

Assignment #2

Due Date: 10/14/18 by 11:59pm

Deliverable:

- Use the object-oriented design principles and utilize the MVC architecture discussed in the class to produce an object-oriented web-based enterprise application that is reusable, flexible, and extensible.
- Use **Servlets** to implement the functionalities listed below.
- Record 10 minutes demo of your assignment's run using screencast. The tool can be downloaded from this URL <http://screencast-o-matic.com/home>
- Capture most important 10 screen-shots of your output and save them in a file called output.pdf
- All source code and byte code shall be submitted.
- Readme text file that illustrates how to compile/install/run your application
- Post your homework as a single zipped file on Blackboard with the name "HW2_YourLastName,FirstName"

Important Notes:

- NO IDE to be used in any shape/form in the implementation of this assignment
- Do NOT communicate or share your assignment with others

High-Level Requirements:

Extend Assignment #1 for the online retailer SmartPortables to add the following features:

- All accounts login information must be stored in SQL database (MySQL)
- All Customers transactions/orders must be stored in SQL database (MySQL)
- All order updates to insert/delete/update orders must be reflected in the MySQL database; not only the HashMap objects
- Customer must be able to submit product reviews
- Product reviews must be stored in NoSQL database (MongoDB)
- Add Trending & Data Analytics feature (detailed below)
- All new code added for MySQL shall be placed in a class called MySQLDataStoreUtilities.java
- All new code added for MongoDB shall be placed in a class called MongoDBDataStoreUtilities.java

Required Functionalities:

- Extend Assignment #1 to use MySQL and MongoDB database engines to support the following functionalities.
- Use MySQL to store all accounts login information
- Use MySQL to store All Customers transactions/orders
- Extend Assignment #1 in order to allow the customer to write and submit a **Product Review** online that has the following form:
 1. ProductModelName: Samsung Galaxy 6
 2. ProductCategory: phone
 3. ProductPrice: \$499
 4. RetailerName: SmartPortables
 5. RetailerZip: 60616
 6. RetailerCity: Chicago
 7. RetailerState: IL
 8. ProductOnSale: Yes
 9. ManufacturerName: Samsung
 10. ManufacturerRebate: Yes

11. UserID: whksa8
12. UserAge: 24
13. UserGender: Male
14. UserOccupation: accountant
15. ReviewRating: 4
16. ReviewDate: 12/15/2015
17. ReviewText: It has excellent video/audio clarity, however, it overheats after 5 hours of use

- Extend Assignment #1 to add **Trending** link on the left navigation bar that the user can use to see trends for sold products
- Once the user clicks the **Trending**, the user must be presented with :
 1. Top five most liked products
 2. Top five zip-codes where maximum number of products sold
 3. Top five most sold products regardless of the rating

Product Review Form:

The product review Form has the following fields:

1. ProductModelName: Samsung Galaxy 6
2. ProductCategory: phone
3. ProductPrice: \$499
4. RetailerName: SmartPortables
5. RetailerZip: 60616
6. RetailerCity: Chicago
7. RetailerState: IL
8. ProductOnSale: Yes
9. ManufacturerName: Samsung
10. ManufacturerRebate: Yes
11. UserID: whksa8
12. UserAge: 24
13. UserGender: Male
14. UserOccupation: accountant
15. ReviewRating: 4
16. ReviewDate: 12/15/2015
17. ReviewText: It has excellent video/audio clarity , however, it overheats after 5 hours of use

Bonus Features

if you want to earn EXTRA points
(50% Bonus points for this assignment)

The bonus points can be considered for grading ONLY if the functionalities listed above are completely implemented.

No hard-coded queries will be accepted. You are allowed only to extend the required functionalities listed above to implement the bonus features listed below. If you HARD-CODE the queries in your implementation, you will get ZERO credit for the bonus feature.

For the Data Visualization/Trending feature in requirement #12 listed below, consider Google charts documentation at the following URL:

<https://developers.google.com/chart/interactive/docs/gallery/barchart>

Requirements:

- Add the **Data Analytics** link that is accessible only to the Store Manager
- The **Data Analytics** link will allow the store manager to perform different analytical queries.
- The following are **only some examples (a sample of queries)** that your implementation must be able to answer (Please do NOT hard-code those queries in your source):
 1. Print the list of all products and their ratings
 2. Print a list of reviews where rating greater than 3
 3. Get a list of products that got review rating 5 and price more than thousand
 4. Print a list of how many reviews for every product
 5. Get the list of reviews for shoppers in Chicago

6. Find highest price product reviewed/sold in every city
7. Find highest price product reviewed/sold in every zip-code
8. Get the top 5 list of liked products for every city
9. Print a list of reviews grouped by City
10. Print a list of reviews grouped by zip-code
11. Get the total number of products reviewed and got Rating 5 in Every City
12. Provide a **Trending (Data Visualization)** button that shows a Bar Chart for the total number of submitted reviews of every product for top 3 most liked products in every city. (x-axis is the total number of reviews submitted, y-axis is the city name, the 3 bar charts per city will represent the 3 top products per city along with total number of reviews)
13. Print the median product prices per city
14. Get top 5 list of most liked and expensive products sorted by retailer name for every city
15. Get the top 5 list of most Disliked products sorted by retailer name for every city
16. Get the top 5 list of most Disliked products sorted by retailer name for every zip-code
17. Get the top 2 list of zip-codes where highest number of products got review rating 5
18. Get a list of reviews where reviewer age greater than 50 and the list is sorted by age in every city
19. Get the top 5 list of most liked products sorted by manufacturer name for every city
20. Search reviews text for keywords (pattern-matching) and print the list of reviews that have the matched keywords. For example, print the list of reviews that have " XBOX overheat" keywords in the review text