Homework 6: Server-side Scripting

1. Objectives

- Get experience with the PHP programming language;
- Get experience with the Alpha Vantage real time stock quote API, HighCharts graphics and Seeking Alpha News RSS feed;
- Get experience using XML and JSON parsers in PHP.

2. Description

In this exercise, you are asked to create a webpage that allows you to search for stock information using the *Stock quote API*, and the results will be displayed in both tabular format and charts format using HighCharts. You will also provide News clips for the selected stock.

2.1. Description of the Search Form

A user first opens a page, called **stock.php (or any valid web page name)**, where he/she can enter a stock ticker symbol. An example is shown in Figure 1. Providing a value for the "Stock Ticker Symbol" field is mandatory.

Figure 1: Initial Search Screen

The search form has two buttons:

- **Search** button: An example of valid input is shown in Figure 2. Once the user has provided valid data, your client script should send a request to your web server script **stock.php** with the form data. You can use either GET or POST to transfer the form data to the web server script. The PHP script will retrieve the from data, reformat it to the syntax of the API and send it to the Alpha Vantage restful web service. If the user clicks

on the search button without providing a value in the “Enter Stock Ticker Symbol” field, there should be an alert window that pops up (an example is shown in Figure 3).

- **Clear** button: This button must clear the result area (below the search area) and the text field (stock ticker symbol). The Clear operation must be done using a JavaScript function.

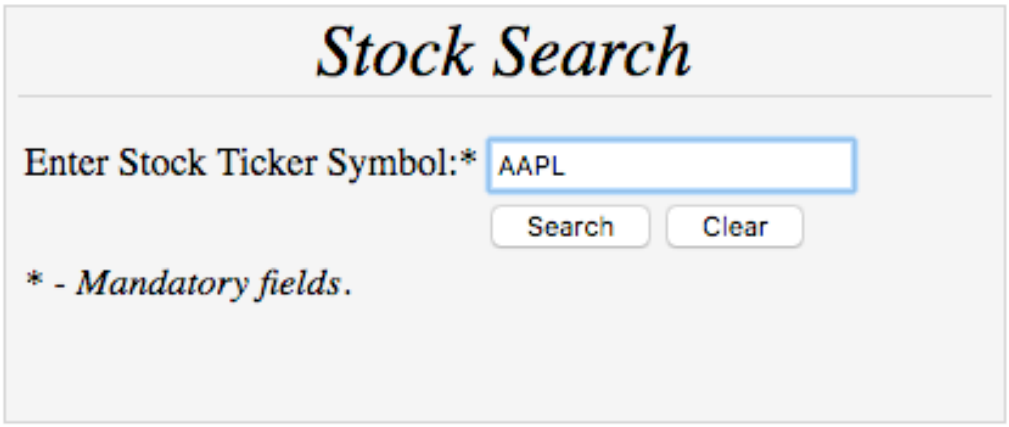

Figure 2: An Example of Valid Input

Figure 3: An Example of Empty Input

2.2 Displaying Results

In this section, we outline how to use the form data to construct the calls to RESTful web services of the Alpha Vantage APIs, Seeking Alpha News and display the results in the web page. There are basically **three** components to display "vertically" from top to bottom of the web page: stock information table, stock price/volume/indicator chart and stock news. A sample result is shown in Figure 4.

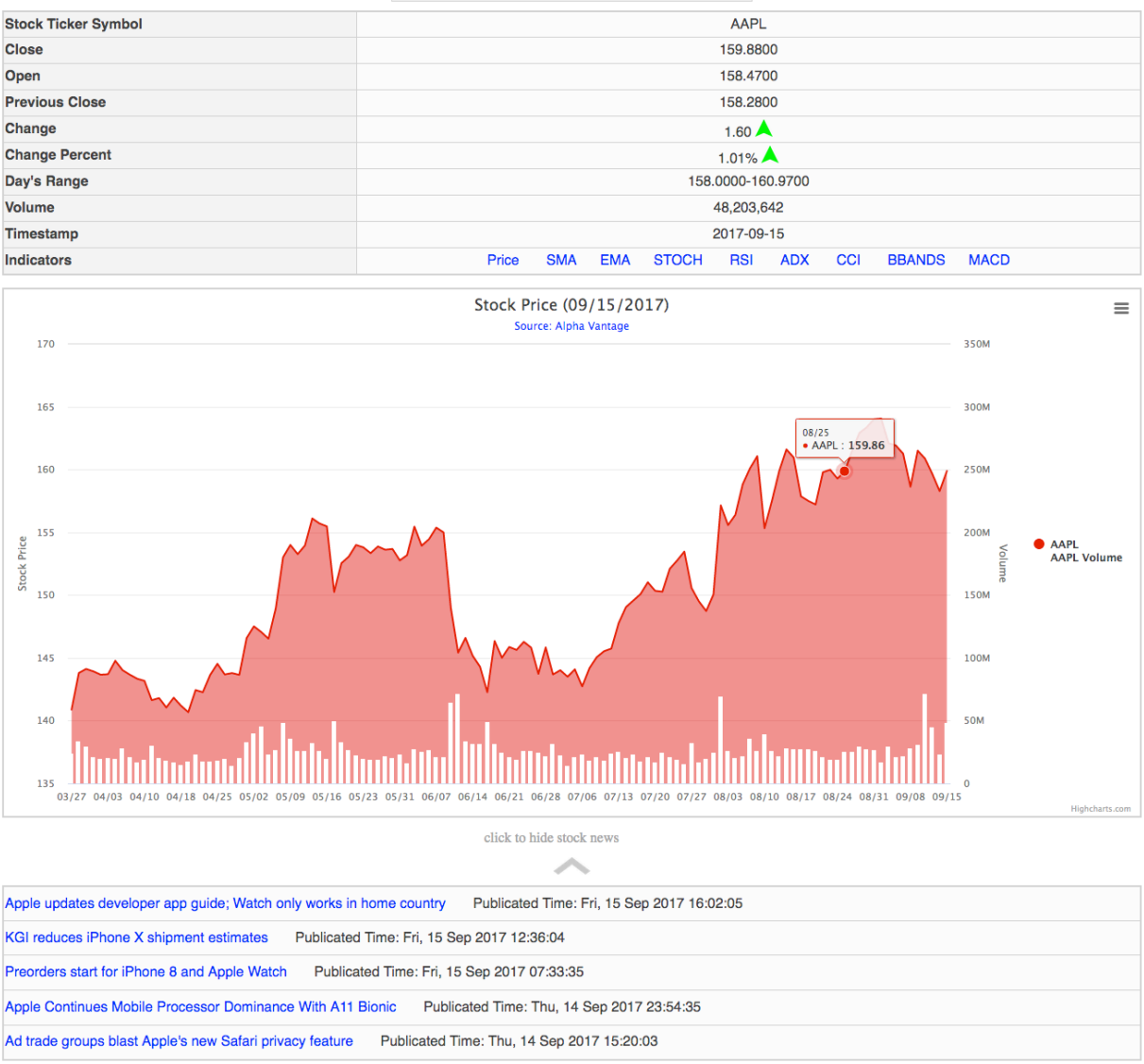

Figure 4: A Sample Search Result for Apple Inc(AAPL)

2.2.1 Displaying Stock Information Table

From the Alpha Vantage APIs, we use two kinds of web services: Daily Time Series Data and Technical Indicators. The Daily Time Series Data web service returns a JSON file which contains the historical stock quote of the company.

The PHP script (i.e., **stock.php**) uses the input information (stock symbol or "name of equity") to construct a restful web service URL to retrieve all companies matching the query input:

https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=MSFT&apikey=demo

The Daily Time Series Data web service URL has a parameter called symbol. The value of the symbol parameter should be the text entered in the "Stock Ticker Symbol" edit box. The above link is an example of calling lookup service with the input "MSFT". The response of this URL is a JSON object. Figure 5 shows an example of the returned response of the lookup service.


```
{
    "Meta Data": {
        "1. Information": "Daily Prices (open, high, low, close) and Volumes",
        "2. Symbol": "MSFT",
        "3. Last Refreshed": "2017-09-15",
        "4. Output Size": "Compact",
        "5. Time Zone": "US/Eastern"
    },
    "Time Series (Daily)": {
        "2017-09-15": {
            "1. open": "74.8300",
            "2. high": "75.3900",
            "3. low": "74.0700",
            "4. close": "75.3100",
            "5. volume": "37901927"
        },
        "2017-09-14": {
            "1. open": "75.0000",
            "2. high": "75.4900",
            "3. low": "74.5200",
            "4. close": "74.7700",
            "5. volume": "15373384"
        },
        "2017-09-13": {
```


Figure 5: A Sample Result of the Daily Time Series Data service

The PHP script (i.e., **stock.php**) should parse the returned JSON object and extract the necessary fields. After extracting the data, the PHP script should display the data in a tabular and chart format below the search form. A sample output is shown in Figure 6.

Stock Ticker Symbol	AAPL
Close	159.8800
Open	158.4700
Previous Close	158.2800
Change	1.60 ▲
Change Percent	1.01% ▲
Day's Range	158.0000-160.9700
Volume	48,203,642
Timestamp	2017-09-15
Indicators	Price SMA EMA STOCH RSI ADX CCI BBANDS MACD

Figure 6: An Example of Stock Information Table

The table contains these fields: Stock Ticker Symbol, Close, Open, Previous Close, Change, Change Percent, Day's Range, Volume, Timestamp and Indicators. Indicators includes Price (the default), SMA, EMA, STOCH, RSI, ADX, CCI, BBANDS and MACD (see Hints about details of each field).

To request the Indicators, use the following URL to invoke the API. Like the Series Data API, the Technical Indicator service URL also has a "symbol" parameter, whose value is the same as the one used for retrieving a stock quote. For example, the following RESTful call is invoking the Technical Indicators service API, to retrieve the Simple Moving Average (SMA) for the symbol "MSFT" (the stock symbol of Microsoft). The response of this API call is a JSON object. Figure 7 shows an example of the returned response of the SMA API nrequest:

https://www.alphavantage.co/query?function=SMA&symbol=MSFT&interval=weekly&time_period=10&series_type=open&apikey=demo


```
{
    "Meta Data": {
        "1: Symbol": "MSFT",
        "2: Indicator": "Simple Moving Average (SMA)",
        "3: Last Refreshed": "2017-09-15",
        "4: Interval": "weekly",
        "5: Time Period": 10,
        "6: Series Type": "open",
        "7: Time Zone": "US/Eastern"
    },
    "Technical Analysis: SMA": {
        "2017-09-15": {
            "SMA": "72.8130"
        },
        "2017-09-08": {
            "SMA": "72.3150"
        },
        "2017-09-01": {
            "SMA": "72.1210"
```


Figure 7: An Example of returned JSON for Indicator Request

Note that you can use JavaScript to request and parse a JSON file for indicators (i.e., SMA, EMA, STOCH, RSI, ADX, CCI, BBANDS and MACD). But for stock price/volume (JSON file) and stock news (XML file), you must use PHP to request and parse these JSON and XML objects.

2.2.2 Displaying Stock Price/Volume/Indicator Chart

The PHP script (i.e., **stock.php**) should extract the content of **Time Series Data** from the returned JSON object to construct a chart which is responsible for displaying (close) price/ volume and indicators. The chart is provided by HighCharts (find more information about HighCharts on https://www.highcharts.com/demo). Initially, the chart shows the historical stock price (in red line with filling the area below, two digits after decimal) and volume (in white bar)

for the **PAST SIX MONTHs** by an interval of **ONE DAY**. Figure 8 shows an example of the Stock Price/Volume chart.

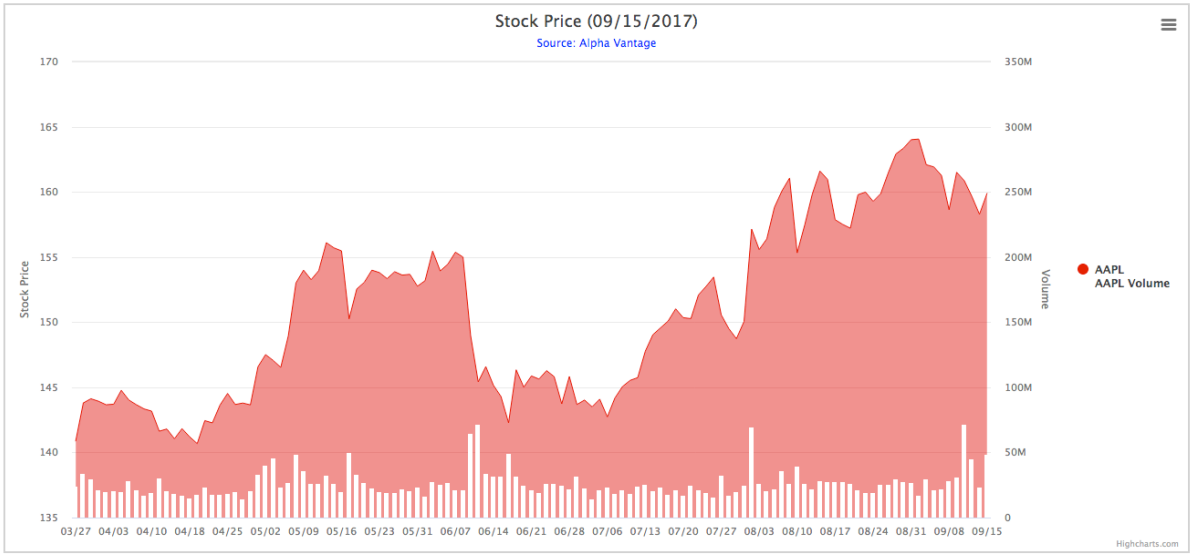

Figure 8: An Example of Chart showing Stock Price/Volume

The title of the chart for showing price/volume is "**Stock Price (MM/DD/YYYY)**", where "MM/DD/YYY" is today's date. The subtitle of the chart should be "Source: Alpha Vantage" and should hyperlink to the Alpha Vantage website: https://www.alphavantage.co/. The title of the Y-axis is "**Stock Price**" when showing price, and "**ACRONYM**" for indicators. The X-axis tickers display the date in the format of **MM/DD,** and have an interval of **ONE WEEK**.

When users click on the indicators in the stock information table, the chart should switch to show the corresponding indicator values. The range and interval for indicators are the same as price/ volume. The title should be updated to "**FULL NAME OF INDICATOR (ACRONYM)**" and the title of Y-axis should be "**ACRONYM**". Figure 9 shows the chart displaying the indicator of SMA of Apple Inc.

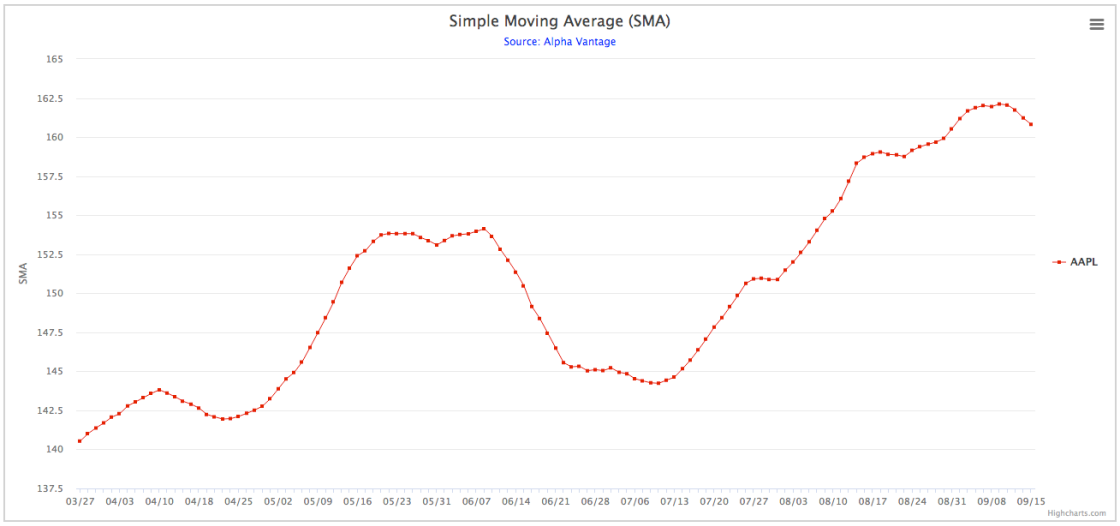

Figure 9: A Sample indicator chart showing SMA of Apple Inc.

Note some indicators may have multiple fields. The chart needs to handle this case and show all of them in different color. Figure 10 shows the chart displaying Bollinger Bands which contains 3 values.

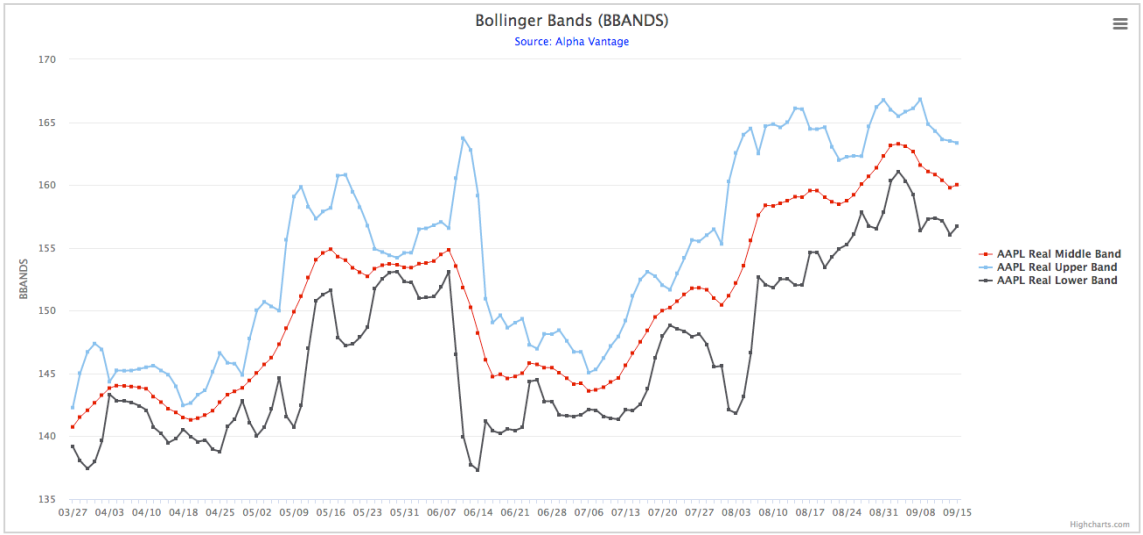

Figure 10: An Example of multiple valued indicator-Bollinger Bands

2.2.3 Displaying Stock News Table

The PHP script (i.e., **stock.php**) uses the input information (Stock Ticker Symbol) to request a XML file containing stock news about the searched company. All the stock news comes from the Seeking Alpha News RSS feed. The URL to request the news is:

https://seekingalpha.com/api/sa/combined/SYMBOL.xml (SYMBOL is the same as the stock ticker "symbol")

The response is a XML-formatted object (whose content is maintained by Seeking Alpha News). The PHP script should parse the returned XML-formatted object, extract the necessary fields, and build a JSON object to be set to the client. The JavaScript client component should display the data in a tabular format to show the title of the **latest five** pieces of news, and publication time shows after title, as shown in Figure 11.

The titles should hyperlkink to the corresponding news:

Figure 11: An Example of stock news about AAPL (Apple Inc)

Note that initially this table is hidden and when the button

is clicked, the table should toggle between expanded or hidden.

2.2.4 Displaying Error Message

If the Alpha Vantage API service returns an Invalid Call error message, the page should display "*No Record has been found*". Figure 12 shows an example when searching for a symbol "*BLABLA*" and Figure 13 shows the corresponding returned JSON.

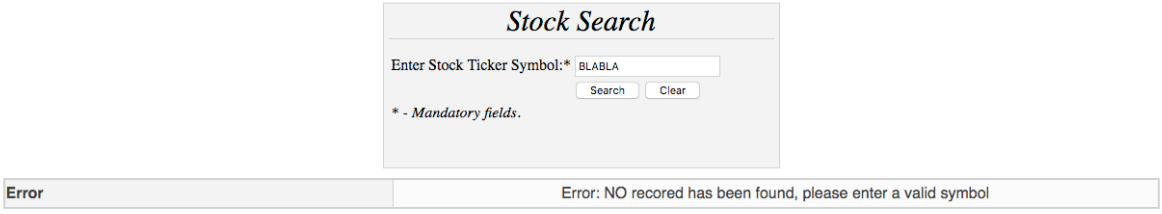

Figure 12: Search Result when there is no matching result


```
{
    "Error Message": "Invalid API call, please visit documentation
(https://www.alphavantage.co/documentation/) for function: TIME_SERIES_DAILY"
}
```

Figure 13 returned JSON when there is no matching result

2.3 Saving Previous Inputs

In addition to displaying the results, the PHP page should maintain the provided values to display the current result. For example, if a user searches for "*Stock Sticker Symbol: GE*", the user should see what was provided in the search form when displaying the results. Specifically, when clicking on the "Search" button, the page should display the result retrieved from the Alpha Vantage API service and keep the value provided in the search form. In addition, when clicking on any indicators link, the page should display the corresponding values in the chart and keep the value provided in the search form. It follows that you need to keep the whole search box/input fields and buttons even while displaying results/errors.

In summary, the search mechanism to be implemented behaves as follows:

- Based on the input data (stock ticker symbol) in the search form, construct a web service URL to retrieve the output from the Alpha Vantage API service.
- Parse the returned JSON object and extract the necessary values.
- Call the Seeking Alpha News RSS feed service and retrieve the XML-formatted output.
- Parse the returned XML-formatted output and extract the necessary values and convert to a JSON object.
- Display the stock information in tabular and chart formats.
- Display the indicators in chart formats.
- Display the stock news in tabular format.

Important Note: It is mandatory to retrieve the result from Alpha Vantage API service using JSON format and the result from Seeking Alpha News RSS feed service using XML format.

3. Hints

3.1. Alpha Vantage on demand API Documentation

To apply for and API key on Alpha Vantage and read the API documentation, please go to:

https://www.alphavantage.co/documentation/#

Click on "Claim your free API key" in the first paragraph, which will take you here:

https://www.alphavantage.co/support/#api-key

Enter your First Name, Last Name, select Student, enter your @usc.edu e-mail address and click **GET FREE API KEY**.

3.2 Stock Table Information

When the returned JSON is valid, the PHP script should retrieve the data from the response to do calculations for the fields in **Table 1** below.

Note that the "last day session" refers to the first entry of the key "Time Series". "previous closed day session" refers to the second entry of the key "Time Series".

Table Column	**Quote Service Response**
Symbol	The value of *Symbol*
Close	The Close Price of last day session
Open	The Open Price of last day session
Previous Close	The Close Price of previous closed day session
Change	The difference between current Close and previous Close
Change Percent	The value of *Change Percent* rounded to two decimal points followed by a percentage "%" character and a marker representing the price change trend of the stock
Day's Range	The Low and High of last day session
Volume	The Volume of last day session
Timestamp	The value of *Timestamp* displayed in YYYY-MM-DD format.

Table 1: Stock Table Fields

Regarding the marker icon rendered beside the values of *Change*, *Change Percent*, a red-down arrow icon is displayed if the value is negative while a green-up arrow icon is displayed if the value is positive. The icons can be found at:

http://cs-server.usc.edu:45678/hw/hw6/images/Green_Arrow_Up.png

http://cs-server.usc.edu:45678/hw/hw6/images/Red_Arrow_Down.png
http://cs-server.usc.edu:45678/hw/hw6/images/Gray_Arrow_Down.png
http://cs-server.usc.edu:45678/hw/hw6/images/Gray_Arrow_Up.png

3.3 HighCharts

Please apply for your license at:
https://shop.highsoft.com/noncomform

Once you receive the license, you can download HighCharts from the website or include the CDN version in your script. Use of the CDN version is highly recommended.

3.4 Parsing JSON-formatted data in PHP

In PHP 5, you can parse JSON-formatted data using the "*json_decode*" function. For more information, please go to http://php.net/manual/en/function.json-decode.php.

To read the contents of a JSON-formatted object, you can use the "*file_get_contents*" function.

3.5 Parsing XML files in PHP

You are free to choose any XML parsing library for PHP, but we recommend the SimpleXML library. The SimpleXML library provides a simple way of getting an XML element's name, attributes, and text. As of PHP 5, the SimpleXML library functions are part of the PHP core. No installation is required to use these functions. The following two tables show a set of functions which you may use. For more detailed information, please go to:

- http://www.w3schools.com/php/php_xml_simplexml.asp
- http://php.net/manual/en/book.simplexml.php
- http://www.w3schools.com/php/php_ref_simplexml.asp

Function	Description
__construct()	Creates a new SimpleXMLElement object
addAttribute()	Adds an attribute to the SimpleXML element
addChild()	Adds a child element the SimpleXML element
asXML()	Formats the SimpleXML object's data in XML (version 1.0)
attributes()	Returns attributes and values within an XML tag

children()	Finds the children of a specified node
count()	Counts the children of a specified node
getDocNamespaces()	Returns the namespaces DECLARED in document
getName()	Returns the name of the XML tag referenced by the SimpleXML element
getNamespaces()	Returns the namespaces USED in document
registerXPathNamespace()	Creates a namespace context for the next XPath query
saveXML()	Alias of asXML()
simplexml_import_dom()	Returns a SimpleXMLElement object from a DOM node
simplexml_load_file()	Converts an XML file into a SimpleXMLElement object
simplexml_load_string()	Converts an XML string into a SimpleXMLElement object
xpath()	Runs an XPath query on XML data

Table 2: PHP 5 SimpleXML Functions

Function	Description
current()	Returns the current element
getChildren()	Returns the child elements of the current element
hasChildren()	Checks whether the current element has children
key()	Return the current key
next()	Moves to the next element
rewind()	Rewind to the first element
valid()	Check whether the current element is valid

Table 3: PHP 5 SimpleXML Iteration Functions

3.6 Encoding data into JSON-formatted objects in PHP

In PHP 5, you can encode data into JSON-formatted objects using the "*json_encode*" function. For more information, please go to http://php.net/manual/en/function.json-encode.php.

4. Files to Submit

In your course homework page, you should update the **HW6 link** to refer to your new initial web page for this exercise. Also, submit your files (only a single .PHP file) 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 discussions and explanations in Piazza related to this homework are part of the homework description and grading guidelines. So please review all Piazza threads, before finishing the assignment. If there is a conflict between Piazza and this description and/or the grading guidelines, **Piazza always rules**.
- You should not use JQuery for Homework 6.
- You **should not call the Alpha Vantage APIs directly from JavaScript**, bypassing the Apache proxy, except for the Indicators. Similarly, you **should also not call the Seeking Alpha XML RSS feed directly from Javascript**. Doing either this will result in an **8-point penalty.**
- **The link to the video is** : https://www.youtube.com/watch?v=lkCrvaPTkgs