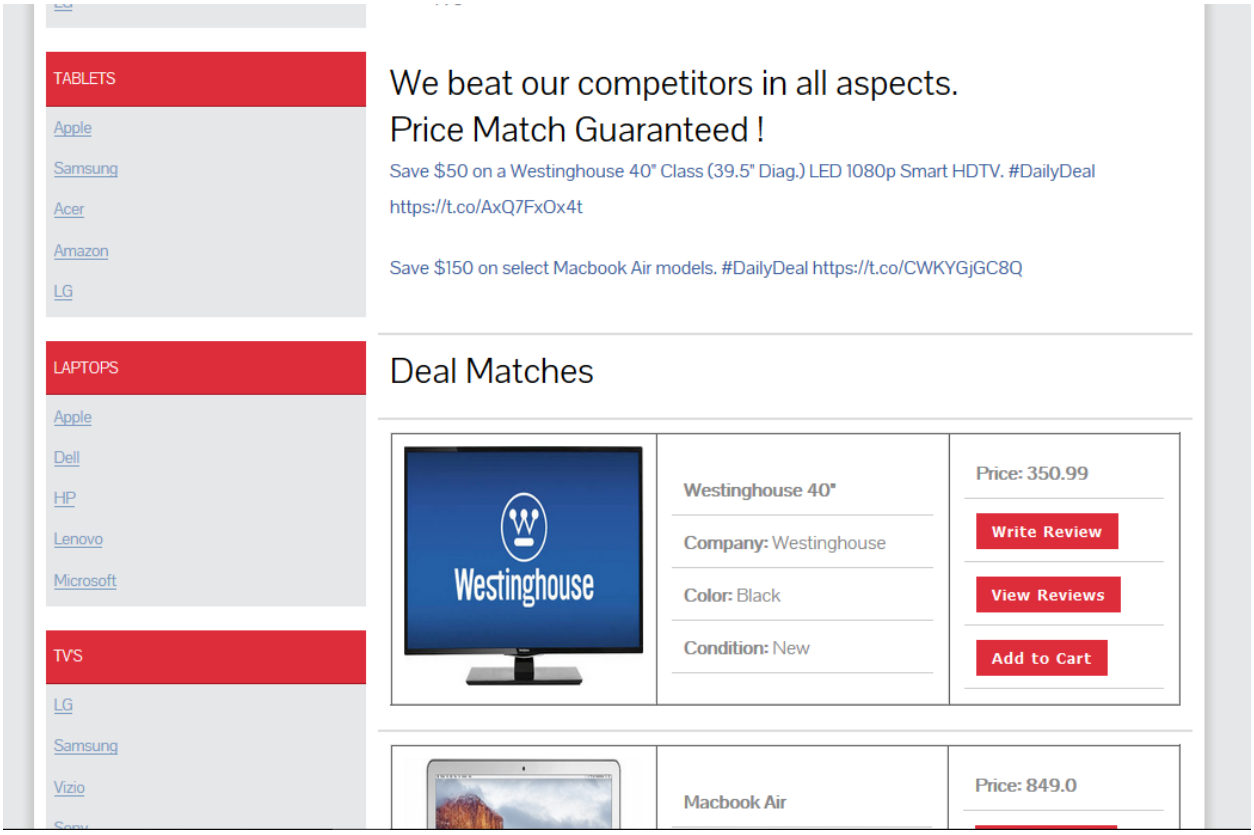
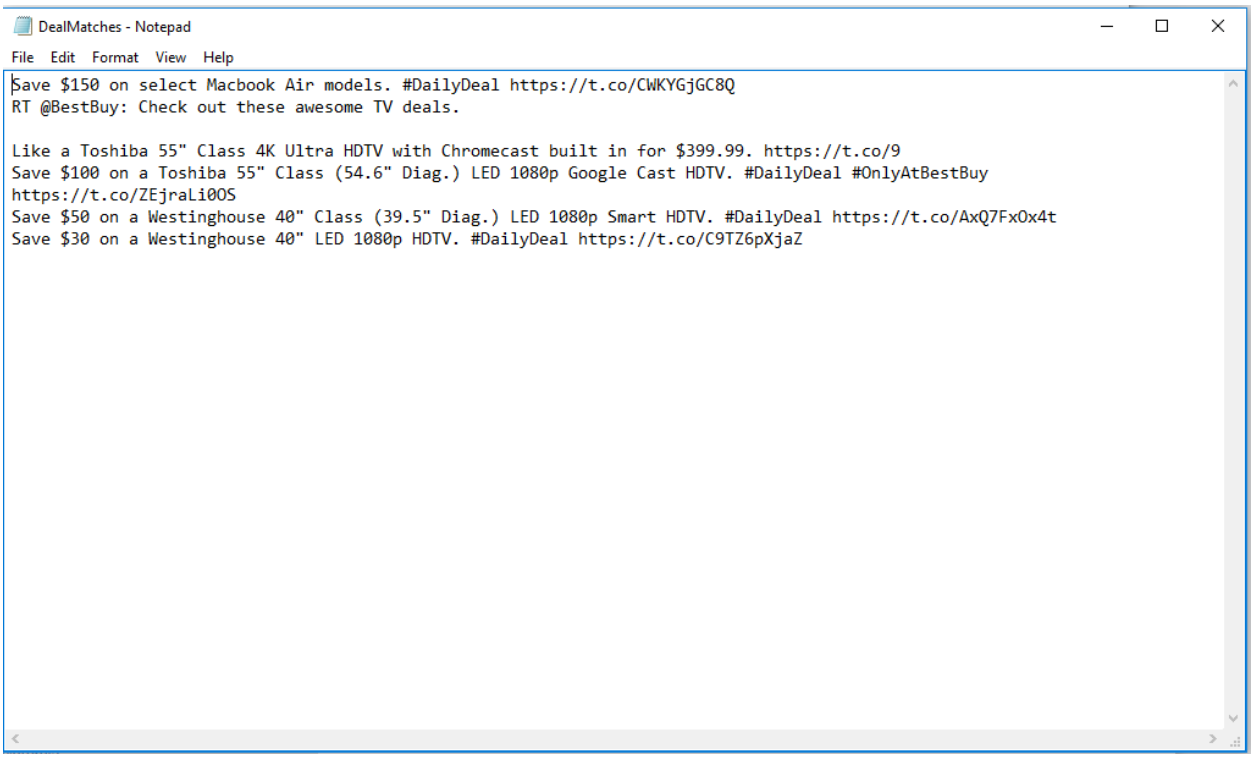
1. Home Page showing 2 tweets and 2 products matched from MySQL table Products



2. DealMatches.txt where tweets are stored



3. Python Script for fetching tweets and storing it in txt file after comparing from MySQL table

jupyter bestbuydealmatches Last Checkpoint: 11/12/2016 (autosaved)

File Edit View Insert Cell Kernel Help Python 2

Markdown CellToolbar

Find the deals in the BestBuy_Deal tweets that match products in BestDeal MySQL product table

In [326]:
```
import re
import pymysql

cnx = pymysql.connect(user='root', password='root',
                              host='127.0.0.1',
                              database='bestdealdatabase')
cursor = cnx.cursor()

query = ("SELECT name FROM products")
cursor.execute(query)


dealMatchGauranteed=[]
for product in cursor:
    for tweet in timeline:
        deal = (tweet['text']).encode('ascii','ignore')
        if (len(re.findall(r'\s'+product[0]+'\s',deal)) >= 1):
            dealMatchGauranteed = dealMatchGauranteed + [deal]
```

In [ ]:

In [327]:
```
# Sanity Test that  we got some deals
dealMatchGauranteed[:5]
```

4. DealMatches.java where code to get 2 tweets and products from MySQL

DealMatches.java | HomeServlet.java | MySqlDataStoreUtilities.java | MongoDBDataStoreUtilities.java

```
import java.util.ArrayList;
import java.util.List;
import java.util.HashMap;
import java.io.*;
import java.util.*;

public class DealMatches implements java.io.Serializable {

    HashMap<String, Product> selectedProducts;

    HashMap<String, Product> productMap;

    ArrayList<Object> products;

    ArrayList<String> tweets = new ArrayList<String>();

    public ArrayList<String> getTweets()
    {
        return tweets;
    }

    public HashMap<String, Product> getSelectedProductsFromTweets()
    {
        try{
            //Process p = Runtime.getRuntime().exec("python C:\\apache-tomcat-7.0.34\\webapps\\assignment5\\BestBuyDealMatches.ipynb");
        }
        catch(Exception e)
        {
            e.printStackTrace();
        }


        selectedProducts = new HashMap<String, Product>();

        products = MySqlDataStoreUtilities.getProductsFromMySQL();

        productMap = (HashMap<String, Product>)products.get(5);

        tweets = new ArrayList<String>();
```

Java source file | length : 2023 lines : 95 | Ln : 7 Col : 25 Sel : 11 | 1 | Dos\Windows | UTF-8 | INS

5.HomeServlet.java where tweets and products are displayed

DealMatches.java | HomeServlet.java | MySqlDataStoreUtilities.java | MongoDBDataStoreUtilities.java

```java
        if(tweets.isEmpty())
        {
            out.println("<p style='color:#325b9e'>"+"No Offers Found !"+"</p>");
        }
        else
        {
            for(String tweet: tweets)
            {
                out.println("<p style='color:#325b9e'>"+tweet+"</p>");
            }
        }

        out.println("</article><article><h2>Deal Matches</h2></article>");

        String fname = null;

        if(selectedProducts.isEmpty())
        {
            out.println("<article>");
            out.println("<p style='color:#325b9e'>"+"No Deals Found !"+"</p>");
            out.println("</article>");
        }
        else
        {
            for(Map.Entry<String,Product> m :selectedProducts.entrySet()){

                Product s = m.getValue();

                String productType = s.getType();

                out.println("<article>");
                out.println("<table style=\"width:100%\" style=\"height:100%\" border=\"1\" bordercolor=\"#aaa\" cellspacing=\"0\" cellpadding=\"0\">");
                out.println("<tr><td width=\"30%\">");
                out.println("<a href=\"ProductDetails.html\"><img style=\"width:200px;height:200px;\" style=\"display:block;\"  src=\"");
                out.println(s.getImage());
                out.println("\" /></a>");
                out.println("</td>");
                out.println("<td width=\"40%\"><table><tr><td width=\"40\"><b>");
```

Java source file | length : 12533 lines : 226 | Ln : 6 Col : 25 Sel : 11 | 1 | Dos\Windows | UTF-8 | INS

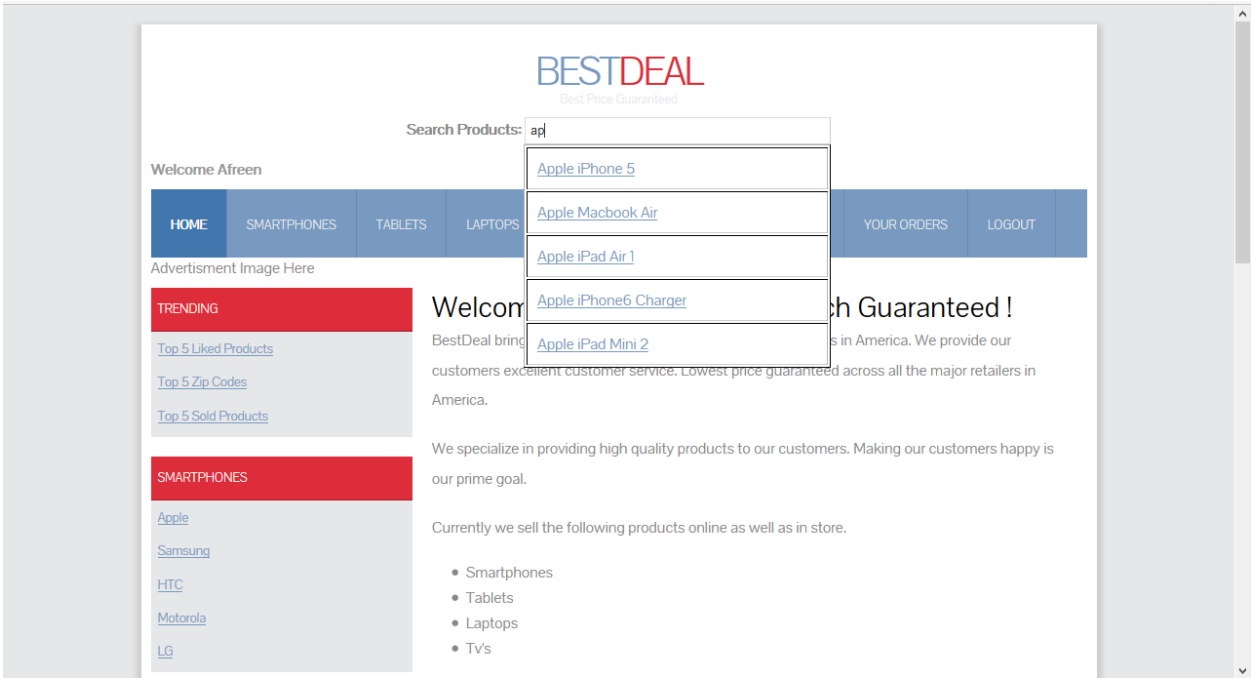
6.MySQLDataStoreUtilities.java showing method to get products from MySQL table

DealMatches.java | HomeServlet.java | MySqlDataStoreUtilities.java | MongoDBDataStoreUtilities.java

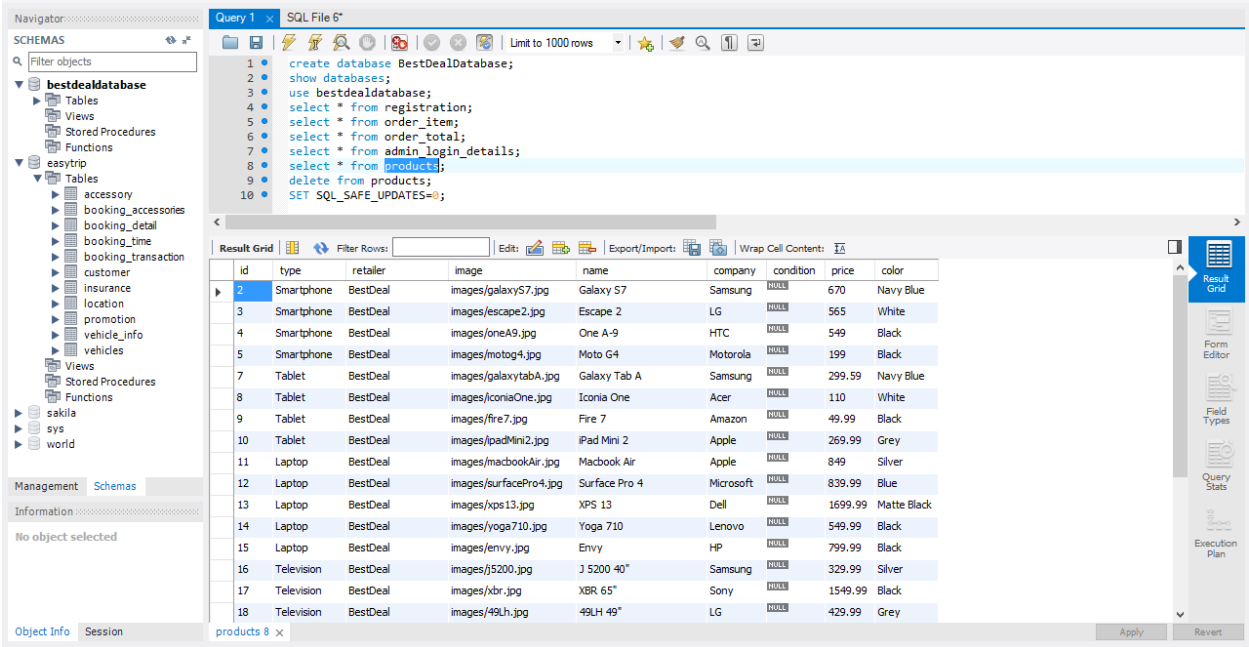
```java
    public static Connection getConnection()
    {

    public static ArrayList<Object> getProductsFromMySQL()
    {
        ArrayList<Object> products = new ArrayList<Object>();

        HashMap<String, Smartphone> smartphones= new HashMap<String, Smartphone>();
        HashMap<String, Laptop> laptops= new HashMap<String, Laptop>();
        HashMap<String, Tablet> tablets= new HashMap<String, Tablet>();
        HashMap<String, Television> televisions= new HashMap<String, Television>();
        HashMap<String, Accessory> accessories= new HashMap<String, Accessory>();

        HashMap<String, Product> productsMap= new HashMap<String, Product>();

        Smartphone smartphone;
        Laptop laptop;
        Tablet tablet;
        Television television;
        Accessory accessory;
        Product product;

        try
        {
            Connection conn = null;

            Class.forName("com.mysql.jdbc.Driver").newInstance();
            conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/bestdealdatabase?autoReconnect=true&useSSL=false", "root", "root");

            Statement s = conn.createStatement ();
            s.executeQuery ("SELECT * FROM products");
            ResultSet rs = s.getResultSet();

            while (rs.next ())
            {
                Integer id1 = rs.getInt("id");
                String id = id1.toString();
                String type = rs.getString("type");
                String retailer = rs.getString ("retailer");
```

Java source file | length : 23701 lines : 802 | Ln : 31 Col : 57 Sel : 20 | 1 | Dos\Windows | UTF-8 | INS

7. Page showing AJAX Auto-Completion Feature

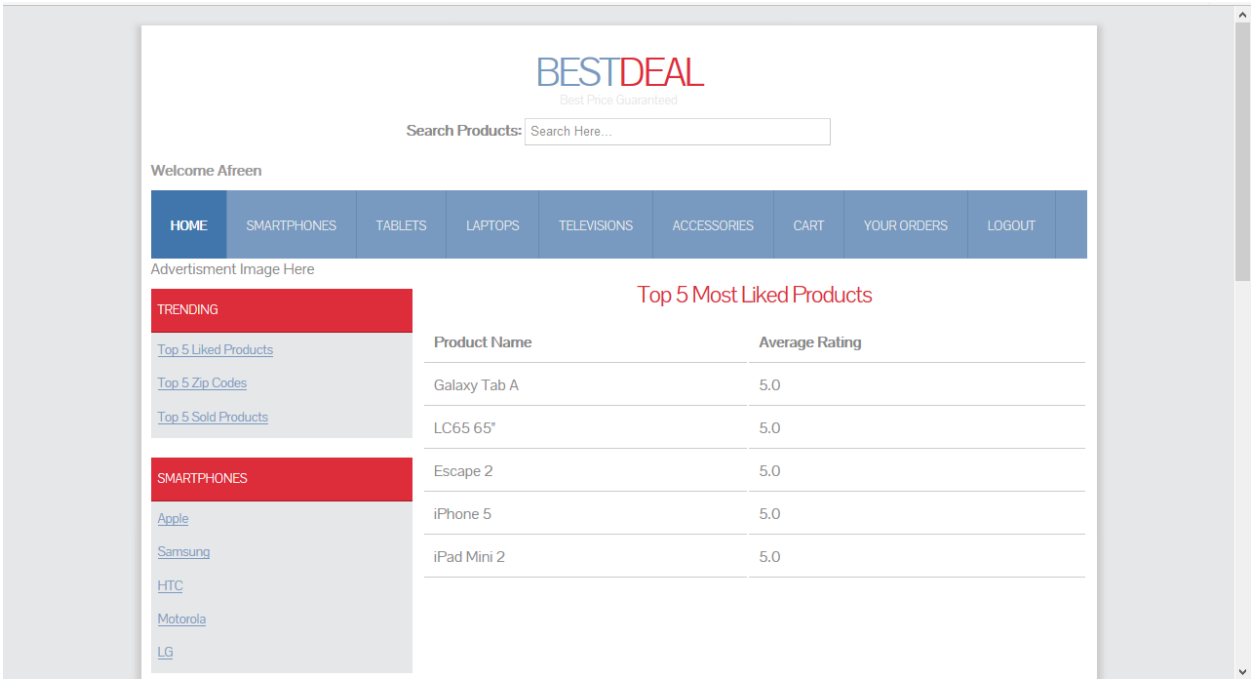


8.MySQL Database showing products table content

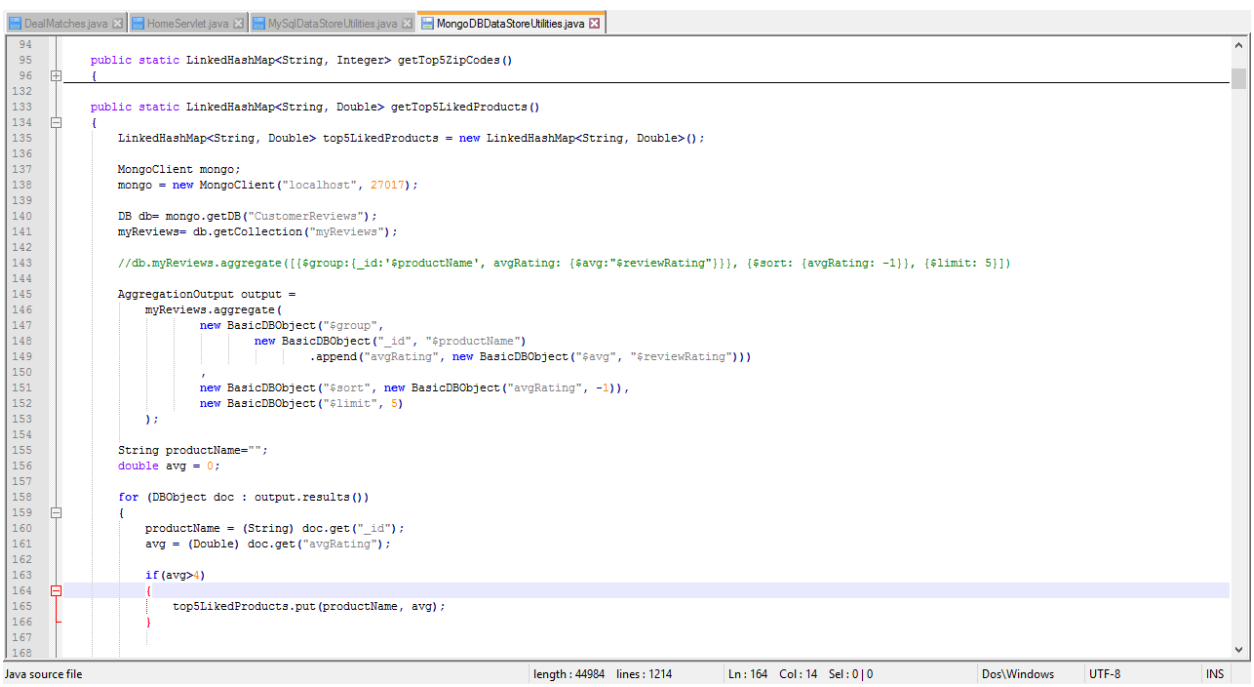


id	type	retailer	image	name	company	condition	price	color
2	Smartphone	BestDeal	images/galaxyS7.jpg	Galaxy S7	Samsung	NULL	670	Navy Blue
3	Smartphone	BestDeal	images/escape2.jpg	Escape 2	LG	NULL	565	White
4	Smartphone	BestDeal	images/oneA9.jpg	One A-9	HTC	NULL	549	Black
5	Smartphone	BestDeal	images/motog4.jpg	Moto G4	Motorola	NULL	199	Black
7	Tablet	BestDeal	images/galaxytabA.jpg	Galaxy Tab A	Samsung	NULL	299.59	Navy Blue
8	Tablet	BestDeal	images/iconiaOne.jpg	Iconia One	Acer	NULL	110	White
9	Tablet	BestDeal	images/fire7.jpg	Fire 7	Amazon	NULL	49.99	Black
10	Tablet	BestDeal	images/ipadMini2.jpg	iPad Mini 2	Apple	NULL	269.99	Grey
11	Laptop	BestDeal	images/macbookAir.jpg	Macbook Air	Apple	NULL	849	Silver
12	Laptop	BestDeal	images/surfacePro4.jpg	Surface Pro 4	Microsoft	NULL	839.99	Blue
13	Laptop	BestDeal	images/xps13.jpg	XPS 13	Dell	NULL	1699.99	Matte Black
14	Laptop	BestDeal	images/yoga710.jpg	Yoga 710	Lenovo	NULL	549.99	Black
15	Laptop	BestDeal	images/envy.jpg	Envy	HP	NULL	799.99	Black
16	Television	BestDeal	images/j5200.jpg	J 5200 40"	Samsung	NULL	329.99	Silver
17	Television	BestDeal	images/xbr.jpg	XBR 65"	Sony	NULL	1549.99	Black
18	Television	BestDeal	images/49Lh.jpg	49LH 49"	LG	NULL	429.99	Grey

9. Showing Trending for Top 5 Most Liked Products



10. Changed code for Top 5 Liked Products (added the '>4' logic to display) where marks were deducted in the last assignment



```
public static LinkedHashMap<String, Integer> getTop5ZipCodes()
{

public static LinkedHashMap<String, Double> getTop5LikedProducts()
{
    LinkedHashMap<String, Double> top5LikedProducts = new LinkedHashMap<String, Double>();

    MongoClient mongo;
    mongo = new MongoClient("localhost", 27017);

    DB db= mongo.getDB("CustomerReviews");
    myReviews= db.getCollection("myReviews");

    //db.myReviews.aggregate([{$group:{_id:'$productName', avgRating: {$avg:"$reviewRating"}}}, {$sort: {avgRating: -1}}, {$limit: 5}])

    AggregationOutput output =
        myReviews.aggregate(
                new BasicDBObject("$group",
                        new BasicDBObject("_id", "$productName")
                                .append("avgRating", new BasicDBObject("$avg", "$reviewRating")))
                ,
                new BasicDBObject("$sort", new BasicDBObject("avgRating", -1)),
                new BasicDBObject("$limit", 5)
        );

    String productName="";
    double avg = 0;

    for (DBObject doc : output.results())
    {
        productName = (String) doc.get("_id");
        avg = (Double) doc.get("avgRating");

        if(avg>4)
        {
            top5LikedProducts.put(productName, avg);
        }
```