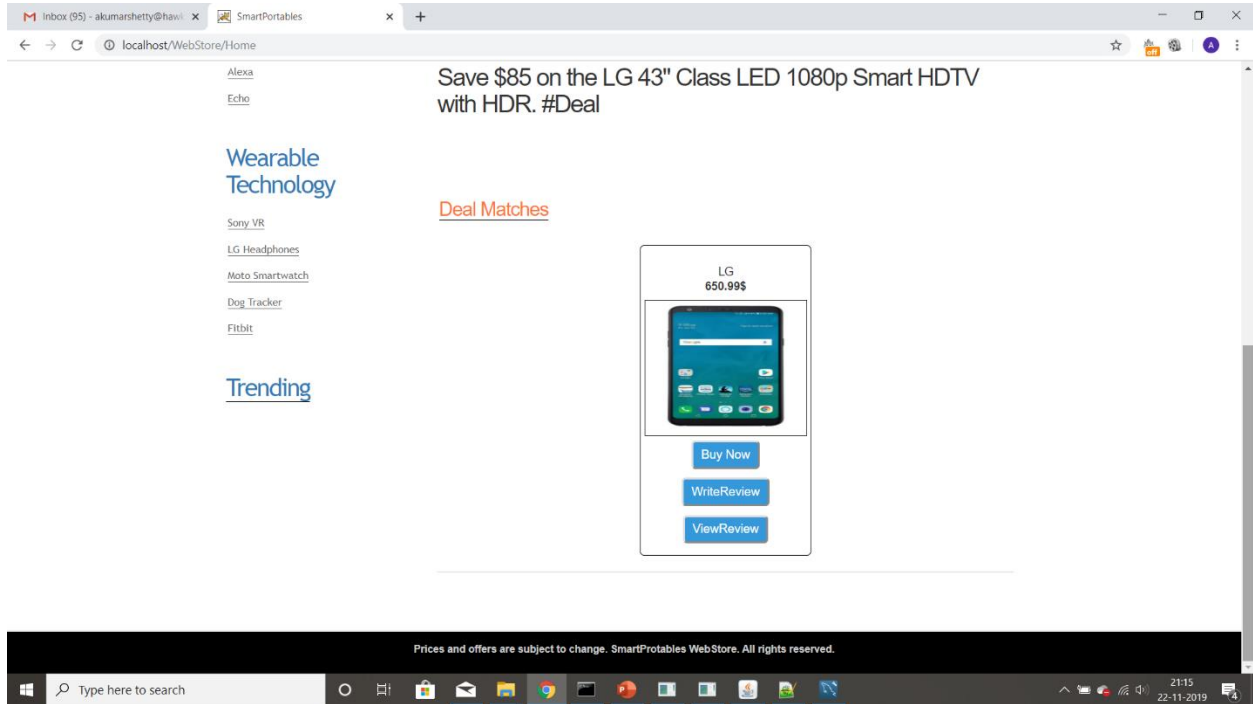


Figure 1: Home page showing deal matches

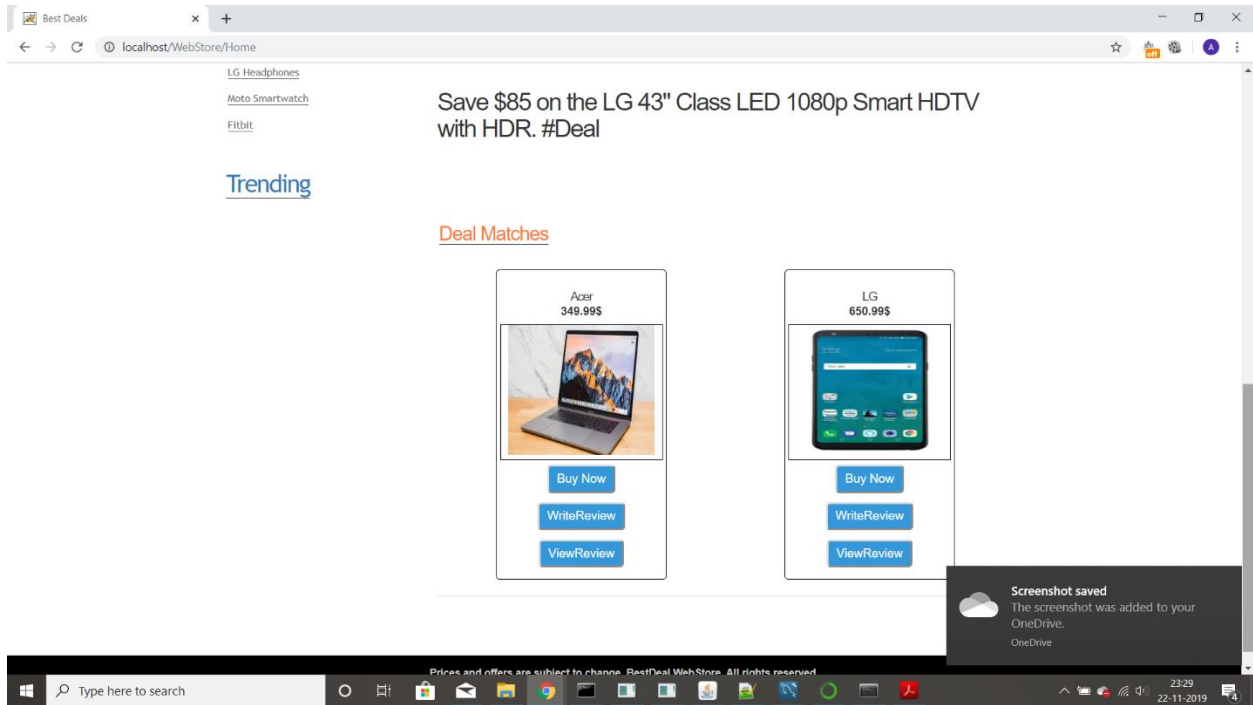
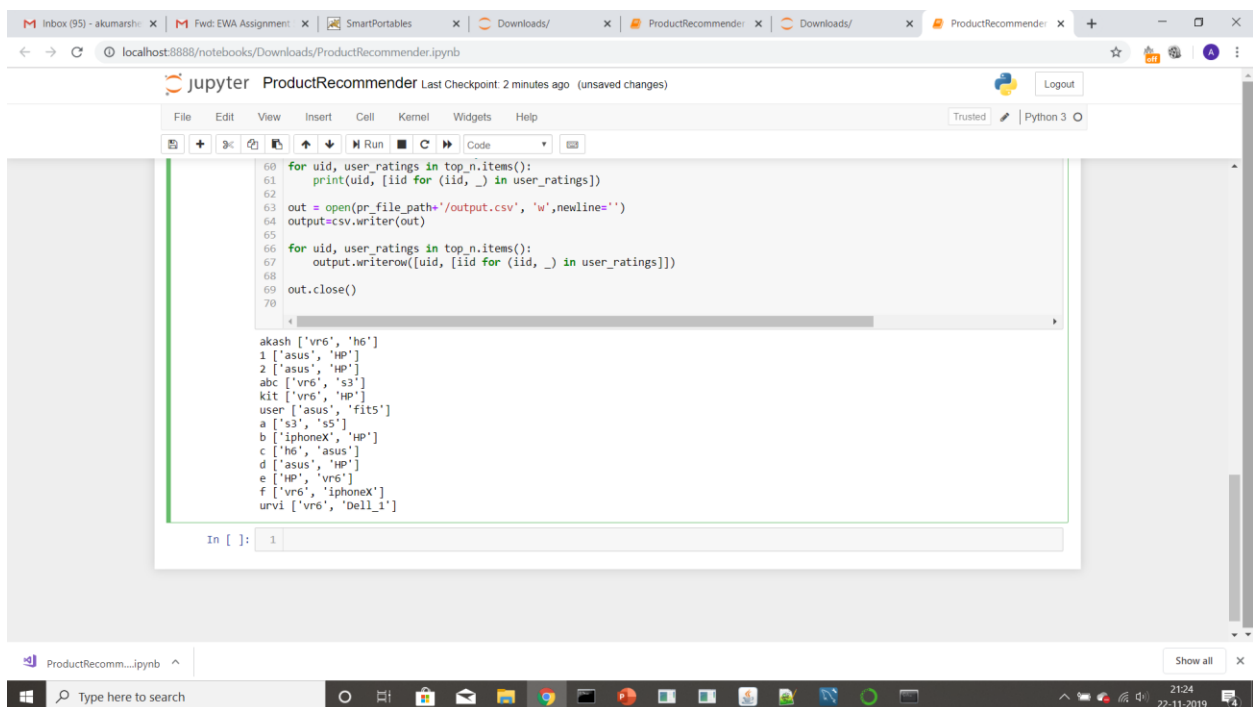
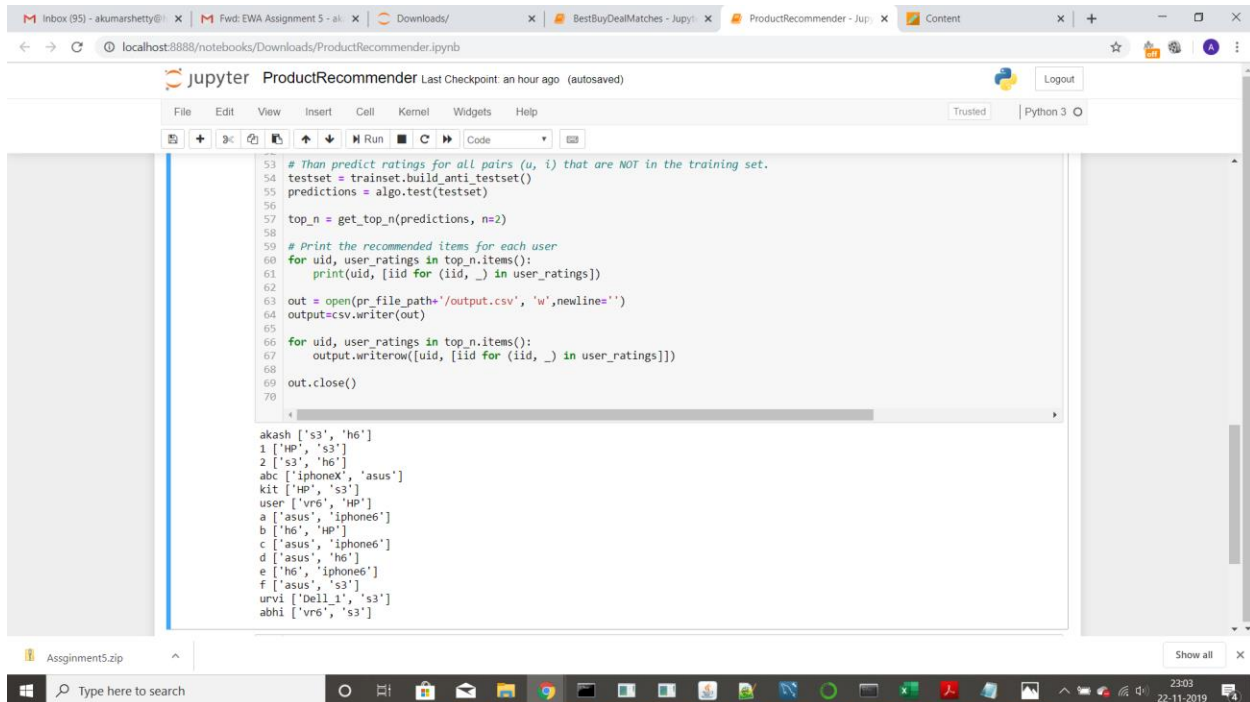


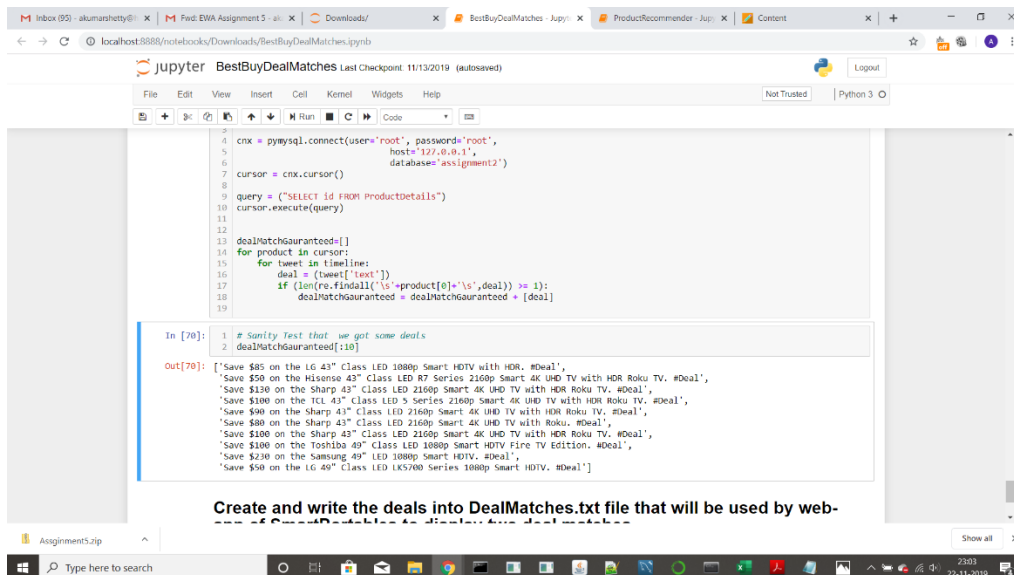
Figure 2: Two items shown in Deal Matches





```
53 # Then predict ratings for all pairs (u, i) that are NOT in the training set.
54 testset = trainset.build_anti_testset()
55 predictions = algo.test(testset)
56
57 top_n = get_top_n(predictions, n=2)
58
59 # Print the recommended items for each user
60 for uid, user_ratings in top_n.items():
61     print(uid, [iid for (iid, _) in user_ratings])
62
63 out = open(pr_file_path+'output.csv', 'w', newline='')
64 output = csv.writer(out)
65
66 for uid, user_ratings in top_n.items():
67     output.writerow([uid, [iid for (iid, _) in user_ratings]])
68
69 out.close()
70
71 akash ['s3', 'h6']
72 1 ['HP', 's3']
73 2 ['s3', 'h6']
74 abc ['iphoneX', 'asus']
75 kit ['HP', 's3']
76 user ['vr6', 'HP']
77 a ['asus', 'iphone6']
78 b ['h6', 'HP']
79 c ['asus', 'iphone6']
80 d ['asus', 'h6']
81 e ['h6', 'iphone6']
82 f ['asus', 's3']
83 urvi ['dell_i', 's3']
84 abhi ['vr6', 's3']
```

Figure 3: Recommendation item list after running the python script



```
3 cnx = pymysql.connect(user='root', password='root',
4                       host='127.0.0.1',
5                       database='assignment2')
6 cursor = cnx.cursor()
7
8 query = ("SELECT id FROM ProductDetails")
9 cursor.execute(query)
10
11
12 dealMatchGauranteed[]
13 for product in cursor:
14     for tweet in timeline:
15         deal = (tweet['text'])
16         if (len(re.findall('\$'+product[0]+'\\s', deal)) >= 1):
17             dealMatchGauranteed = dealMatchGauranteed + [deal]
18
19
20 In [70]: 1 # Sanity Test that we got some deals
21          2 dealMatchGauranteed[:10]
22
23 Out[70]: ['Save $85 on the LG 43" Class LED 1080p Smart HDTV with HDR. #Deal',
24           'Save $50 on the Hisense 43" Class LED R7 Series 2160p Smart 4K UHD TV with HDR Roku TV. #Deal',
25           'Save $130 on the Sharp 43" Class LED 2160p Smart 4K UHD TV with HDR Roku TV. #Deal',
26           'Save $100 on the TCL 43" Class LED S-Series 2160p Smart 4K UHD TV with HDR Roku TV. #Deal',
27           'Save $80 on the Sharp 43" Class LED 2160p Smart 4K UHD TV with HDR Roku TV. #Deal',
28           'Save $80 on the Sharp 43" Class LED 2160p Smart 4K UHD TV with Roku. #Deal',
29           'Save $100 on the Sharp 43" Class LED 2160p Smart 4K UHD TV with HDR Roku TV. #Deal',
30           'Save $100 on the Toshiba 49" Class LED 1080p Smart HDTV Fire TV Edition. #Deal',
31           'Save $230 on the Samsung 49" LED 1080p Smart HDTV. #Deal',
32           'Save $50 on the LG 49" Class LED LK5700 Series 1080p Smart HDTV. #Deal']
```

Create and write the deals into DealMatches.txt file that will be used by web-

Figure 4: The best deals list after running the python script

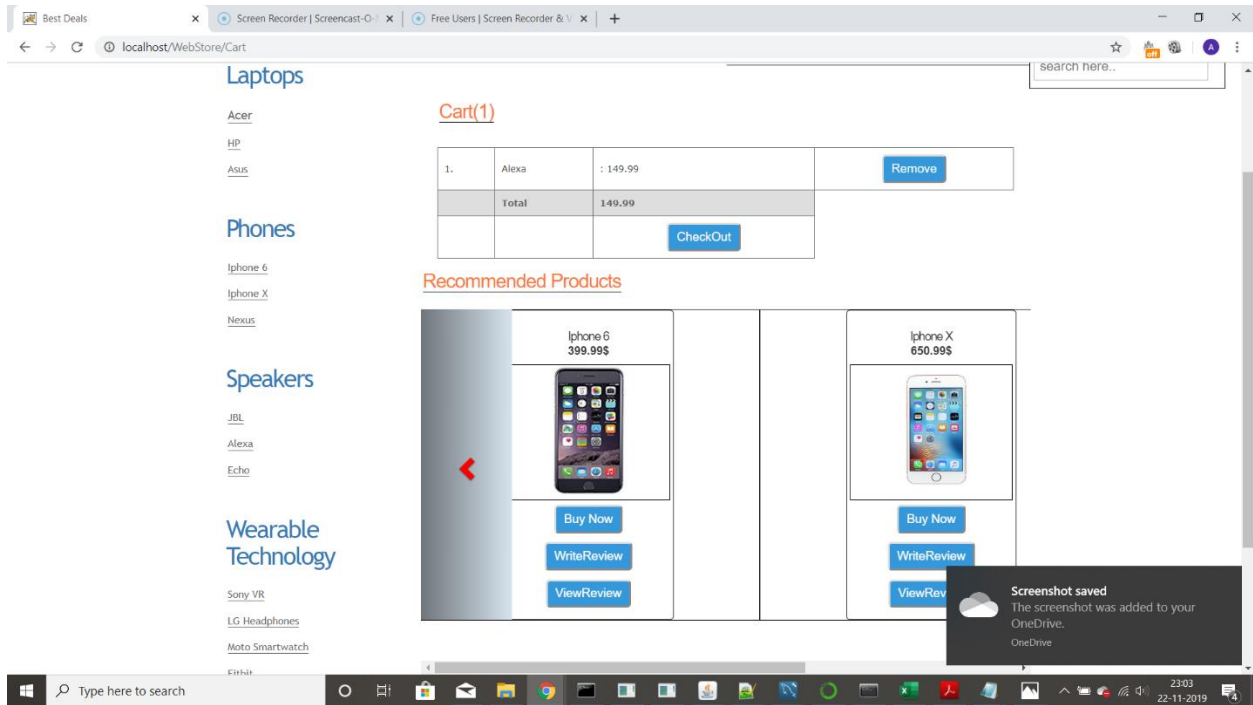


Figure 5: The recommended items shown when the user is checking out(3 items are shown)

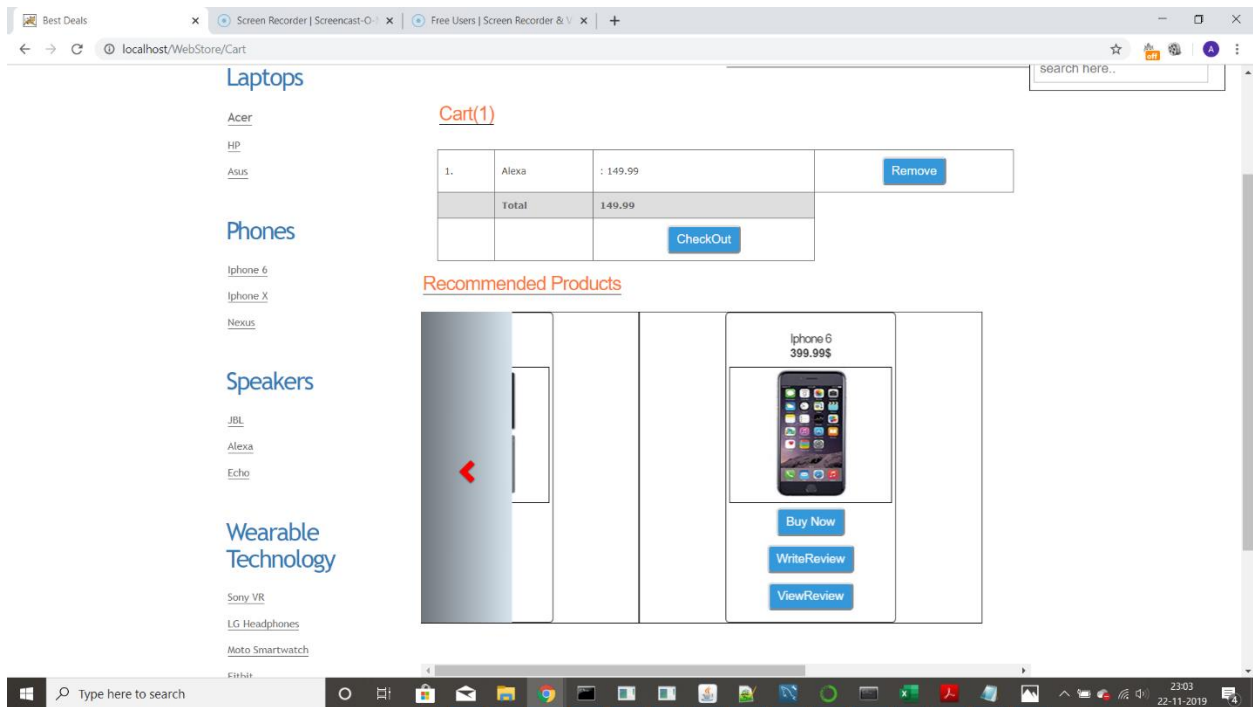


Figure 6: The other recommended products

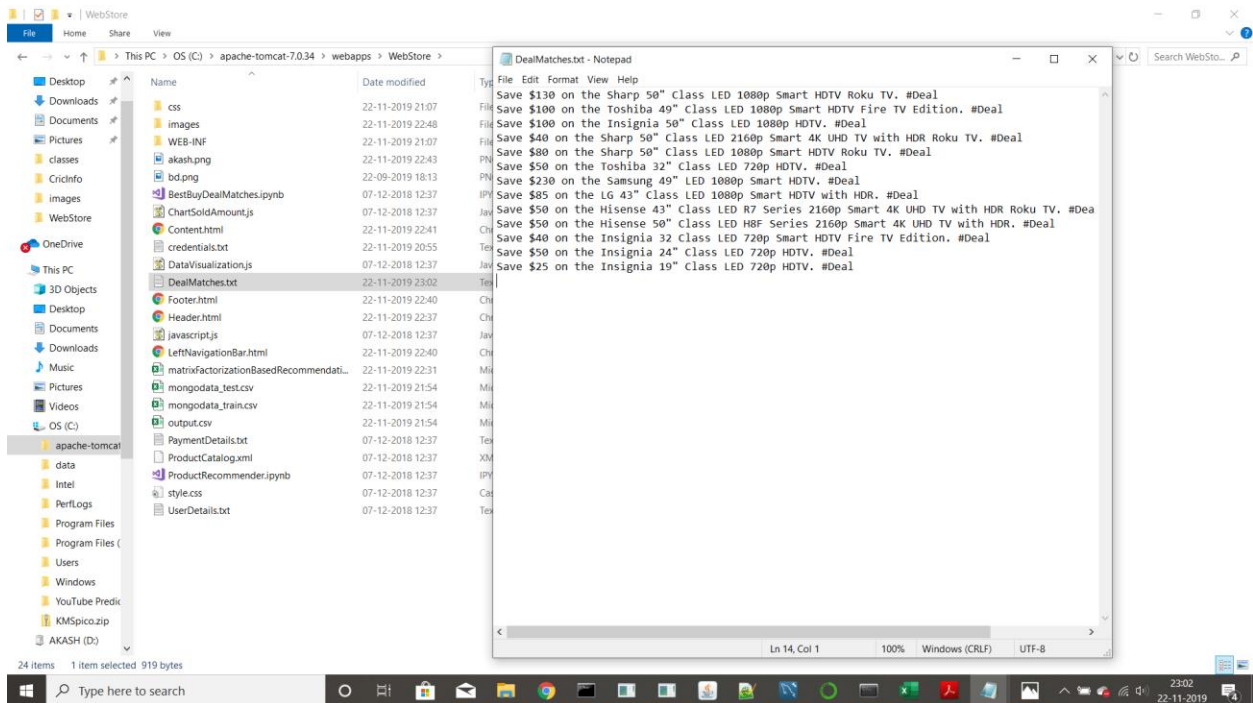


Figure 7: The deal matches list stored in the dealmatches.txt after running the python script

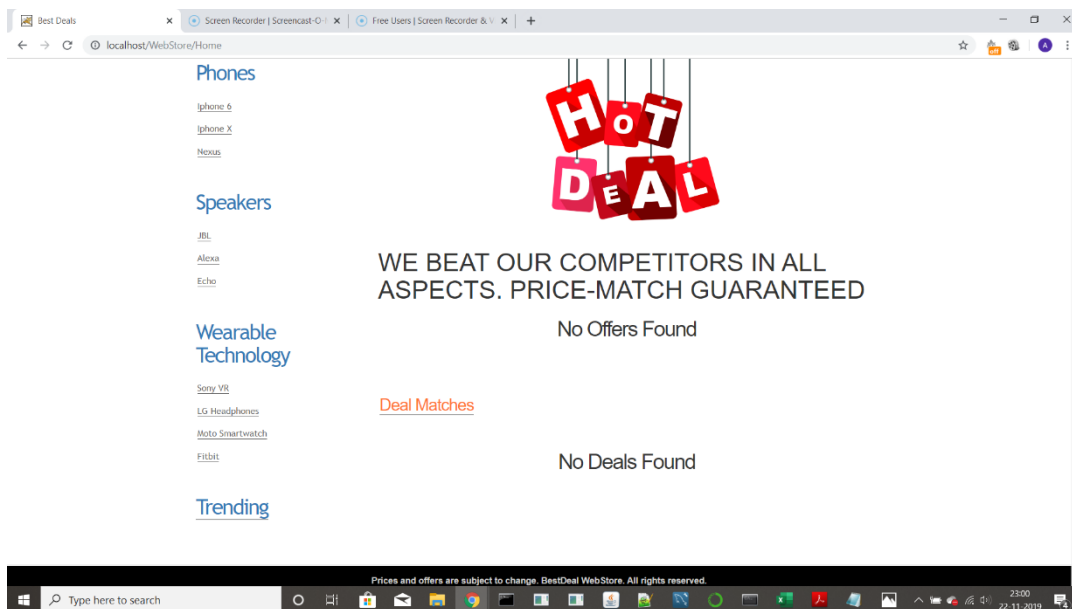


Figure 8: No deals found when the dealmatches.txt is empty

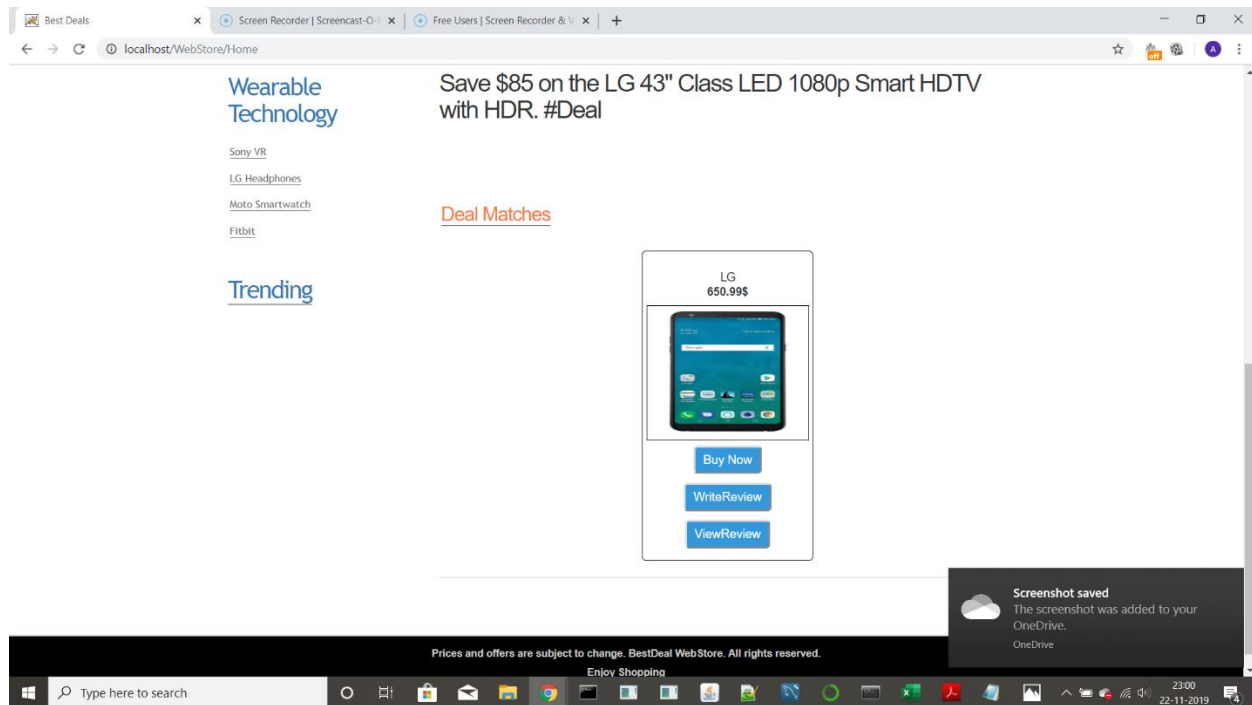


Figure 9: Reshowing the deal matches after the dealmatches.txt file is filled with content

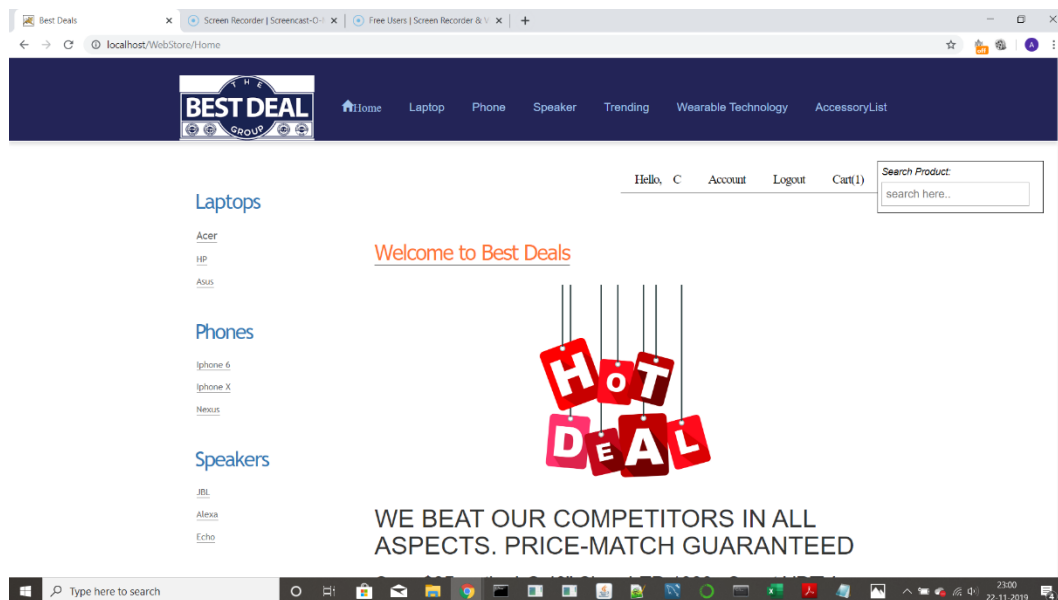


Figure 10: The home pace with new features of hot deals

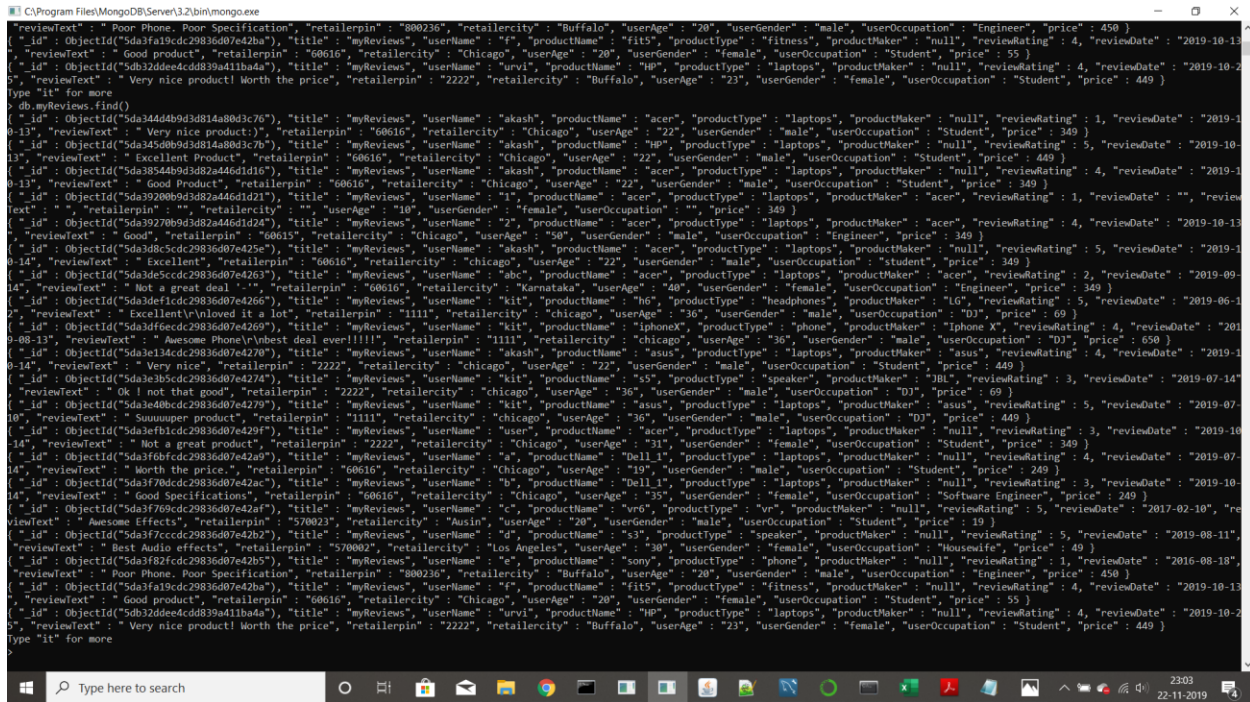


Figure 10: MongoDB showing the reviews of customers

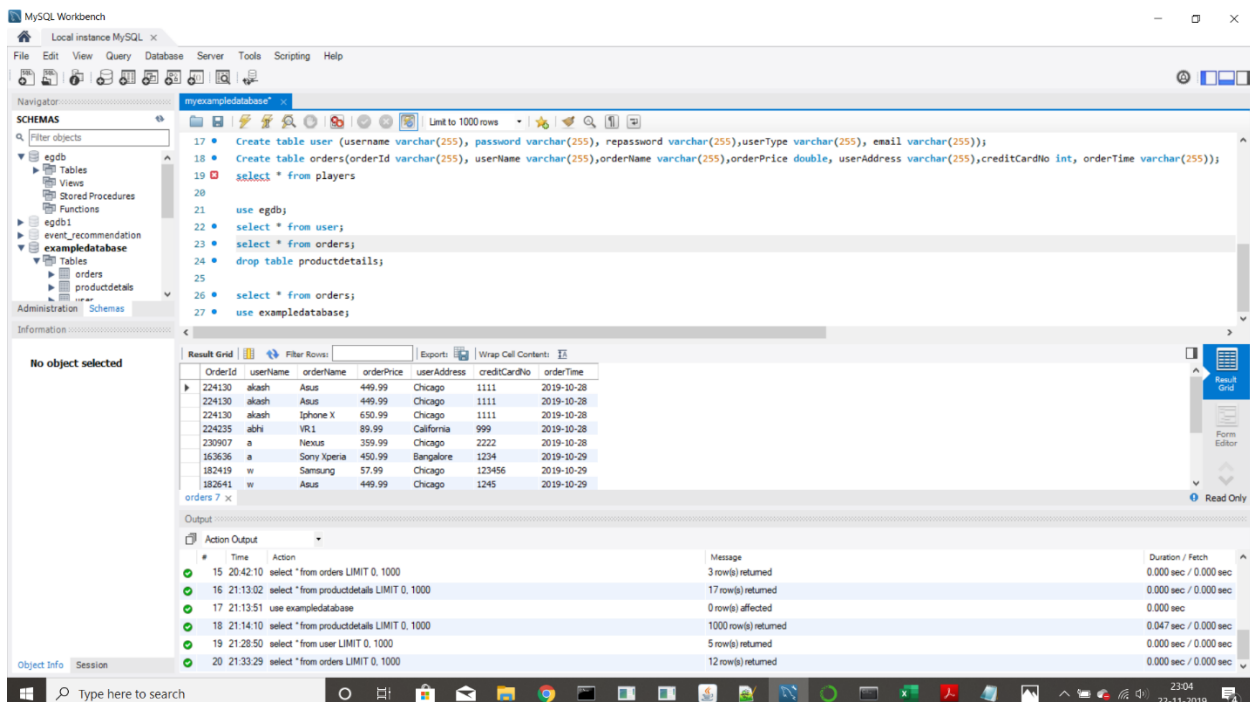


Figure 11: The MySQL showing the orders table

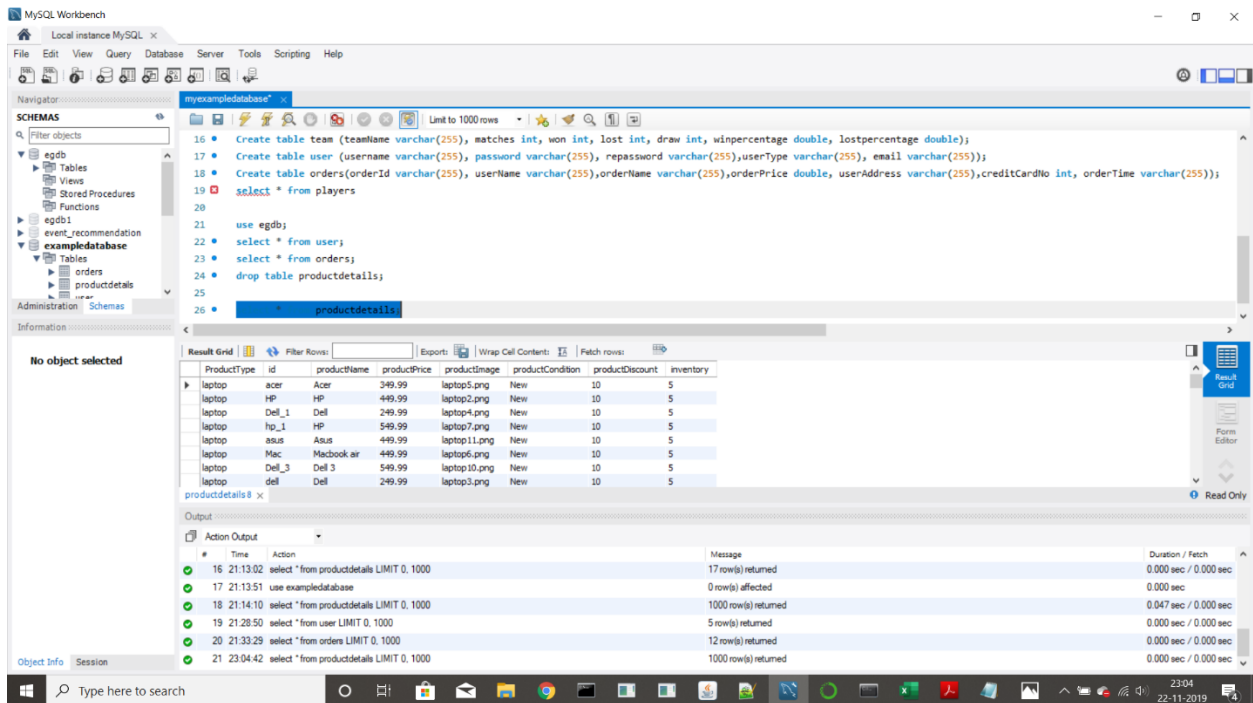


Figure 12: MySQL showing the product details

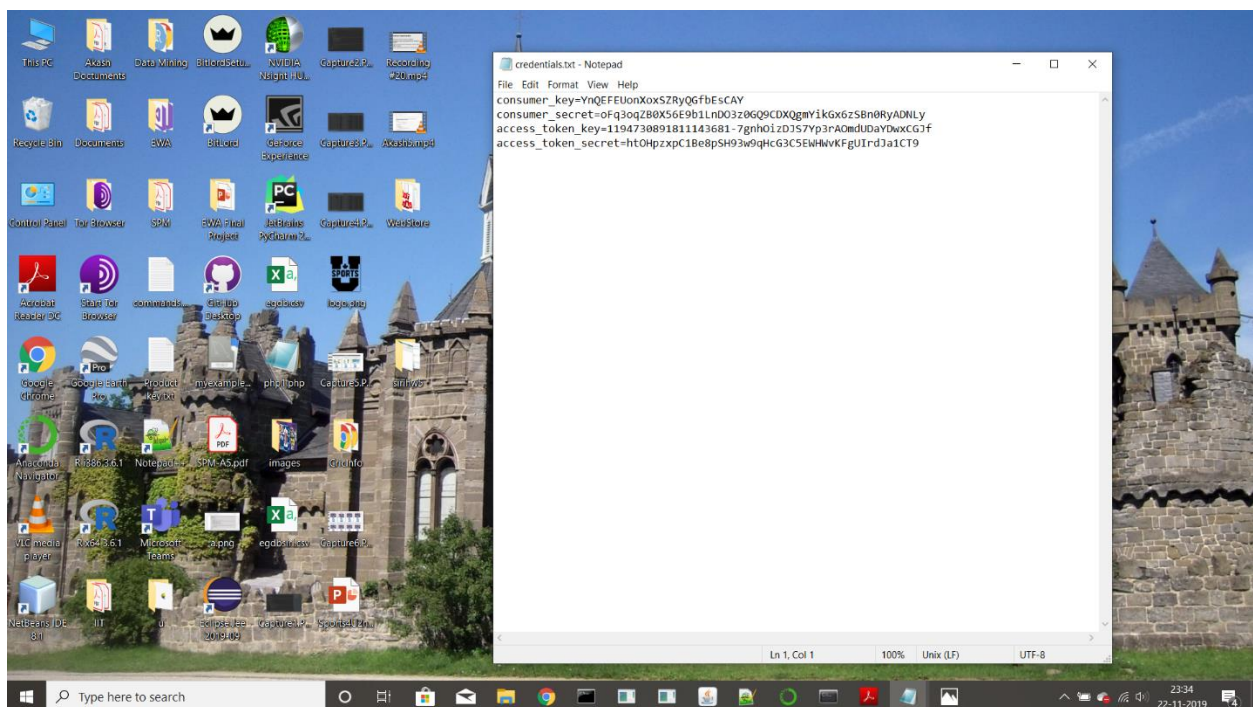


Figure 13: The credentials used to get the Twitter data