

Assignment #3

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** or **React** to implement the features and functionalities listed below.
- Record 10 minutes demo of your assignment's run using Panopto.
- 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 Canvas with the name "HW3_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 #2 for the online retailer to add the following features:

1. Inventory Report
2. Sales Report
3. Search Auto-Completion

For the visual reports in the requirements listed below, consider Google charts documentation at the following URL:

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

Requirements:

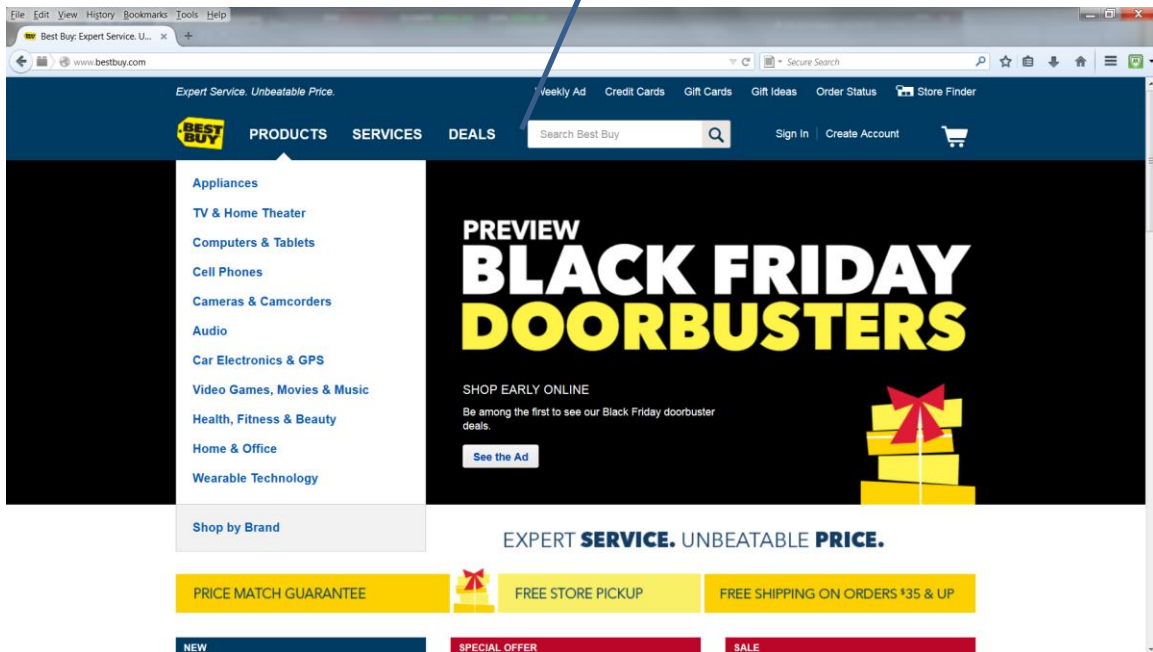
Add the **Inventory** and **Sales Reports** links that are accessible only to the Store Manager. And the **Search Auto-Completion** that is accessible to all users.

- Under the **Inventory** link, the store manager shall be able to:
 1. Generate a table of all products and how many items of every product currently available in the store; list only product name, price, how many items of that product available
 2. Generate a Bar Chart that shows the product names and the total number of items available for every product
 3. Generate a table of all products currently on sale
 4. Generate a table of all products currently that have manufacturer rebates
- Under the **Sales Report** link, the store manager shall be able to:
 1. Generate a table of all products sold and how many items of every product sold; list only product name, product price, number of items sold, and total sales of every product sold
 2. Generate a Bar Chart that shows the product names and the total sales for every product
 3. Generate a table of total daily sales transactions; that is, you list the dates and total sales for every day-date
- **Search Auto-Completion:** See below an illustration for a screenshot of BestBuy web site for the **Search Auto-Completion** feature. Your auto-complete-feature must be implemented as follows:

1. 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.
2. Since a store manager can insert/update/delete products, all of these operations must be reflected in the hashmap and then MySQL database
3. 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

