

Assignment #4

Due Date: 11/10/17 by 9:00am

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 "HW4_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#3 to add the **Search Auto-Completion** feature.

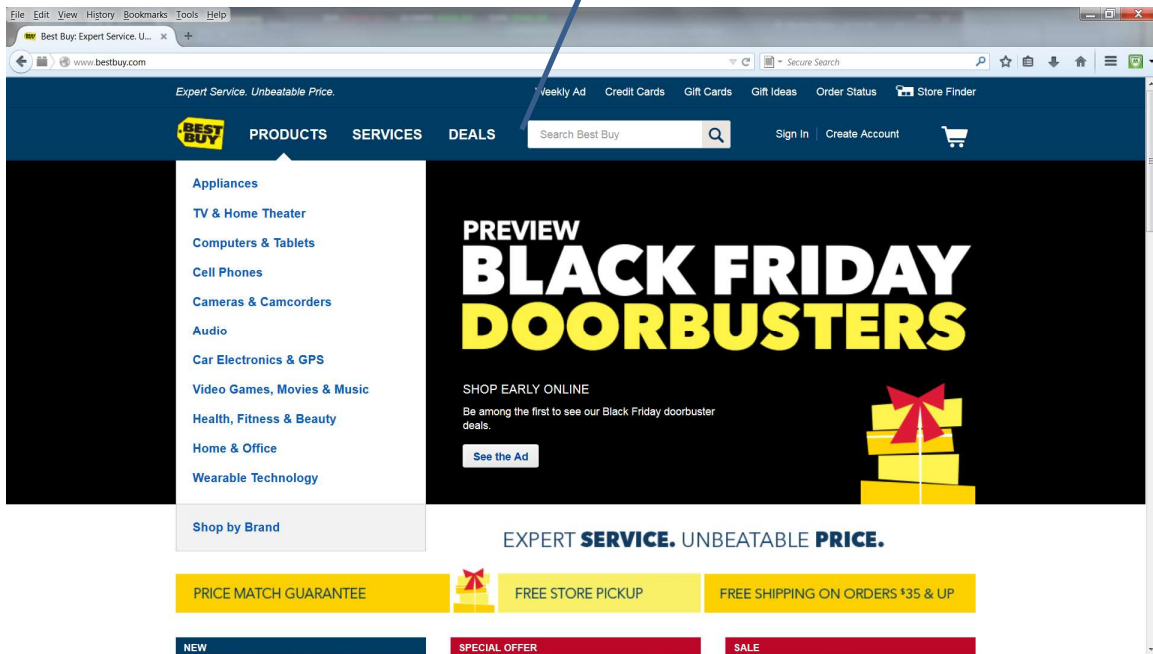
Required Functionalities:

See below an illustration for a screen-shot of BestBuy web site for the **Search Auto-Completion** feature

- Your auto-complete-feature must be implemented as follows:
When the app-server starts up, the Products are first read into a hashmap from ProductCatalog.xml file and then stored in MySQL database; follow this sequence.
- Since a store manager can insert/update/delete products, all of these operations must be reflected in the hashmap and then MySQL database
- All new code added for the auto-complete-complete feature shall be placed in a class called AjaxUtility.java

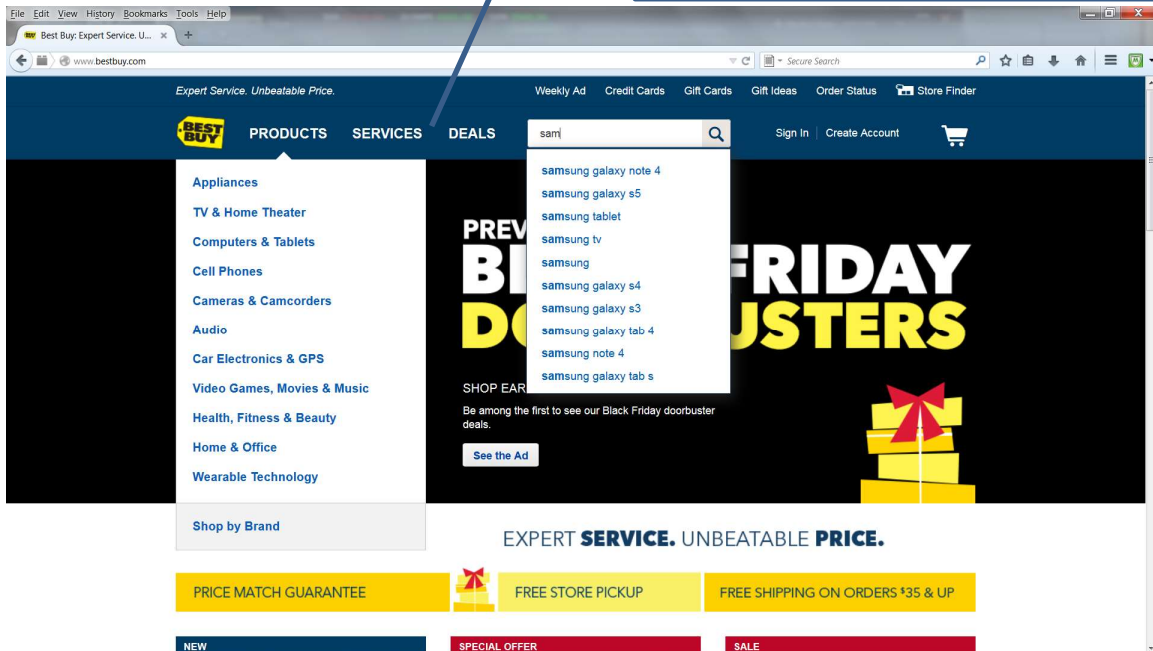
1

Textfield is available for users to enter search keywords



2

Once the user starts typing search keyword, a table of potential keyword-matches will be displayed



Bonus Feature

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.

- All new servlets code added for the Data Exploration feature must be placed in a class called DataExplorationUtility.java

No hard-coded data for the queries will be accepted. You write a servlet for the server backend to retrieve the data needed for the following queries from the MySQL and MongoDB databases/

For the choropleth feature listed below, use the d3js library URL:

<https://d3js.org/>

Requirements:

- Add the **Data Exploration** link that is accessible only to the Store Manager
- The **Data Exploration** link will allow the store manager to explore and visualize the data retrieved from MySQL and MongoDB.
- Create the USA Choropleth maps for every query listed below:
 1. Total number of products REVIEWED per state and their min, max and avg ratings; you read the data from MongoDB
 2. Total number of products BOUGHT per state and their min, max and avg ratings; you read the data from MySQL
 3. Total number of products reviewed and got Rating 5 in Every State; you read the data from MongoDB
 4. Average product prices for products sold in every state; you read the data from MySQL
 5. Total price of products sold in every state; you read the data from MySQL

File Edit View History Bookmarks Tools Help


D3.js - Data-Driven Documents

https://d3js.org

Overview Examples Documentation Source

D3 Data-Driven Documents

For more on D3.js



D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization

See more examples.

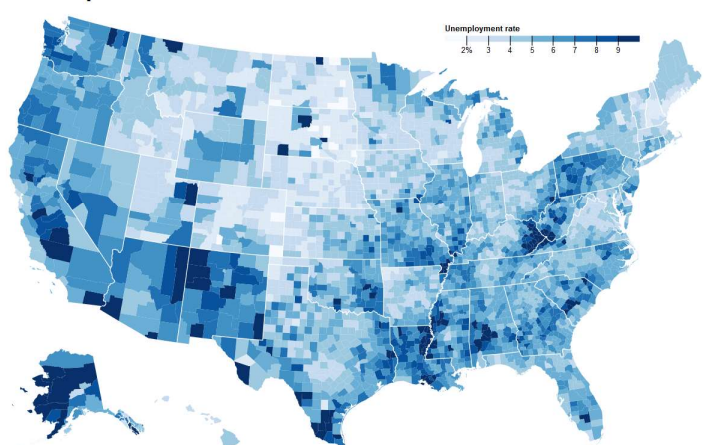
www.jasondavies.com/maps/clip/

File Edit View History Bookmarks Tools Help

Choropleth - bl.ocks.org

https://bl.ocks.org/mbostock/4060606

Choropleth



Unemployment rate

2% 3 4 5 6 7 8 9

Source: Bureau of Labor Statistics, Census Bureau

Open

choro Highlight All Match Case Whole Words 1 of 13 matches