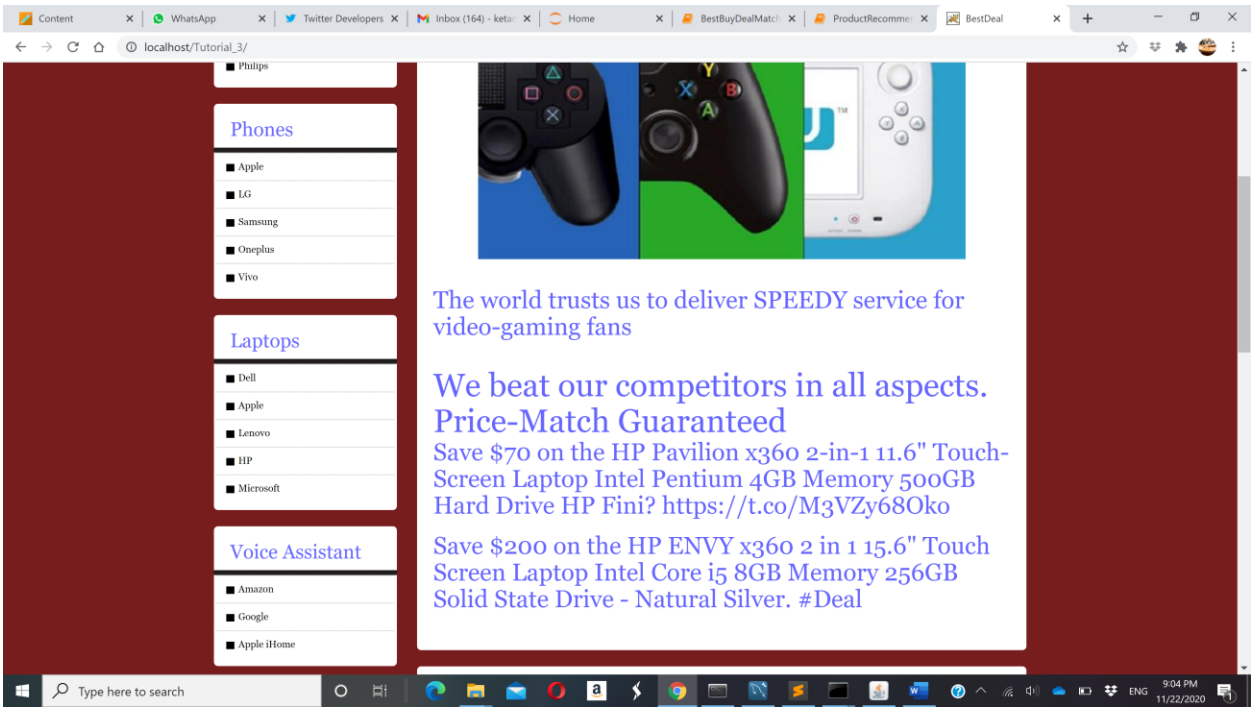
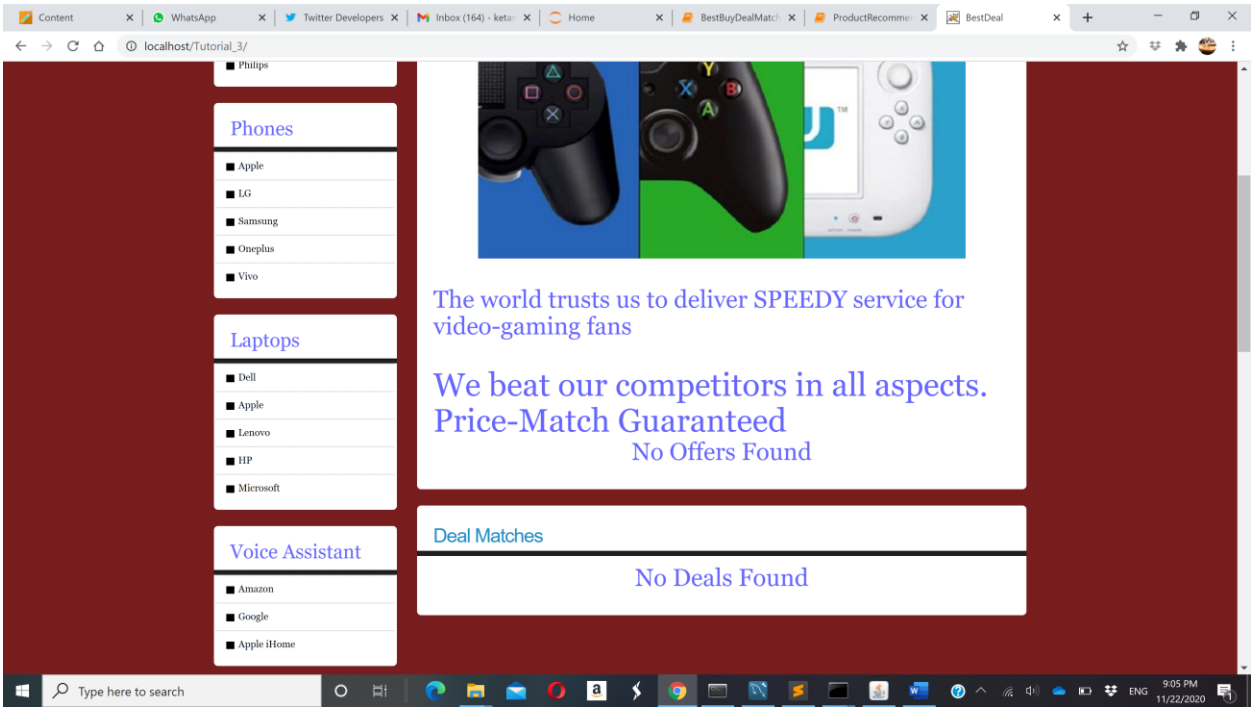
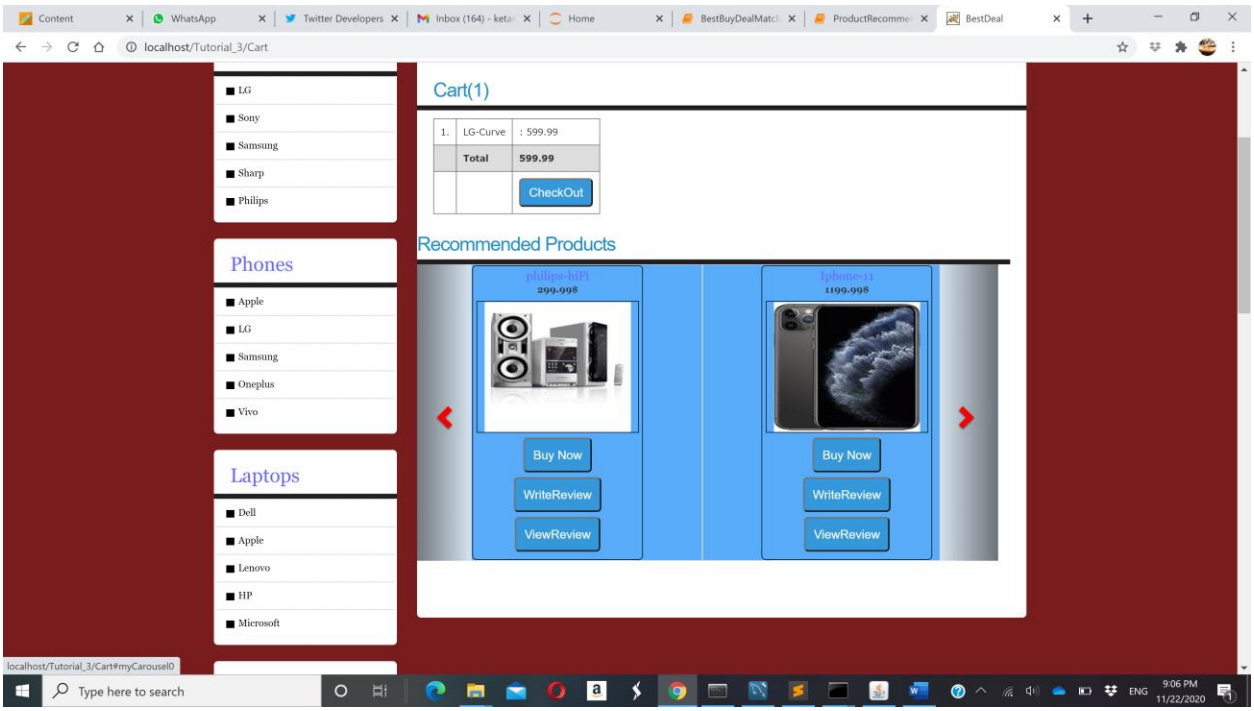
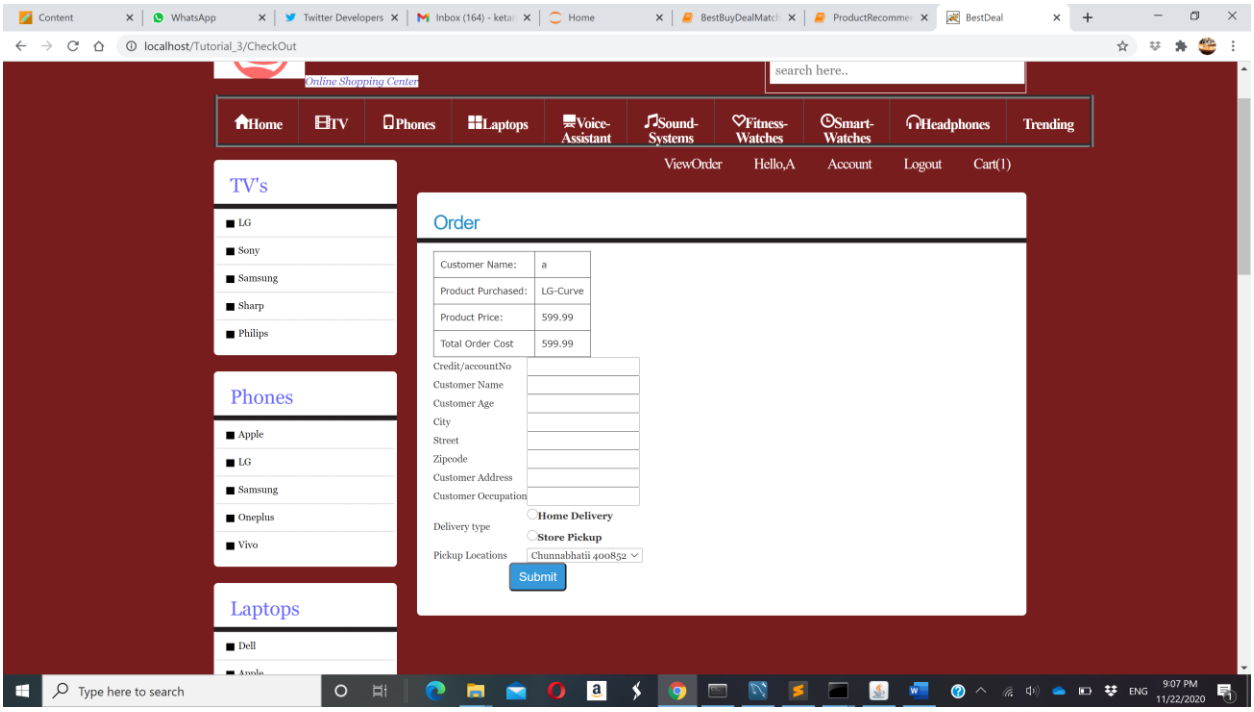
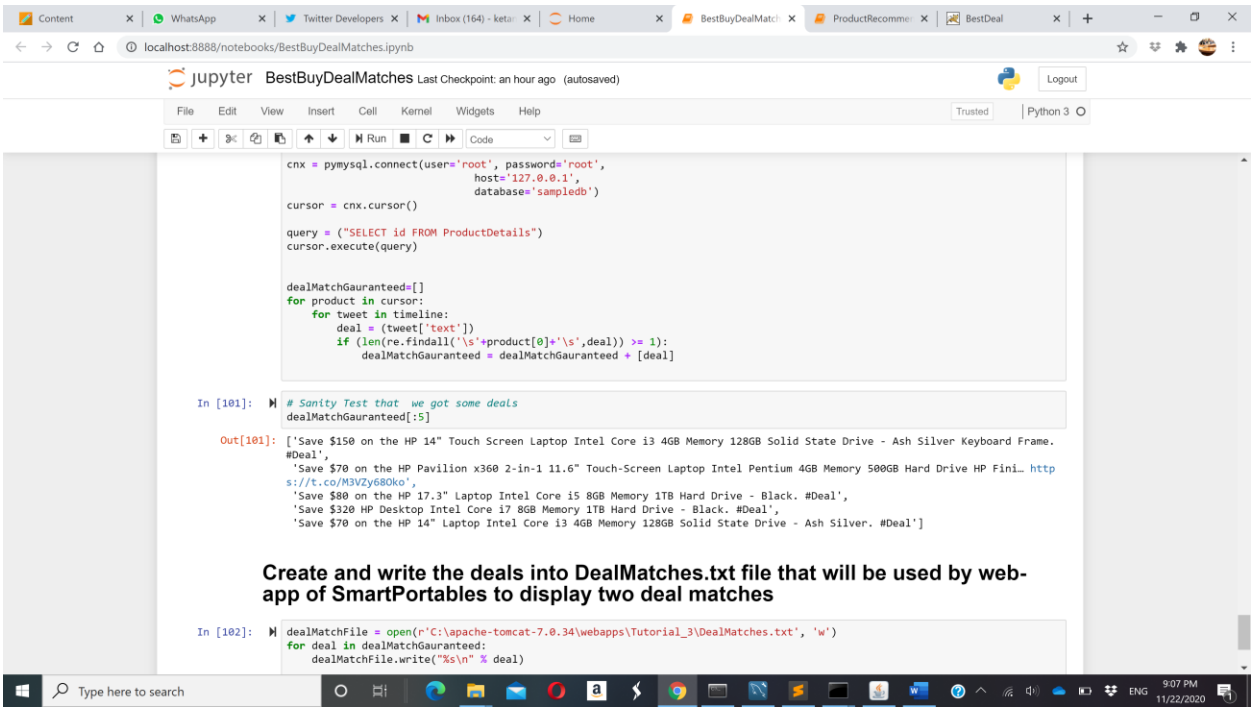
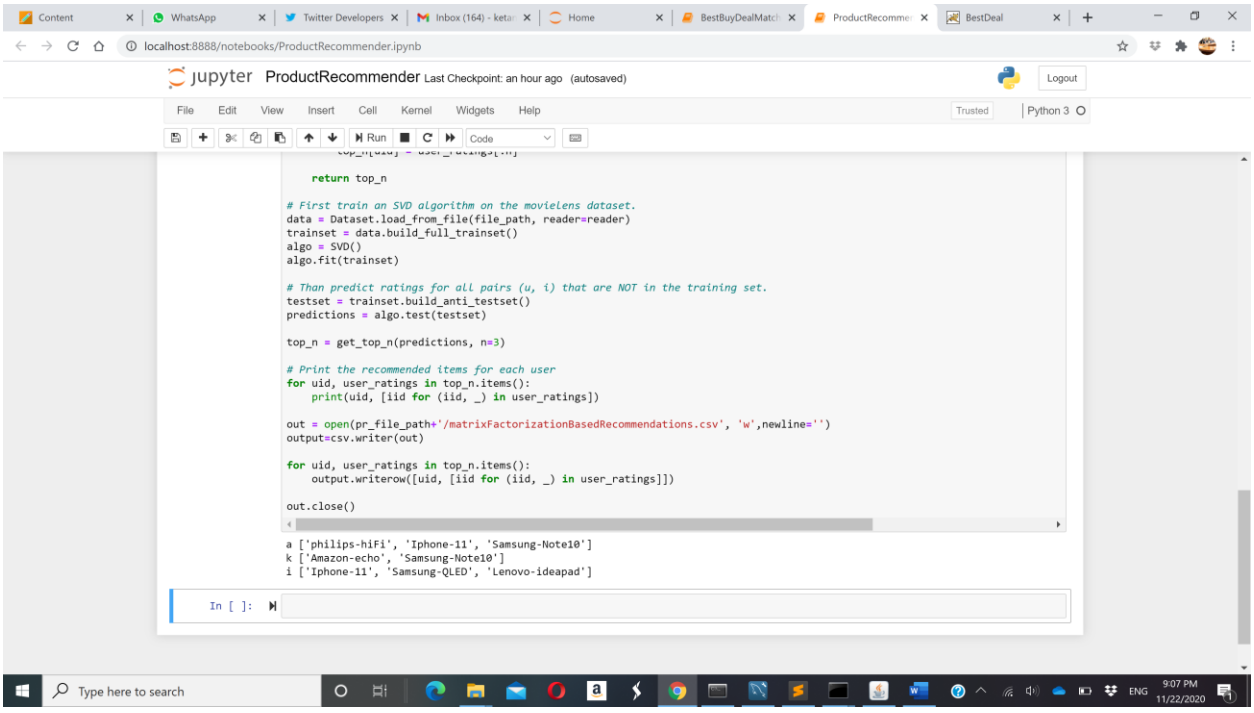

Content
WhatsApp
Twitter Developers
Inbox (164) - ketan
Home
BestBuyDealMatch
ProductRecommender
BestDeal
localhost/Tutorial_3/
Philips
Phones
Apple
LG
Samsung
Oneplus
Vivo
Laptops
Dell
Apple
Lenovo
HP
Microsoft
Voice Assistant
Amazon
Google
Apple iHome
The world trusts us to deliver SPEEDY service for video-gaming fans
We beat our competitors in all aspects.
Price-Match Guaranteed
Save $70 on the HP Pavilion x360 2-in-1 11.6" Touch-Screen Laptop Intel Pentium 4GB Memory 500GB Hard Drive HP Fini? https://t.co/M3VZy68Oko
Save $200 on the HP ENVY x360 2 in 1 15.6" Touch Screen Laptop Intel Core i5 8GB Memory 256GB Solid State Drive - Natural Silver. #Deal
Type here to search
ENG
9:04 PM
11/22/2020

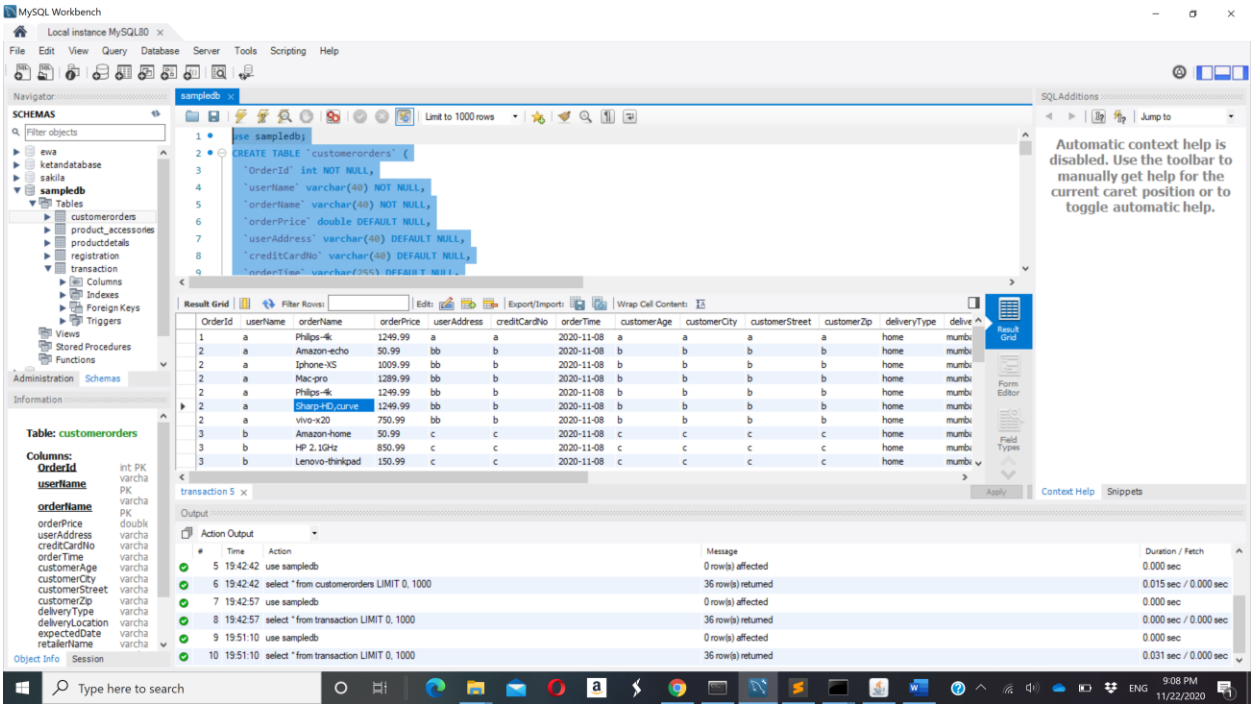

Content
WhatsApp
Twitter Developers
Inbox (164) - ketan
Home
BestBuyDealMatch
ProductRecommender
BestDeal
localhost/Tutorial_3/
Philips
Phones
Apple
LG
Samsung
Oneplus
Vivo
Laptops
Dell
Apple
Lenovo
HP
Microsoft
Voice Assistant
Amazon
Google
Apple iHome
The world trusts us to deliver SPEEDY service for video-gaming fans
We beat our competitors in all aspects.
Price-Match Guaranteed
No Offers Found
Deal Matches
No Deals Found
Type here to search
ENG
9:05 PM
11/22/2020

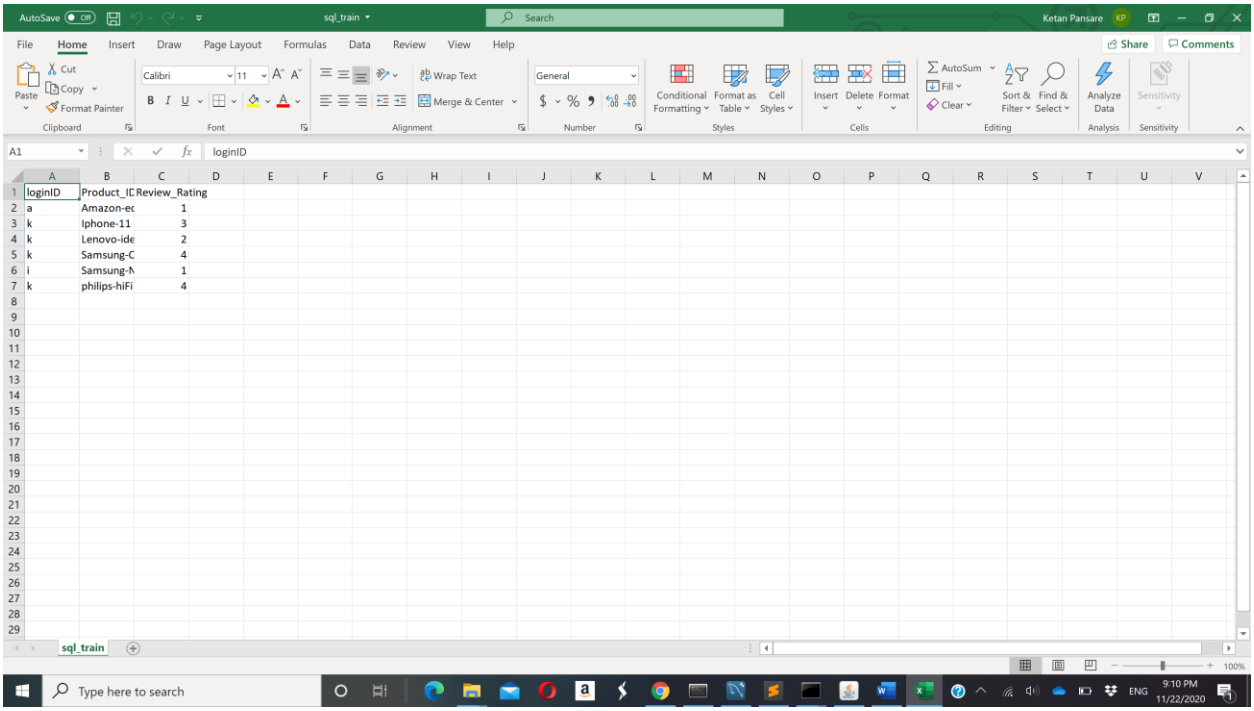

localhost/Tutorial_3/Cart
LG
Sony
Samsung
Sharp
Philips
Phones
Apple
LG
Samsung
Oneplus
Vivo
Laptops
Dell
Apple
Lenovo
HP
Microsoft
Cart(1)
1. LG-Curve : 599.99
Total 599.99
CheckOut
Recommended Products
philips-hifi
299.99$
Buy Now
WriteReview
ViewReview
Iphone-11
1199.99$
Buy Now
WriteReview
ViewReview
localhost/Tutorial_3/Cart#myCarousel0
Type here to search
ENG
9:06 PM
11/22/2020

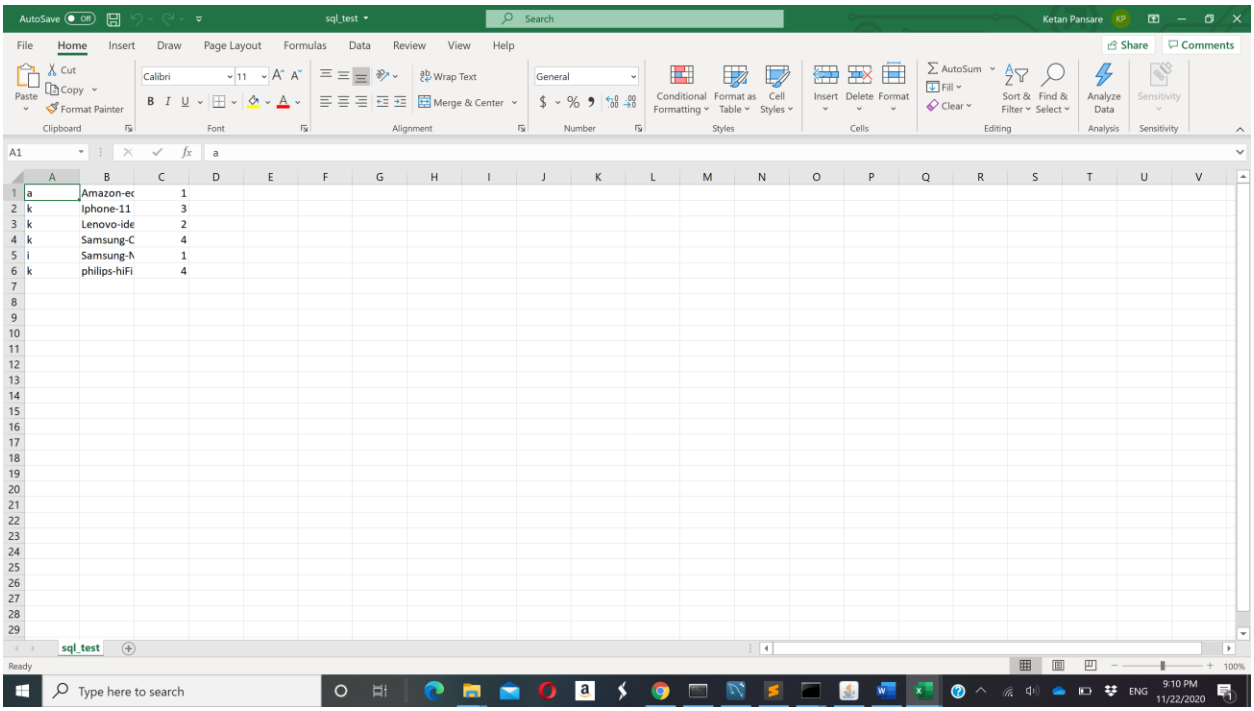

localhost/Tutorial_3/CheckOut
Online Shopping Center
search here..
Home
TV
Phones
Laptops
Voice-Assistant
Sound-Systems
Fitness-Watches
Smart-Watches
Headphones
Trending
ViewOrder
Hello,A
Account
Logout
Cart(1)
TV's
LG
Sony
Samsung
Sharp
Philips
Phones
Apple
LG
Samsung
Oneplus
Vivo
Laptops
Dell
Order
Customer Name: a
Product Purchased: LG-Curve
Product Price: 599.99
Total Order Cost 599.99
Credit/accountNo
Customer Name
Customer Age
City
Street
Zipcode
Customer Address
Customer Occupation
Delivery type
Home Delivery
Store Pickup
Pickup Locations
Chunnabhatii 400852
Submit
Type here to search
ENG
9:07 PM
11/22/2020

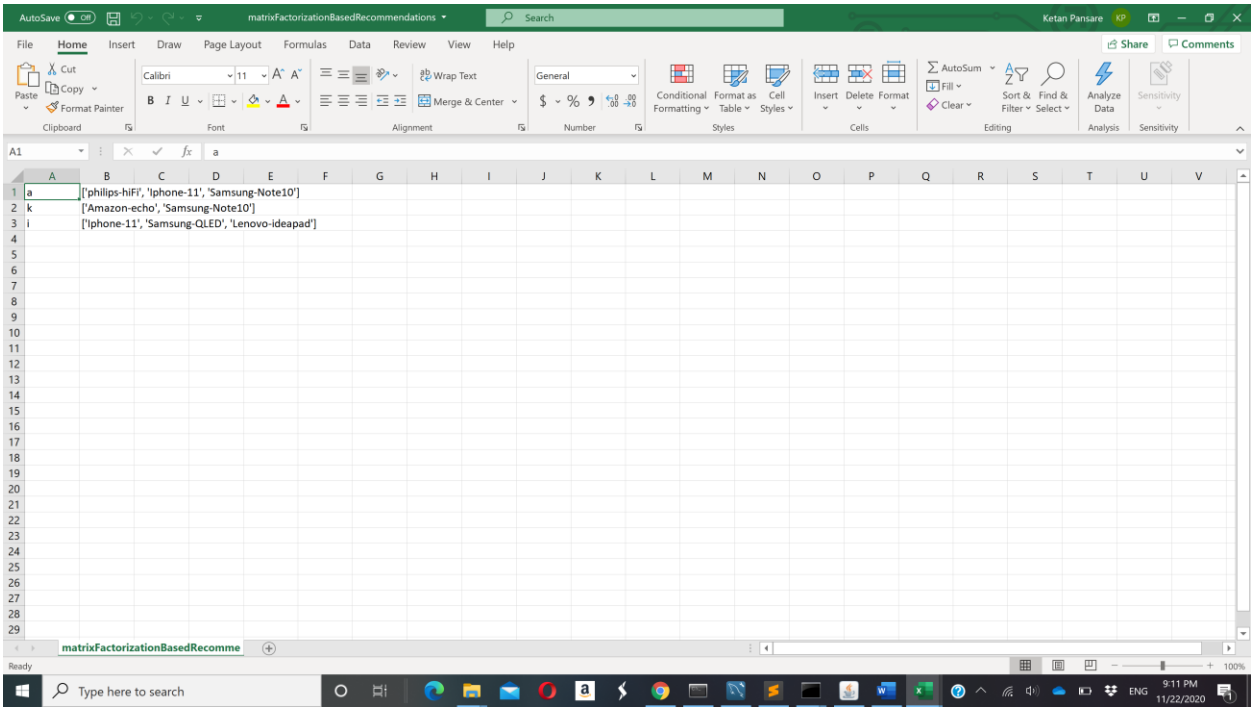

BestBuyDealMatches
Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
cnx = pymysql.connect(user='root', password='root',
host='127.0.0.1',
database='sampledb')
cursor = cnx.cursor()
query = ("SELECT id FROM ProductDetails")
cursor.execute(query)
dealMatchGauranteed=[]
for product in cursor:
for tweet in timeline:
deal = (tweet['text'])
if (len(re.findall('\s'+product[0]+'\s',deal)) >= 1):
dealMatchGauranteed = dealMatchGauranteed + [deal]
In [101]:
Sanity Test that we got some deals
dealMatchGauranteed[:5]
Out[101]:
['Save $150 on the HP 14" Touch Screen Laptop Intel Core i3 4GB Memory 128GB Solid State Drive - Ash Silver Keyboard Frame. #Deal',
'Save $70 on the HP Pavilion x360 2-in-1 11.6" Touch-Screen Laptop Intel Pentium 4GB Memory 500GB Hard Drive HP Fini… https://t.co/M3VZy68Oko',
'Save $80 on the HP 17.3" Laptop Intel Core i5 8GB Memory 1TB Hard Drive - Black. #Deal',
'Save $320 HP Desktop Intel Core i7 8GB Memory 1TB Hard Drive - Black. #Deal',
'Save $70 on the HP 14" Laptop Intel Core i3 4GB Memory 128GB Solid State Drive - Ash Silver. #Deal']
Create and write the deals into DealMatches.txt file that will be used by web-app of SmartPortables to display two deal matches
In [102]:
dealMatchFile = open(r'C:\apache-tomcat-7.0.34\webapps\Tutorial_3\DealMatches.txt', 'w')
for deal in dealMatchGauranteed:
dealMatchFile.write("%s\n" % deal)
Type here to search
ENG
9:07 PM
11/22/2020


ProductRecommender
Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
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()
a ['philips-hiFi', 'Iphone-11', 'Samsung-Note10']
k ['Amazon-echo', 'Samsung-Note10']
i ['Iphone-11', 'Samsung-QLED', 'Lenovo-ideapad']
In [ ]:
Type here to search
ENG
9:07 PM
11/22/2020


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SCHEMAS
sampledb
use sampledb;
CREATE TABLE `customerorders` (
`OrderId` int NOT NULL,
`userName` varchar(40) NOT NULL,
`orderName` varchar(40) NOT NULL,
`orderPrice` double DEFAULT NULL,
`userAddress` varchar(40) DEFAULT NULL,
`creditCardNo` varchar(40) DEFAULT NULL,
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Table: customerorders
Columns:
OrderId
userName
orderName
orderPrice
userAddress
creditCardNo
orderTime
customerAge
customerCity
customerStreet
customerZip
deliveryType
deliveryLocation
expectedDate
retailerName
Output
Action Output
Type here to search
ENG
9:08 PM
11/22/2020


AutoSave
sql_train
Search
Ketan Pansare
File Home Insert Draw Page Layout Formulas Data Review View Help
Share
Comments
Paste
Cut
Copy
Format Painter
Clipboard
Calibri
Font
Wrap Text
Merge & Center
Alignment
General
Number
Conditional Formatting
Format as Table
Cell Styles
Styles
Insert
Delete
Format
Cells
AutoSum
Fill
Clear
Sort & Filter
Find & Select
Editing
Analyze Data
Analysis
Sensitivity
A1
loginID
loginID Product_ID Review_Rating
a Amazon-ec 1
k Iphone-11 3
k Lenovo-ide 2
k Samsung-C 4
i Samsung-N 1
k philips-hiFi 4
sql_train
Type here to search
ENG
9:10 PM
11/22/2020


sql_test
Ketan Pansare
Amazon-ec 1
k Iphone-11 3
k Lenovo-ide 2
k Samsung-C 4
i Samsung-N 1
k philips-hiFi 4
sql_test
Ready
Type here to search
9:10 PM
11/22/2020


matrixFactorizationBasedRecommendations
Ketan Pansare
a ['philips-hiFi', 'Iphone-11', 'Samsung-Note10']
k ['Amazon-echo', 'Samsung-Note10']
i ['Iphone-11', 'Samsung-QLED', 'Lenovo-ideapad']
matrixFactorizationBasedRecomme
Ready
Type here to search
9:11 PM
11/22/2020