

Homework 8: eBay Item search using Bootstrap, JQuery, and Facebook Mashup

1. Objectives

- Become familiar with the AJAX, JSON & XML technologies.
- Use a combination of HTML, CSS, DOM, XMLHttpRequest, PHP and JSON.
- Get hands-on experience in Amazon cloud computing (AWS)
- Get hands-on experience on how to use Bootstrap and jQuery UI to enhance the user experience
- Provide an interface to perform eBay item search and post details to Facebook.

2. Background

2.1 AJAX & JSON

AJAX (Asynchronous JavaScript + XML) incorporates several technologies:

- Standards-based presentation using XHTML and CSS;
- Dynamic display and interaction using the Document Object Model (DOM);
- Data interchange and manipulation using XML and XSLT;
- Asynchronous data retrieval using XMLHttpRequest;
- JavaScript binding everything together.

See the class slides at <http://cs-server.usc.edu:45678/slides/ajax.pdf>

JSON, short for JavaScript Object Notation, is a lightweight data interchange format. Its main application is in AJAX web application programming, where it serves as an alternative to the use of the XML format for data exchange between client and server. See the class slides at <http://cs-server.usc.edu:45678/slides/JSON1.pdf>

2.2 eBay API

The eBay findItemsAdvanced API returns details about items matching your search criteria. A comprehensive reference about this API is available at:

<http://developer.eBay.com/DevZone/finding/CallRef/findItemsAdvanced.html>.

2.3 Bootstrap Library

Bootstrap is a free collection of tools for creating responsive websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. To learn more details about Bootstrap please refer to the lecture material on Responsive Design (See the class slides at <http://cs-server.usc.edu:45678/slides/Responsive.pdf>) and [http://en.wikipedia.org/wiki/Bootstrap_\(front-end_framework\)](http://en.wikipedia.org/wiki/Bootstrap_(front-end_framework))

2.4 Amazon Web Services (AWS)

AWS is Amazon's implementation of cloud computing. Included in AWS is Amazon Elastic Compute Cloud (EC2), which delivers scalable, pay-as-you-go compute capacity in the cloud, and AWS Elastic Beanstalk, an even easier way to quickly deploy and manage applications in the AWS cloud. You simply upload your application, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring. Elastic Beanstalk is built using familiar software stacks such as the Apache HTTP Server, PHP, and Python, Passenger for Ruby, IIS 7.5 for .NET, and Apache Tomcat for Java.

The Amazon Web Services homepage is available at:

<http://aws.amazon.com/>

2.5 Facebook

Facebook provides developers with an API called the Facebook Platform.

Facebook Connect is the next iteration of Platform, which provides a set of API's that enable Facebook members to log onto third-party websites, applications and mobile devices with their Facebook identity. While logged in, users can connect with friends via these media and post information and updates to their Facebook profile.

Below are a few links for Facebook Connect:

<https://developers.facebook.com/>

<https://developers.facebook.com/docs/javascript>

3. High Level Description

Similar to HW6, in this exercise you are asked to create a webpage that allows users to search items for sale on eBay.com using their API and the results will be displayed in a list format on the same page below the form.

The difference being, in this homework you will create a PHP file to return a JSON formatted data to the front-end (making asynchronous AJAX calls) and the

front-end will parse the JSON data and show it in a nicer-looking responsive UI (using Bootstrap JS).

A user will first open a page as shown below in Figure 1, where he/she can enter a keyword (mandatory) and search criteria for finding items in eBay if available. Description of Search Form is given in Section 4.1. Instructions on how to use the API are given in Section 5.

The screenshot shows the eBay Shopping search form. At the top is the eBay logo and the word "Shopping". Below it is a search bar labeled "Enter keyword". Underneath are sections for "Price range" (with "from (\$)" and "to (\$)" fields), "Condition" (checkboxes for New, Used, Very Good, Good, Acceptable), "Buying formats" (checkboxes for Buy It Now, Auction, Classified Ads), "Seller" (checkbox for Return accepted), "Shipping" (checkboxes for Free Shipping, Expedited shipping, with a "Max handling time(days)" input field), "Sort by" (dropdown menu set to "Best Match"), and "Results per page" (dropdown menu set to 5). At the bottom are "Clear" and "Search" buttons.

Figure 1: Initial Search Page

Once the user has provided data and clicks on the Search button, validation must be done to check that the data is valid. Form validation rules along with screenshots are given in Section 4.2.

Once the validation is successful, the jQuery function `.ajax()` is executed to start an asynchronous transaction with a PHP file (script) running on your AWS server, and passing the search form data as parameters of the transaction.

The PHP file you request is based on your HW#6, the difference is that this time the file does not need to display the data as HTML but instead will convert the XML data received from eBay to JSON and return it to your search webpage. The webpage must then use JavaScript to extract data from the JSON and display the results on the same webpage. Description of how to display results in success and failure states are given in Section 6.

4. SEARCH FORM

4.1. Description

You **must replicate** the form display as in Figure 1 using Bootstrap Form. For the header, eBay logo can be found at: <http://cs-server.usc.edu:45678/hw/hw8/ebay.jpg>. The form fields are same as your homework 6. The following is a brief description of the fields.

1. **Key Words:** This is a text box, which enables the user to search for matching items by inputting keywords. This must be a non-empty string.
2. **Price Range:** These are two text boxes, which enable the user to assign a price range for the matching items.
3. **Condition:** These are five check boxes “*New, Used, Very Good, Good, and Acceptable*” which enable the user to control the item condition retrieved in the results. The user can select one of the options or a combination of them to search for items in specific conditions.
4. **Buying formats:** These are three check boxes “*Buy It Now, Auction or Classified Ad*”. The user can select one of these options or a combination of them to control the buying format of items retrieved in the results.
5. **Seller:** Specifies if the user wants only items sold by a seller who accepts returns.
6. **Shipping:** Specify if only “free shipping” items should be returned; if only items available with expedited shipping should be returned, and/or maximum handling time, whose minimum is 1 day (time taken by the seller to deliver the purchased item to the carrier, NOT the shipment time). Default is “all” not specified.
7. **Sort By:** This specifies the ordering of the results table, which can be one of:
 - a. Best Match - ***THIS IS THE DEFAULT***
 - b. Price: highest first
 - c. Price + Shipping: highest first
 - d. Price + Shipping: lowest first
8. **Results Per Page:** This specifies the number of items to be retrieved per API call. **Default is 5**, but it can take the values of **5 or 10 ONLY**.

The search form has two buttons:

1. **SEARCH button:** On the button click validations must be performed. If success, then an AJAX asynchronous request is made to your web server (PHP file hosted on AWS), providing it with the form data that was entered. If failure, appropriate validation messages must be displayed as detailed in Section 4.2 and an AJAX call should **NOT** be made further with the invalid data.
2. **CLEAR button:** This button **must** clear the result area, all text fields, uncheck all checkboxes and reset “sort by” and “results per page” fields to their default values mentioned above. The clear operation is done using a JavaScript function.

Note: As we are making an asynchronous AJAX call in the process of generating results, the page isn't refreshed, hence you need not maintain state of the form fields explicitly as it was required in HW6.

Responsive to Devices

You are supposed to make the search form responsive to devices. If the page is loading on a smart phone or a tablet, the form should display according to the width of the devices. One example is shown in the Figure 2 below.

The figure shows a mobile browser displaying the eBay Shopping search form. At the top, the status bar indicates signal strength (2 bars), T-Mobile network, Wi-Fi icon, time (7:11 PM), battery level (38%), and a battery icon. The URL www-scf.usc.edu is visible. The main content is the eBay Shopping search form. It includes fields for "Key words:" (with a placeholder "Enter keyword"), "Price range:" (with input fields for "from (\$)" and "to (\$)"), "Condition:" (with checkboxes for New, Used, Very Good, Good, and Acceptable), "Buying formats:" (with checkboxes for Buy It Now, Auction, and Classified Ads), "Seller:" (with a checkbox for Return accepted), "Shipping:" (with checkboxes for Free Shipping and Expedited shipping), "Max handling time(days)" (an input field), "Sort by:" (a dropdown menu set to Best Match), "Results per page:" (a dropdown menu set to 5), and "Clear" and "Search" buttons at the bottom. The right side of the screen shows the continuation of the search form's content, which is typically cut off on a standard desktop view.

Figure 2: Search form on iPhone

4.2. Validations

If the user did not enter one of the data items as expected, then a message should be shown with appropriate text requesting the user to provide the information properly. The validations to be done along with the messages to be displayed are being listed below. The validation texts and CSS classes are all provided using iQuery validation and Bootstrap API. Popups are not acceptable.

Validation Set 1: Key word is a mandatory field

- 1.1. On click of submit button, if the Keyword field is empty then below the field a message should be displayed, "Please enter a key word", as in Figure 3.
- 1.2. Further, if a user is deleting the content of a required field, and while deleting if the required field becomes fully empty, then the same message should be displayed, "Please enter a key word", as in Figure 3.

The screenshot shows a form field labeled "Key words: *". Below the field, a red error message "Please enter a key word" is displayed. The input field itself is empty.

Figure 3: Clear keyword input

Validation Set 2: Price Range restrictions

- 2.1. "from (\$)" input i.e., the first input field in the line after the "Price range:" label must always contain a valid number if specified. So whenever you specify anything apart from a number, the message "Price should be a valid decimal number" should be displayed as in Figure 4. Please note that the validation error message must be displayed as and when you enter an invalid input into the field and click somewhere on the webpage, irrespective of the submit button click.

The screenshot shows two input fields for a price range. The first field is labeled "Price range:" and contains the text "abc". The second field is labeled "to (\$)" and is empty. Below the first field, a red error message "Price should be a valid number" is displayed.

Figure 4: Price from not a valid number

- 2.2. "from (\$)" input i.e., the first input field in the line after "Price range:" label cannot have a value below zero. So whenever you specify a number lower than zero, the message "Minimum price cannot be below 0" should be displayed as in Figure 5. Please note that the validation error message must be displayed as and when you enter an invalid input into the field and click somewhere on the webpage, irrespective of the submit button click.

Price range:

to (\$)

Minimum price cannot be below 0

Figure 5: Price from below zero

Please note that based on validation 2.1 and 2.2, the values for “from (\$)” input can be positive integers or decimal numbers $[0, \infty)$.

2.3. “to (\$)” input i.e., the second input field in the line after “Price range:” label must always contain a valid number if specified. So whenever you specify anything apart from a number, the message “Price should be a valid number” should be displayed as in Figure 6. Please note that the validation error message must be displayed as and when you enter an invalid input into the field and click somewhere on the webpage, irrespective of the submit button click.

Price range:

abc

Price should be a valid number

Figure 6: Price to not a valid number

2.4. “to (\$)” input i.e., the second input field in the line after “Price range:” label cannot have a value less than the value in “from (\$)” input or below zero, if specified. So whenever you specify a number less than “from (\$)” input or zero, the message “Maximum price cannot be less than minimum price or below 0” should be displayed as in Figure 7. Please note that the validation error message must be displayed as and when you enter an invalid input into the field and click somewhere on the webpage, irrespective of the submit button click.

Price range:

0	-1
---	----

Maximum price cannot be less than minimum price or below 0

Figure 7: Maximum price cannot be less than minimum price or below zero

Validation Set 3: Max handling time restrictions

3.1. The “Max handling time(days)” input in the line after the “Shipping:” label must always contain a valid integer if specified. So whenever you specify anything apart from an integer, the message “Max handling time should be a valid digit” should be displayed as in Figure 8. Please note that the validation error message must be displayed as and when you enter an invalid input into the field and click somewhere on the webpage, irrespective of the submit button click.

Shipping: Free Shipping Expedited shipping

1.1

Max handling time should be a valid digit

Figure 8: Max handling time should be a valid digit

3.2. The “Max handling time(days)” input in the line after the “Shipping:” label must always contain an integer value greater than or equal to 1 if specified. So whenever you specify a integer value less than 1, the message “Max handling time should be greater than or equal to 1” should be displayed as in figure 9. Please note that the validation error message must be displayed as and when you enter an invalid input into the field and click somewhere on the webpage, irrespective of the submit button click.

Shipping: Free Shipping Expedited shipping

0

Max handling time should be greater than or equal to 1

Figure 9: Max handling time should be greater than or equal to 1

Please note that these are the **ALL AND ONLY** validations required on the search form for which your homework would be evaluated.

5. eBay API Usage

The same eBay App ID created for homework 6 can be used for homework 8 too. Also, the procedure of constructing URL for eBay “**findItemsAdvanced**” API Calls remains the same.

5.1. List of Parameters for API call

The mappings of the search form fields to the URL parameters and the additional parameters remain the same as listed in Section 3.2 of homework 6.

Additional Parameters to be used here are described below.

- **outputSelector[i]=SellerInfo**
Includes information about the seller in the response
- **outputSelector[i]=PictureURLSuperSize**
Include item's picture URL with size 800x800
- **outputSelector[i]=StoreInfo**
Includes information about the seller's eBay store in the response
- **paginationInput.pageNumber=<page number>**
Specifies which subset of data (or "page") to return in the call response.

Please note that `outputSelector[]` must use different indexes $i = 1, 2, 3, 4$ etc. for different options. For more information about filters, you can read this page:

<http://developer.ebay.com/DevZone/finding/CallRef/findItemsAdvanced.html>.

5.2. Parsing Xml Results

Figure 10 below is an example of the high level XML response of the `findItemsAdvanced` API when queried with all the parameters set as described above.

```
- <findItemsAdvancedResponse>
  <ack>Success</ack>
  <version>1.13.0</version>
  <timestamp>2015-02-14T01:32:48.621Z</timestamp>
  - <searchResult count="5">
    + <item></item>
    + <item></item>
    + <item></item>
    + <item></item>
    + <item></item>
  </searchResult>
  - <paginationOutput>
    <pageNumber>1</pageNumber>
    <entriesPerPage>5</entriesPerPage>
    <totalPages>20630</totalPages>
    <totalEntries>103146</totalEntries>
  </paginationOutput>
  - <itemSearchURL>
    http://www.ebay.com/sch/i.html?_nkw=harry+potter&_ddo=1&_ipg=5&_pgn=1
  </itemSearchURL>
</findItemsAdvancedResponse>
```

Figure 10: High-level successful response xml

In Figure 10, the item tags have been minimized. Expanding an item element gives us the following (Figure 11). All information specific to an item will be within its `<item>` element as shown above. The root `<item>` tag will be actually `findItemsAdvancedResponse->searchResult->item`.

```

▼<item>
  <itemId>281493533311</itemId>
  ▼<title>
    Apple iPhone 5 - 16GB - (Factory Unlocked) Smartphone - White or Black
  </title>
  <globalId>EBAY-US</globalId>
  <subtitle>Top Seller - Warranty - Fast Free Shipping</subtitle>
  ▼<primaryCategory>
    <categoryId>9355</categoryId>
    <categoryName>Cell Phones & Smartphones</categoryName>
  </primaryCategory>
  ▼<galleryURL>
    http://thumbs4.ebaystatic.com/pict/281493533311404000000002_1.jpg
  </galleryURL>
  ▼<viewItemURL>
    http://www.ebay.com/itm/Apple-iPhone-5-16GB-Factory-Unlocked-Smartphone-White-Black-/281493533311?pt=LH_DefaultDomain_0&var=580528005810
  </viewItemURL>
  <paymentMethod>PayPal</paymentMethod>
  <autoPay>true</autoPay>
  <location>USA</location>
  <country>US</country>
  ▼<storeInfo>
    <storeName>YouBuyRite</storeName>
    <storeURL>http://stores.ebay.com/YouBuyRite</storeURL>
  </storeInfo>
  ▼<sellerInfo>
    <sellerUserName>youbuyrite</sellerUserName>
    <feedbackScore>54743</feedbackScore>
    <positiveFeedbackPercent>98.6</positiveFeedbackPercent>
    <feedbackRatingStar>PurpleShooting</feedbackRatingStar>
    <topRatedSeller>false</topRatedSeller>
  </sellerInfo>
  ▼<shippingInfo>
    <shippingServiceCost currencyId="USD">0.0</shippingServiceCost>
    <shippingType>Free</shippingType>
    <shipToLocations>US</shipToLocations>
    <expeditedShipping>true</expeditedShipping>
    <oneDayShippingAvailable>true</oneDayShippingAvailable>
    <handlingTime>1</handlingTime>
  </shippingInfo>
  ▼<sellingStatus>
    <currentPrice currencyId="USD">224.88</currentPrice>
    <convertedCurrentPrice currencyId="USD">224.88</convertedCurrentPrice>
    <sellingState>Active</sellingState>
    <timeLeft>P2DT21H36M13S</timeLeft>
  </sellingStatus>
  ▼<listingInfo>
    <bestOfferEnabled>false</bestOfferEnabled>
    <buyItNowAvailable>false</buyItNowAvailable>
    <startTime>2014-11-10T21:31:59.000Z</startTime>
    <endTime>2015-03-10T21:36:59.000Z</endTime>
    <listingType>StoreInventory</listingType>
    <gift>false</gift>
  </listingInfo>
  <returnsAccepted>true</returnsAccepted>
  ▼<condition>
    <conditionId>2500</conditionId>
    <conditionDisplayName>Seller refurbished</conditionDisplayName>
  </condition>
  <isMultiVariationListing>true</isMultiVariationListing>
  ▼<pictureURLSuperSize>
    http://i.ebayimg.com/00/s/OTAwWDkwMA==/z/2hUAAOSwy4hUT~90/$_3.JPG
  </pictureURLSuperSize>
  <topRatedListing>false</topRatedListing>
</item>

```

Figure 11: <item> Detailed

The result tags from which you should extract information for display are given in Table 1. Your PHP program should parse the resulting XML and create a JSON which must be further used for result display.

XML Response	JSON output	Result Display
<i>ack</i>	"ack"	To be used to check success or failure case
<i>paginationOutput->totalEntries</i>	"resultCount"	To display the total number of entries in results header

<i>paginationOutput->pageNumber</i>	"pageNumber"	To keep track of the current page number
<i>paginationOutput->entriesPerPage</i>	"itemCount"	To keep track of the number of entries being displayed.
<i>searchResult->item->title</i>	"item"->"basicInfo"->"title"	Item title for the item header
<i>searchResult->item->viewItemURL</i>	"item"->"basicInfo"->"viewItemURL"	Item URL for the associated eBay webpage
<i>searchResult->item->galleryURL</i>	"item"->"basicInfo"->"galleryURL"	Item image URL to be used for a thumbnail image displayed on the left side of the item header.
<i>searchResult->item->pictureURLSuperSize</i>	"item"->"basicInfo"->"pictureURLSuperSize"	Item super size image URL to be used for an item modal dialog
<i>searchResult->item->sellingStatus->convertedCurrentPrice</i>	"item"->"basicInfo"->"convertedCurrentPrice"	Item price in dollars
<i>searchResult-->item->shippingInfo->shippingServiceCost</i>	"item"->"basicInfo"->"shippingServiceCost"	If the value of the field is equal to 0.0 or empty, it's considered FREE Shipping.
<i>searchResult->item->condition->conditionDisplayName</i>	"item"->"basicInfo"->"conditionDisplayName"	Used in Results display Basic Info tab for Condition field
<i>searchResult->item->listingInfo->listingType</i>	"item"->"basicInfo"->"listingType"	Used in Results display Basic Info tab for Buying Format field. If the value of field is "FixedPrice" or "StoreInventory", print "Buy It Now". If the value is "Auction" print "Auction". If the value is "Classified" print "Classified Ad. Else display the value as is.
<i>searchResult->item->location</i>	"item"->"basicInfo"->"location"	Item location field used in the item header after

		price.
<code>searchResult->item->primaryCategory->categoryName</code>	<code>"item"->"basicInfo"->"categoryName"</code>	Used in Results display Basic Info tab for Category name field
<code>searchResult-->item->topRatedListing</code>	<code>"item"->"basicInfo"->"topRatedListing"</code>	If the value of <i>field</i> is true, display the image itemTopRated.jpg after item location in item header
<code>searchResult-->item->sellerInfo->sellerUserName</code>	<code>"item"->" sellerInfo "->"sellerUserName"</code>	Used in Results display Seller Info tab for User name field
<code>searchResult-->item->sellerInfo->feedbackScore</code>	<code>"item"->" sellerInfo "->"feedbackScore"</code>	Used in Results display Seller Info tab for Feedback score field
<code>searchResult-->item->sellerInfo->positiveFeedbackPercent</code>	<code>"item"->" sellerInfo "->"positiveFeedbackPercent"</code>	Used in Results display Seller Info tab for Positive feedback field
<code>searchResult-->item->sellerInfo->feedbackRatingStar</code>	<code>"item"->" sellerInfo "->"feedbackRatingStar"</code>	Used in Results display Seller Info tab for Feedback rating field
<code>searchResult-->item->sellerInfo->topRatedSeller</code>	<code>"item"->" sellerInfo "->"topRatedSeller"</code>	Used in Results display Seller Info tab for Top rated field
<code>searchResult-->item->storeInfo-> storeName</code>	<code>"item"->" sellerInfo "->"sellerStoreName"</code>	Used in Results display Seller Info tab for Store field
<code>searchResult-->item->storeInfo-> storeURL</code>	<code>"item"->" sellerInfo "->"sellerStoreURL"</code>	Used in Results display Seller Info tab for Store field as a hyperlink
<code>searchResult->item->shippingInfo->shippingType</code>	<code>"item"->" shippingInfo "->"shippingType"</code>	Used in Results display Shipping Info tab for Shipping type field. It must be space separated.
<code>searchResult->item->shippingInfo->shipToLocations</code>	<code>"item"->" shippingInfo "->"shipToLocations"</code> Please note that there might be several shipToLocations tags inside <code>shippingInfo</code> tag. All	Used in Results display Shipping Info tab for Shipping locations field.

	the results comma separated must be returned.	
<i>searchResult->item->shippingInfo->expeditedShipping</i>	"item"->" shippingInfo "->"expeditedShipping"	Used in Results display Shipping Info tab for Expedited shipping field.
<i>searchResult->item->shippingInfo->oneDayShippingAvailable</i>	"item"->" shippingInfo "->"oneDayShippingAvailable"	Used in Results display Shipping Info tab for One day shipping field.
<i>searchResult->item->returnsAccepted</i>	"item"->" shippingInfo "->"returnsAccepted"	Used in Results display Shipping Info tab for Returns accepted field.
<i>searchResult->item->shippingInfo->handlingTime</i>	"item"->" shippingInfo "->"handlingTime"	Used in Results display Shipping Info tab for Handling time field.

Table 1: Mapping between XML Tags, JSON keys and Result Display

JSON output format

Your JSON output from the PHP file should have all the necessary data and links to generate the front-end information.

JSON should be formatted as below:

```
{
    "ack": "Success",
    "resultCount": "108331",
    "pageNumber": "1",
    "itemCount": "5",
    "item0": {
        "basicInfo": {
            "title": "HOT New Harry Potter 14.5\" Magical Wand Replica Cosplay In Box",
            "viewItemURL": "http://www.ebay.com/itm/HOT-New-Harry-Potter-14-5-Magical-Wand-Replica-Cosplay-Box-/331128976689?pt=LH_DefaultDomain_0",
            "galleryURL": "http://thumbs2.ebaystatic.com/m/mICawRUsvUULmXv2hqI7Y3A/140.jpg",
            "pictureURLSuperSize": "http://i.ebayimg.com/00/s/MTYwMFgxNjAw/z/c08AA0xyOypSZVqH/$T2eC16NHJGYFFkn,pk3TBSZVqFH!pg~~~60_3.JPG",
            "convertedCurrentPrice": "11.98",
            "shippingServiceCost": "1.0",
            "conditionDisplayName": "New",
            "listingType": "StoreInventory",
        }
    }
}
```

```

    "location": "Hong Kong",
    "categoryName": "Harry Potter",
    "topRatedListing": "false"
},
"sellerInfo": {
    "sellerUserName": "easygoing182",
    "feedbackScore": "4978",
    "positiveFeedbackPercent": "99.2",
    "feedbackRatingStar": "Red",
    "topRatedSeller": "false",
    "sellerStoreName": "easygoing182",
    "sellerStoreURL": "http://stores.ebay.com/easygoing182"
},
"shippingInfo": {
    "shippingType": "Flat",
    "shipToLocations": "Worldwide",
    "expeditedShipping": "false",
    "oneDayShippingAvailable": "false",
    "returnsAccepted": "true",
    "handlingTime": "1"
}
},
"item1": {
    <similar to item0 above>
},
"item2": {
    <similar to item0 above>
},
"item3": {
    <similar to item0 above>
},
"item4": {
    <similar to item0 above>
}
}

```

Case: No results found

The case where no results are found is shown in Figure 12. You can then query the value of *paginationOutput->totalEntries*. If the value of *totalEntries* is zero, display an appropriate message in the results area as shown in section 6.2.

```

- <findItemsAdvancedResponse>
  <ack>Success</ack>
  <version>1.13.0</version>
  <timestamp>2015-02-14T02:23:07.293Z</timestamp>
  <searchResult count="0"/>
- <paginationOutput>
  <pageNumber>0</pageNumber>
  <entriesPerPage>5</entriesPerPage>
  <totalPages>0</totalPages>
  <totalEntries>0</totalEntries>
</paginationOutput>
- <itemSearchURL>
  http://www.ebay.com/sch/i.html?_nkW=sodcnkisbdejkshbdcjkr&_ddo=1&_ipg=5&_pgn=1
</itemSearchURL>
</findItemsAdvancedResponse>

```

Figure 12: No Results XML

In this case a JSON message is returned as following:

```
{"ack": "No results found"}
```

6. Result Display

6.1. Success Case

The list of results should be displayed as shown in Figure 13. On the top you need to mention the range of items from total number of items.

The list of items is to be vertically stacked. You must use Bootstrap Media Object for the same. To the left is the item's display image. On the right, item title hyperlinked with the item's product page on eBay is displayed. Below it, the price of the item (+ \$x for shipping, if applicable) in bold, followed by the location the item ships from in *italics*, top-rated image (if the item is top-rated), "View Details" link and a Facebook post button is displayed. The image *itemTopRated.jpg* can be used from <http://cs-server.usc.edu:45678/hw/hw8/itemTopRated.jpg>.

26-30 items out of 3970738



ATT APPLE IPHONE 4 WHITE 16GB UNLOCKED

Price: \$83.75 (FREE Shipping) Location: Rainbow City, AL, USA [View Details](#)



New Apple iPhone 4S 16GB White Factory Unlocked GSM

Price: \$278.98 (FREE Shipping) Location: USA [View Details](#)



Apple iPhone 5 16GB White & Silver AT&T Smartphone Clean ESN

Price: \$132.5 (FREE Shipping) Location: Brooklyn, NY, USA [View Details](#)



Apple iPhone 4 32GB WiFi Verizon Wireless Black Smartphone Clean Esn

Price: \$78.55 (FREE Shipping) Location: Jackson, NJ, USA [View Details](#)



Apple iPhone 5 - 32GB - Black & Slate (AT&T) Smartphone Jailbroken iOS 6.02

Price: \$156.5 (FREE Shipping) Location: Merchantville, NJ, USA [View Details](#)

« [6](#) [7](#) [8](#) [9](#) [10](#) »

Figure 13: Result Display

On click of the “View Details” link for an item, the details for the item must expand as in Figure 14. You must use **Bootstrap Collapse** for the same.



Factory Unlocked Apple iPhone 5 32GB iOS Smartphone 4G GSM Retina White / Black

Price: \$259.99 (FREE Shipping) Location: USA [View Details](#)

[Basic Info](#)

[Seller Info](#)

[Shipping Info](#)

Category name

Cell Phones & Smartphones

Condition

Manufacturer refurbished

Buying format

Buy It Now

Figure 14: Item details expanded on click of View Details

In the expanded area, 3 tabs must be displayed: Basic Info, Seller Info and Shipping Info. You must use Bootstrap Tab for the same.

Basic Info Tab:

As shown in Figure 14, it must display Category Name, Condition and Buying Format. The same can be obtained from “basicInfo” under “itemx” key of JSON for item x.

Seller Info Tab:

It must display information as shown in Figure 15. The information can be obtained from the “sellerInfo” under “itemx” key of JSON for item x. You must use Bootstrap Glyphicons for cross mark and correct icons used for Top rated. Also the store name must be hyperlinked to store url, which on click must open in a new tab. Incase store is empty, you must not hyperlink it.

STRAIGHT TALK (Verizon) Apple iPhone 4 16GB Black Smartphone
FAST SHIP FROM NY

Price: \$79.99 (FREE Shipping) Location: Brooklyn,NY,USA [View Details](#) [f](#)

Basic Info	Seller Info	Shipping Info
User name	emb-phones	
Feedback score	3789	
Positive feedback	99.9%	
Feedback rating	Red	
Top rated	✓	
Store	emb-phones	

Apple iPhone 4s - 8GB - White (straight talk) Verizon towers

Price: \$105 (FREE Shipping) Location: Rutherford,VA,USA [View Details](#) [f](#)

Basic Info	Seller Info	Shipping Info
User name	matrixx2011	
Feedback score	3	
Positive feedback	100.0%	
Feedback rating	None	
Top rated	✗	
Store	N/A	

Figure 15. Seller Info tab

Shipping Info Tab:

It must display information as shown in Figure 16. The information can be obtained from the “shippingInfo” under “itemx” key of JSON for item x. You must similarly use Bootstrap Glyphicons for cross mark and correct icons as needed. Please note that the Shipping type information must be space separated and not continuous as one string.

 STRAIGHT TALK (Verizon) Apple iPhone 4 16GB Black Smartphone
FAST SHIP FROM NY

Price: \$79.99 (FREE Shipping) Location: Brooklyn, NY, USA  View Details 

Basic Info Seller Info **Shipping Info**

Shipping type	Free
Handling time	1 day(s)
Shipping locations	US
Expedited shipping	✓
One day shipping	✗
Returns accepted	✓

 Apple iPhone 4s - 8GB - White (straight talk) Verizon towers

Price: \$105 (FREE Shipping) Location: Rutherford, NJ, USA  View Details 

Basic Info Seller Info **Shipping Info**

Shipping type	Free
Handling time	2 day(s)
Shipping locations	US, CA, GB, AU, AT, BE, FR, DE, IT, JP, ES, CH, NL, CN, HK, MX
Expedited shipping	✓
One day shipping	✗
Returns accepted	✗

Figure 16. Shipping Info tab

Image Modal:

On click of the Image corresponding to the item, a modal dialog must be displayed with the Plus size picture of the item in the "basicInfo" key under "itemx" as in Figure 17. You must use Bootstrap Modal for the same. The image must be centered



Figure 17. Item image display

Facebook post:

On click of the Facebook post button a Facebook post must be done with the format shown in figure 18.

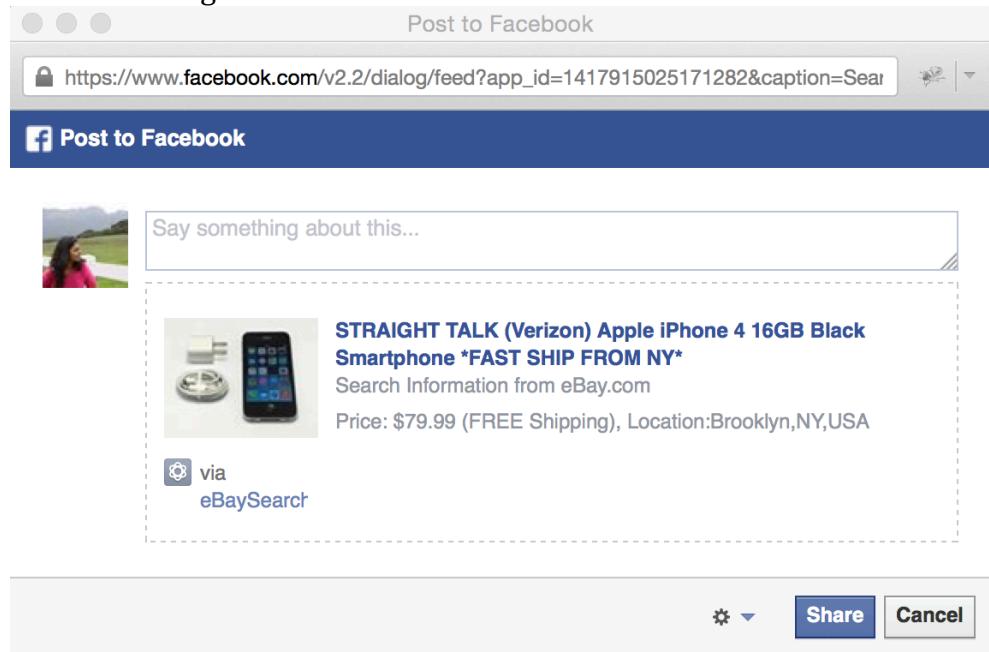


Figure 18: Post to Facebook

Display the appropriate values for item title, item price with shipping cost associated, item location, image (galleryURL). The link on the post should directly link to the eBay item page (viewItemURL) for the item details. Image for button can be used from <http://cs-server.usc.edu:45678/hw/hw8/fb.png>.

When the button is pressed, the web application does the following:

- Authorizes the user to Facebook (i.e. logs him/her in) using the application and user credentials if the user is not already logged in to Facebook;

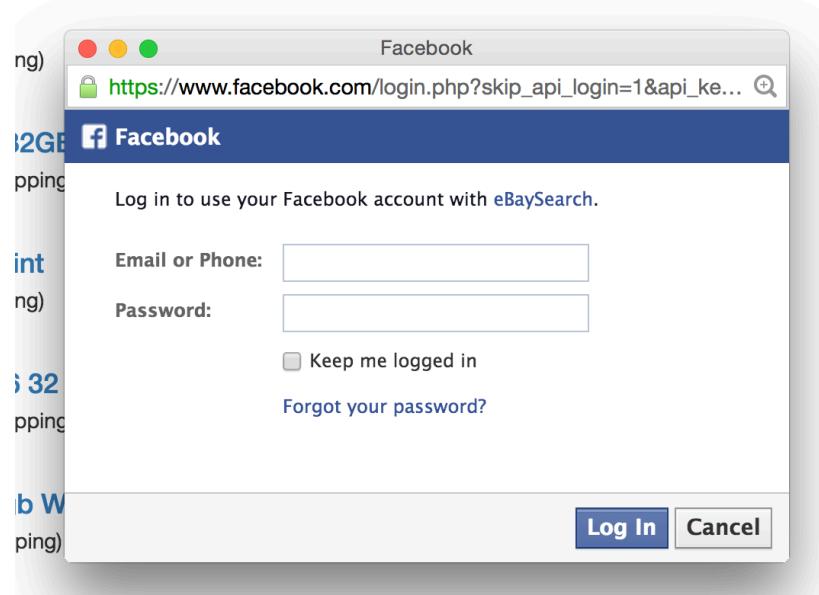


Figure 19: Facebook Login

- Posts an Update Status message to the feed.
- The above two steps can be performed using the Facebook Connect API, using the JavaScript SDK, which provides a rich set of client-side functionality for accessing Facebook's server-side API calls. It is documented at:
<https://developers.facebook.com/docs/reference/javascript/>

Once the post has been published, you should show an alert box informing the user of whether the post has been published or not. For example:

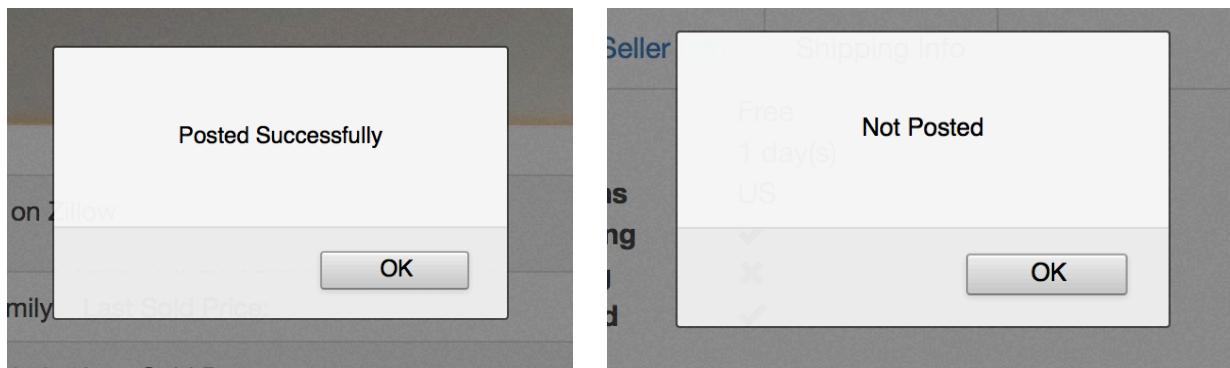


Figure 20: Post status Alert box

Pagination:

Below the list of results a pagination bar must be displayed as shown in Figure 21.

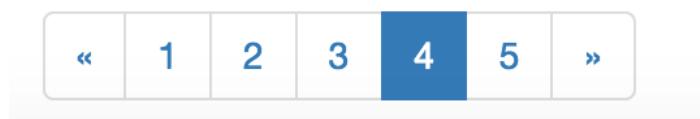


Figure 21: Pagination Bar

On click of a number index in the pagination bar, an asynchronous AJAX request (as before) must be generated with the `paginationInput.pageNumber` option of the eBay API url set to the same. As here, `paginationInput.pageNumber=4` must be added in the API url.

On the click of << box in the pagination bar, an asynchronous AJAX request (as before) must be generated for page number one less than current page. Here, in the example on the click of the << box, results for page number 3 must be displayed eventually. Note that if the current page is page number 1, then << box must be disabled.

On the click of >> box in the pagination bar, an asynchronous AJAX request (as before) must be generated for page number one greater than current page. If the current page is multiple of 5, for example 5, then the results for page number 6 must be displayed and the Pagination Bar must be updated as in Figure 22.



Figure 22: Updated Pagination Bar

The **page number** for which the results are being displayed must be shown as an **active box** on the pagination bar.

Note: If any of the field is unavailable, you should **show “N/A” instead.**

Result Display Responsive to Devices

You are supposed to make result display responsive to devices. If the page is loading on a smart phone or a tablet, the display should be according to the width of the devices.

Make the Item results responsive to Mobile Devices

You are supposed to make the item results list responsive to mobile devices. If the page is loading on a smart phone or a tablet, the information should be adjusted accordingly; one of the examples is shown in the figure 23 below.

1-5 items out of 4062887



att iphone 5s 16gb
Price: \$280
 (FREE Shipping) Location:
 Carrollton, TX, USA [View Details](#) 



**Apple iPhone 5s - 32GB
(Factory Unlocked)
Smartphone - Gray or Silver**
Price: \$329.88
 (FREE Shipping) Location:
 USA [View Details](#) 



iPhone 5 16GB Sprint
Price: \$137
 (FREE Shipping) Location: Blue
 Springs, MO, USA [View Details](#) 

1-5 items out of 4062887



att iphone 5s 16gb
Price: \$280
 (FREE Shipping) Location:
 Carrollton, TX, USA [View Details](#) 

Basic Info
Seller Info

Shipping Info

Category name
 Cell Phones & Smartphones
Condition
 Used
Buying format
 Auction



**Apple iPhone 5s - 32GB
(Factory Unlocked)
Smartphone - Gray or Silver**
Price: \$329.88
 (FREE Shipping) Location:
 USA [View Details](#) 

Figure 23: Result display on mobile

21

Make the Image Modal responsive to Mobile Devices

You are supposed to make the image modal displayed on click of the image responsive to mobile devices. If the page is loading on a smart phone or a tablet, the image should be displayed fully on the device, rather than being displayed partially. An example is shown in Figure 24 below.



Figure 24: Responsive image modal display on iPhone

6.2. Failure Case

Similar to HW6, in case where the eBay API returns no items for a specific keyword such as “*alcnkajcsnkacnka*”, you should display “No results found” as shown in Figure 25.

The screenshot shows the eBay Shopping search interface. The search bar contains the placeholder text "alcnkajcsnkacnka". Below the search bar are several filter options: Price range (from \$ to \$), Condition (New, Used, Very Good, Good, Acceptable), Buying formats (Buy It Now, Auction, Classified Ads), Seller (Return accepted), Shipping (Free Shipping, Expedited shipping), Sort by (Best Match), and Results per page (5). At the bottom are "Clear" and "Search" buttons.

No results found

Figure 25: Example Output for an eBay search, which returns No Information

7. Implementation Hints

- Get started with Bootstrap Library
To know how to get started with Bootstrap, please refer to the link <http://getbootstrap.com/getting-started/>. You need to import the necessary CSS file and JS file provided by Bootstrap.
- jQuery Validation
You are required to use jQuery validation together with Bootstrap to validate user input. Please refer to the links below for more details about the implementation of this functionality.
<http://jqueryvalidation.org/>
<http://getbootstrap.com/css/#forms-control-validation>
- Bootstrap UI Components
Bootstrap provides a complete mechanism to make Web page responsive to different mobile devices. In this exercise you will get hands-on experience with responsive design using the Bootstrap Grid System. You will need to use Bootstrap Form, Media object, Tab, Modal, Collapse and Glyphicons to implement all the functionalities.
Bootstrap Form <http://getbootstrap.com/css/#forms>

Bootstrap Media object <http://getbootstrap.com/components/#media>
Bootstrap Tabs <http://getbootstrap.com/javascript/#tabs>
Bootstrap Modal <http://getbootstrap.com/javascript/#modals>
Bootstrap Collapse <http://getbootstrap.com/javascript/#collapse>
Bootstrap Glyphicons <http://getbootstrap.com/components/#glyphicons>

- **AWS**

You should use the domain name of the AWS server you created in HW#7 to make the request. For example, if your AWS server domain is called default-environment-randomstring.elasticbeanstalk.com, and the user performs a GET request with parameter name="John Smith", then a query of the following type will be generated:

`http://default-environment-randomstring.elasticbeanstalk.com/?name= John%20
Smith`

- **AJAX call**

You can send the request to the PHP file by passing the URL to `$.ajax()`. You must use a GET method to request the resource since you are required to provide this link to your homework list to let graders check whether the PHP code is running on Amazon AWS. (Please refer to the grading guideline for details).

The AJAX call:

```
$.ajax({
    url: 'URL you created in HW#7',
    // this is the parameter list
    data: { name: "John Smith",
            age: "23"
          },
    type: 'GET',
    success: function(output) {
        // parse the data here
      },
    error: function(){
      }
});
```

Files to Submit

On your course homework page, you should update the HW8 link to refer your new initial web page. Also, Submit your files electronically to the csci571 account so that they can be graded and compared to all other students' code via the MOSS code comparison tool.

IMPORTANT:

All discussion and explanation in Piazza related to this homework become part of the homework description. So please review all Piazza threads before finishing the assignment.

In your Facebook application settings, you should go to the “Status & Review” section and choose “Yes” for the question “***Do you want to make this app and all its live features available to the general public?***” as shown in the Figure 26. If you answer “YES”, anyone will be able to log in through her/his Facebook account and post via your web interface. Otherwise, the developer will be the only person who is able to use the Facebook functionality in the web page. If graders try to test the functionality of Facebook button and they are not able to log in you will lose **all** points related to the Facebook component in your homework grade.

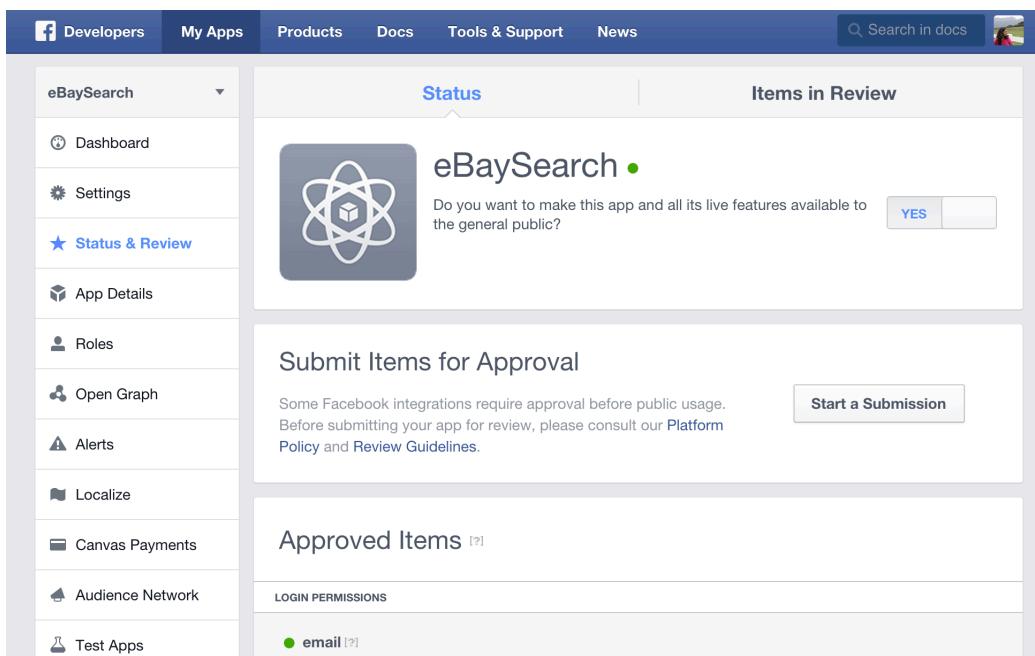


Figure 26 Status & Review of Facebook Application