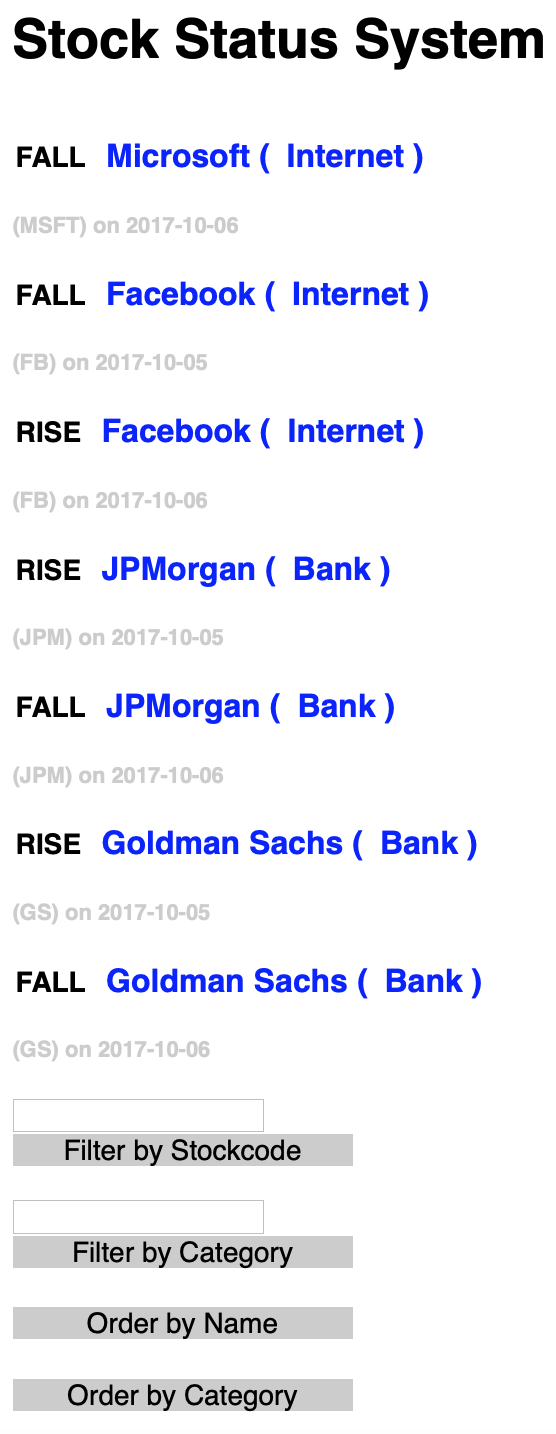
**COMP3322B MODERN TECHNOLOGIES ON WORLD WIDE WEB**

**Lab 4 jQuery and JSON**

**Overview**

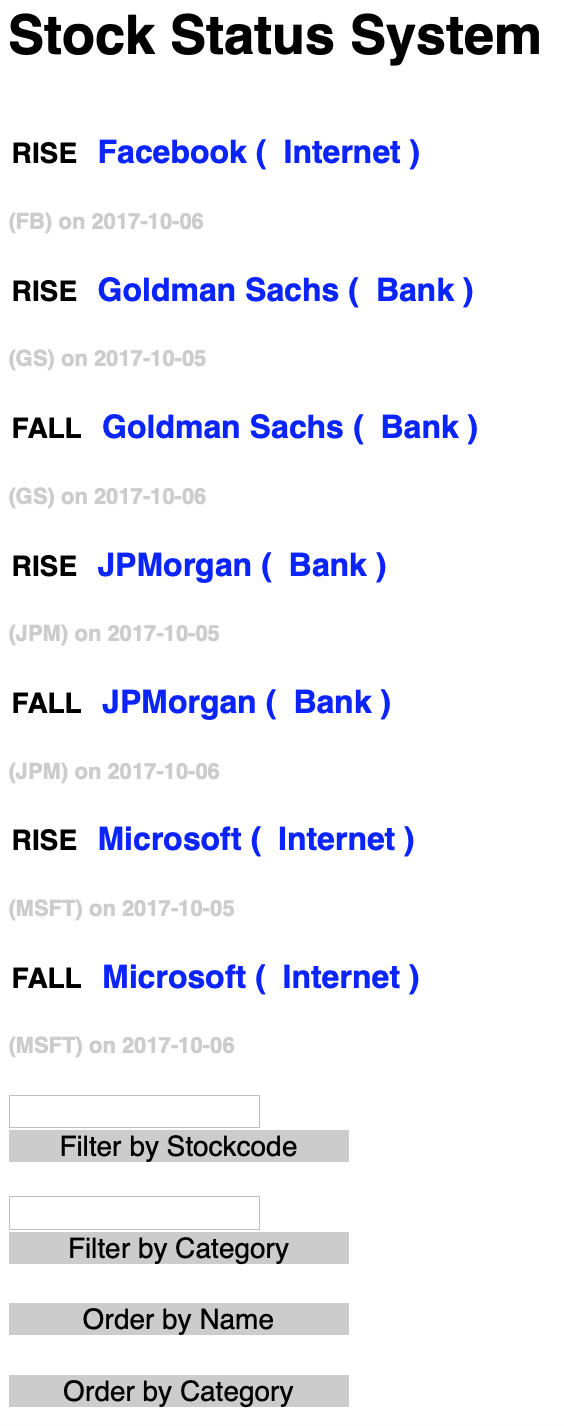
In this lab, we will implement the same web-based stock status system, which we implemented in Lab 3, using jQuery, plus achieving a few additional functionalities. Especially, we add two buttons “Order by Name” and “Order by Category” on the page. When each button is clicked, the stock records displayed will be ordered alphabetically by stock name or stock category, respectively.

Example screenshot of the page after initial loading is given in Fig. 1.



**Fig. 1**

Example screenshots of the page after clicking the “Order by Name” button and after clicking the “Order by Category” button, are given in Figures 2-4. Note that the displayed entries, no matter the full list or filtered list, will be ordered accordingly when one of the two buttons are clicked.

** **

**Fig. 2 Order by Name Fig. 3 Order by Category**

****

**Fig. 4 Order by name after filtering by category**

**Lab Exercise 1 – re-implement Lab 3 functionalities using jQuery and JSON**

**Step1:** We will use the same database which you created and used in Lab 3. Make sure there are at least 8 records in the database table.

**Step2:** Download **lab4\_materials.zip** and unzip it to a folder “**lab4**”. You will find all the 6 files we need in this lab. You will add code in **index.html** and **script.js**, and do not need to revise the other files (except for adding your database login credentials in the two **php** files).

**Step3:** In **index.html**, load the jQuery library by adding the following line (jquery-3.3.1.js has been provided in “**lab4**” folder):

|  |
| --- |
| <script src=" jquery-3.3.1.js"></script> |

In addition, add code in **index.html** to link to **script.js**, in which you will implement all the jQuery code.

**Step4:** In **script.js**, we have provided the following code:

|  |
| --- |
| $(document).ready(function () {  showAll();    $("#button\_all").click(showAll);    $("#FS").click(function () {      });  $("#FC").click(function () {      });  $("#orderByName").click(function () { //order by name      });  $("#orderByCategory").click(function () { //order by category      });  });  function showAll() {      }  function changeState(elem) {      } |

(1) Implement the function **showAll()** using jQuery function $.get (to issue an HTTP GET AJAX request for **queryEntries.php**) and $("#button\_all").hide(); (for removing the “Show All” button from the page), to achieve the same functionality as the showAll() function in Lab 3.

Different from Lab 3, in **queryEntries.php** that we have provided in the lab4 folder, the response data produced is a JSON string (produced from a JSON array of associative arrays), instead of the HTML content as in Lab 3. In **showAll()**, you should parse the received JSON string into a JSON object, retrieve its content and produce the HTML code for stock entry display, in the same format as the HTML content produced in Lab 3. Use $("#entries").html(…); to load the HTML content as innerHTML content of the division of id "entries".

(2)Complete the callback function in $("#FS").click(function () { }); and $("#FC").click(function () { }); to achieve the same functionalities as in **filterS()** and **filterC()** in Lab 3. Use jQuery functions. Again, note that **queryEntries.php** returns a JSON struct and you should construct the HTML content accordingly on the client side.

(3) Implement the function **changeState()** using jQuery method load() to issue an HTTP POST AJAX request for **updateState.php**, to achieve the same functionality as changeState()in Lab 3 (except for sending a POST request instead of a GET request). Note that updateState.php provided in “lab4” is slightly different from updateState.php in Lab 3, by handling POST requests instead of GET requests.

**Lab Exercise 2 – order the displayed entries**

In **script.js**, we also have two event handlers for “click” on the two additional buttons for ordering the entries shown on the page: $("#orderByName").click(function () { }); and $("#orderByCategory").click(function () { });.

**Step 1**. Implement the event handling function in $("#orderByName").click(function () { }); as follows:

|  |
| --- |
| $entrydivs=$("#entries").children();    $entrydivs.sort(function(a,b){  var an = $($(a).find('h3')[0]).text();  var bn = $($(b).find('h3')[0]).text();  if(an > bn) {  return 1;  }  if(an < bn) {  return -1;  }  return 0;  });  $entrydivs.detach().appendTo($("#entries")); |

Understand the above code for sorting the set of child <div> elements in the division with id “entries”, by referring to the following references:

* JavaScript sort() method <https://www.w3schools.com/jsref/jsref_sort.asp>
* jQuery .detach() method <https://api.jquery.com/detach/>
* jQuery .appendTo() method <http://api.jquery.com/appendto/>

**Step 2**. Implement the event handling function in $("#orderByCategory").click(function () { }); by learning from the code in Step 1.

**Submission**

Please finish this lab exercise before 23:59 Sunday March 17, 2019 (no delayed submission of this lab is acceptable). Please upload all files (**script.js, index.html, queryEntries.php, updateState.php, jquery-3.3.1.js, style.css**) to **i.cs.hku.hk** web server under “**lab4**”, similar to what you did in previous labs. The URL to access your page should be **http://i.cs.hku.hk/~[your\_CSID]/lab4/index.html**. We will check the page for your lab 4 marking. Make sure you check the page after uploading it to the server.