

1. Deals at Home page:



2. Deal matched by the python script:

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

In [506]: 1 import re
           2 import pymysql
           3
           4 cnx = pymysql.connect(user='root', password='ashish',
           5                 host='127.0.0.1',
           6                 database='smartpotables')
           7 cursor = cnx.cursor()
           8
           9 query = ("SELECT ProductName FROM Product;")
          10 cursor.execute(query)
          11
          12
          13 dealMatchGauranteed=[]
          14 for product in cursor:
          15     for tweet in timeline:
          16         deal = (tweet['text'])
          17         if (len(re.findall(product[0],deal)) >= 1):
          18             dealMatchGauranteed = dealMatchGauranteed + [deal]
          19

In [507]: 1 # Sanity Test that we got some deals
           2 dealMatchGauranteed[:5]


Out[507]: ['Save $250 on the Dell 2 in 1 15.6" 4K Ultra HD Touch Screen Laptop Intel Core i7 16GB Memory NVIDIA GeForce MX130 5... https://t.co/PMvRfawWld',
           'Save $100 on the HP 2 in 1 12.3" Touch Screen Chromebook Intel Core M 4GB Memory 32GB - Ceramic White. #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 [508]: 1 dealMatchFile = open(r'C:\apache-tomcat-7.0.34\webapps\SmartPortables_5\DealMatches.txt', 'w')
           2
           3 for deal in dealMatchGauranteed:
           4     dealMatchFile.write("%s\n" % deal)
           5
           6 dealMatchFile.close()
```

3.Product Recommender:

[Home](#) | [Wearable Technology](#) | [Voice Assistant/Smart Speaker](#) | [Headphones](#) | [Phones](#) | [Laptops](#) | [Trending](#) | [Hi adam, Logout](#) | [ViewOrder](#) | [Cart\(1\)](#)



ABOUT US

Welcome to Smart-portables. Easy on line shopping with speedy delivery. Available 24 x 7. One click to start browsing through your favorite portable electronics devices. Enjoy Shopping !!!

WEARABLE TECHNOLOGY

- [Fitness Watches LG](#)
- [Samsung Smart Watches](#)
- [Apple Smart/Fitness Watches](#)
- [Virtual Reality](#)
- [Pet Tracker](#)

VOICE ASSISTANT/SMART SPEAKER

- [JBL Speaker](#)
- [Sony Speaker](#)
- [Beats Speaker](#)

HEADPHONES


- [Sennheiser](#)
- [SkullCandy](#)
- [Sony](#)

PHONES

- [Apple](#)
- [One Plus](#)
- [Samsung](#)

LAPTOPS

- [Apple](#)
- [One Plus](#)
- [Samsung](#)



- Name : LG Watch Sport Unlocked GSM
- Price: \$399
- Quantity=1

Total Price :\$ 399.0

Card Holder Name


Card Number

Address

Zip Code

[CHECKOUT](#) [REMOVE](#)

Recommended Products




Apple 13.3 MacBook Air (Silver)

Price:1099\$

[ADD TO CART](#) [WRITE REVIEW](#) [READ REVIEW](#)

Second Product:

[Home](#) | [Wearable Technology](#) | [Voice Assistant/Smart Speaker](#) | [Headphones](#) | [Phones](#) | [Laptops](#) | [Trending](#) | [Hi adam, Logout](#) | [ViewOrder](#) | [Cart\(1\)](#)



ABOUT US

Welcome to Smart-portables. Easy on line shopping with speedy delivery. Available 24 x 7. One click to start browsing through your favorite portable electronics devices. Enjoy Shopping !!!

WEARABLE TECHNOLOGY

- [Fitness Watches LG](#)
- [Samsung Smart Watches](#)
- [Apple Smart/Fitness Watches](#)
- [Virtual Reality](#)
- [Pet Tracker](#)

VOICE ASSISTANT/SMART SPEAKER

- [JBL Speaker](#)
- [Sony Speaker](#)
- [Beats Speaker](#)

HEADPHONES


- [Sennheiser](#)
- [SkullCandy](#)
- [Sony](#)

PHONES

- [Apple](#)
- [One Plus](#)
- [Samsung](#)

LAPTOPS

- [Apple](#)
- [One Plus](#)
- [Samsung](#)



- Name : LG Watch Sport Unlocked GSM
- Price: \$399
- Quantity=1

Total Price :\$ 399.0

Card Holder Name


Card Number

Address

Zip Code

[CHECKOUT](#) [REMOVE](#)

Recommended Products

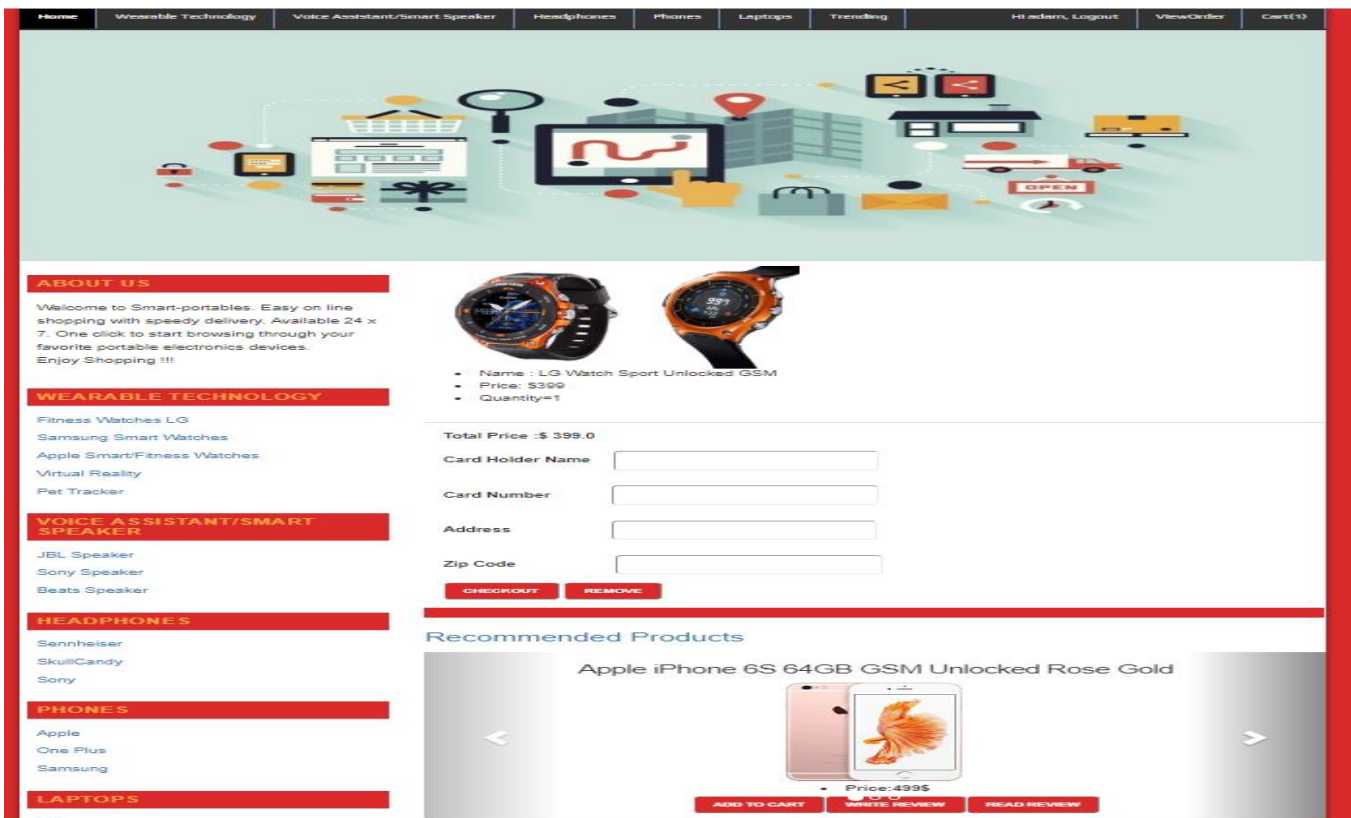


JBL Flip 4 Waterproof Portable Bluetooth Speaker

Price:149\$

[ADD TO CART](#) [WRITE REVIEW](#) [READ REVIEW](#)

Third product:



Product recommended by python script to Adam:

```
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
```

```
adam ['Apple iPhone 6S 64GB GSM Unlocked Rose Gold', 'Apple 13.3 MacBook Air (Silver)', 'JBL Flip 4 Waterproof Portable Bluetooth Speaker']
ashish ['Samsung Galaxy J3 Pro', 'JBL Xtreme 2 Portable Waterproof Wireless Bluetooth Speaker', 'Dell 2 in 1 15.6']
```

Deal Match Code:

```
import java.util.Map;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/DealMatchesUtilities")

public class DealMatchesUtilities extends HttpServlet {

    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter pw = response.getWriter();

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

        try {
            pw.print("<section id='content'>");
            pw.print("<h3>We beat our competitors in all aspects. Price-Match Guaranteed</h3>");

            String line = null;
            String TOMCAT_HOME = System.getProperty("catalina.home");
            HashMap<String, Product> productmap = MySQLDataStoreUtilities.getProducts();

            for (Map.Entry<String, Product> entry : productmap.entrySet()) {
                if (selectedproducts.size() < 2 && !selectedproducts.containsKey(entry.getKey())) {
                    BufferedReader reader = new BufferedReader(new FileReader(new File("C:\\apache-tomcat-7.0.34\\webapps\\SmartPortables_5\\DealMatches.txt")));
                    line = reader.readLine();

                    if (line == null) {
                        pw.print("<h4 align='center'>No Offers Found</h4>");
                        break;
                    } else {
                        do {
                            Product product = (Product) entry.getValue();
                            if (line.contains(product.getName())) {
                                pw.print("<h4>" + line + "</h4>");
                                pw.print("<br>");
                                selectedproducts.put(entry.getKey(), entry.getValue());
                                break;
                            }
                        } while ((line = reader.readLine()) != null);
                    }
                }
            } catch (Exception e) {
                pw.print("<h4 align='center'>No Offers Found</h4>");
            }

            pw.print("<div><div class='post'><h2 class='title meta'>");
            pw.print("<a style='font-size: 24px;'>Deal Matches</a>");
            pw.print("</h2><div class='entry'>");
            if (selectedproducts.size() == 0) {
                pw.print("<h4 align='center'>No Deals Found</h4>");
            } else {
                pw.print("<article>");
                pw.print("<table cellpadding='0' class='shopping-cart'>");
                pw.print("<thead>");
                pw.print("<tr class='headings'>");
                pw.print("<th class='link'>");
                pw.print("<th class='link'>");
                pw.print("</tr>");
                pw.print("</thead>");
                pw.print("<tbody>");
                pw.print("<tr>");
                for (Map.Entry<String, Product> entry : selectedproducts.entrySet()) {
                    Product product = (Product) entry.getValue();
                    pw.print("<a href='\"#\"'>");
                    pw.print("<img src='\"Html\\images\\\" + product.getImage()");
                    pw.print("<br>\" alt='\"Mollunga Norway\" width='\"300\" height='\"200\"'>");
                    pw.print("</a>");
                    pw.print("<ul>");
                    pw.print("<li class='desc'>Name : " + product.getName() + "</li>");
                    pw.print("<li>Price: $" + product.getPrice() + "</li>");
                    pw.print("<button class='\"button\"'><a href='\"AddCart?id=" + product.getId() + "&name="");
                    pw.print("<br>\" + product.getName() + "&price=" + product.getPrice() + "&image=" + product.getImage()");
                    pw.print("<br>\" class='\"btnreview\" style='\"color:white\"'>Add To Cart</a></button>");

                    pw.print("<button class='\"button\"'><a href='\"WriteReview?id=" + product.getId() + "&name="");
                    pw.print("<br>\" + product.getName() + "&price=" + product.getPrice() + "&image=" + product.getImage() + "&category=" + product.getCategory() +");
                    pw.print("<br>\" + product.getManufacturer()");
                    pw.print("<br>\" class='\"btnreview\" style='\"color:white\"'>Write Review</a></button>");

                    pw.print("<button class='\"button\"'><a href='\"ViewReview?id=" + product.getId() + "&name="");
                    pw.print("<br>\" + product.getName() + "&price=" + product.getPrice() + "&image=" + product.getImage() + "&category=" + product.getCategory() +");
                    pw.print("<br>\" + product.getManufacturer()");
                    pw.print("<br>\" class='\"btnreview\" style='\"color:white\"'>Read Review</a></button>");
                }
                pw.print("</ul>");
            }
            pw.print("</tr>");
            pw.print("</tbody>");
            pw.print("</table>");
            pw.print("</div>");
        }
    }
}
```

Recommender Code:

```
import java.io.*;
import java.sql.*;
import java.io.IOException;
import java.util.*;

public class ProductRecommenderUtility{

    static Connection conn = null;
    static String message;

    public static String getConnection()
    {
        try
        {
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/smartpotables","root","ashish");
            message="Successful";
            return message;
        }
        catch(SQLException e)
        {
            message="unsuccessful";
            return message;
        }
        catch(Exception e)
        {
            message="unsuccessful";
            return message;
        }
    }

    public HashMap<String,String> readOutputFile() {

        String TOMCAT_HOME = System.getProperty("catalina.home");
        BufferedReader br = null;
        String line = "";
        String cvsSplitBy = ",";
        HashMap<String,String> prodRecmMap = new HashMap<String,String>();
        try {

            br = new BufferedReader(new FileReader(new File("C:\\apache-tomcat-7.0.34\\webapps\\SmartPortables_5\\output.csv")));
            while ((line = br.readLine()) != null) {

                // use comma as separator
                String[] prod_rec = line.split(cvsSplitBy,2);
                prodRecmMap.put(prod_rec[0],prod_rec[1]);

            }

        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (br != null) {
                try {
                    br.close();
                } catch (IOException e) {
                }
            }
        }
    }

    public static Product getProduct(String product) {
        Product prodObj = new Product();
        try
        {
            String msg = getConnection();
            String selectProd="select * from Product where ProductName='"+product+"'";
            PreparedStatement pst = conn.prepareStatement(selectProd);
            ResultSet rs = pst.executeQuery();
            while(rs.next())
            {
                prodObj = new Product(rs.getString("ProductName")
                    ,rs.getString("Category"),rs.getString("ProductCondition"),
                    rs.getString("Manufacturer"),rs.getString("ProductImage"),
                    rs.getInt("Quantity"),(int)rs.getDouble("Price"),
                    rs.getString("OnSale"),rs.getString("Rebate"),rs.getInt("Id"));
            }
            rs.close();
            pst.close();
            conn.close();
        }
        catch(Exception e)
        {
        }
        return prodObj;
    }
}
```