Homework 8: Yahoo! Weather & Facebook Mashup – an AJAX/JSON/AWS/Java Exercise

1. Objectives

- Become familiar with the AJAX, JSON & XML technologies.
- Use a combination of HTML, CSS, DOM, XMLHttpRequest, XML and Java Servlets.
- Get hands-on experience in Amazon cloud computing (AWS)
- Get hands-on experience on how to use YUI library to beautify a web page
- Provide an interface to perform weather search from Yahoo! and post details to Facebook.

2. Background

2.1 AJAX & JSON

AJAX (**A**synchronous **Ja**vaScript + **X**ML) 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://www-scf.usc.edu/~csci571/Slides/ajax.ppt**.

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://www-scf.usc.edu/~csci571/Slides/JSON1.ppt**.

2.2 Yahoo! Weather

http://weather.yahoo.com/ is an online weather service including current condition and forecast.

In Homework #6 a PHP script together with your Apache server provided the search functionality. In this exercise you will re-use your code to produce XML instead of HTML as you did in homework #6, plus you manage forecasts besides the current weather conditions.

2.3 YUI Library

In this homework, you must use the YUI library (http://yuilibrary.com/) to beautify your page. YUI is a free, open source JavaScript and CSS library for building richly interactive web applications. You will apply its CSS style on the form elements and also use the DataTable widget to display the weather forecast.

2.4 Facebook

Facebook is a global social networking website that is operated and privately owned by Facebook, Inc. Users can add friends and send them messages, and update their personal profiles to notify friends about themselves and what they are doing.

Users can additionally post news feeds to their profiles, and these feeds may include images, besides text messages.

The Facebook homepage is available at: **http://www.facebook.com**

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 few links for Facebook Connect:

http://developers.facebook.com/blog/post/108/

http://developers.facebook.com/docs/guides/web/

2.5 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/

3. Description of the Exercise

In this exercise, you will write a web application that does the following sequence of actions:

- Allow a user to enter a "query" to get weather information from Yahoo!; the query will contain a location which can be a city name with state or country, or a ZIP code. Your JavaScript code will have to determine the location type from the location input. If the user intends to input a city name, then state or country should be provided as well. For example: Los Angeles, CA, USA or Seattle, WA or Pasadena, USA are three valid inputs. The default temperature unit is Fahrenheit;

- Use the query string to retrieve appropriate information from Yahoo!, using a modified PHP script from Homework #6. The PHP script must run on AWS, instead of cs-server.usc.edu;

- Display the current weather details, along with a table showing five-day weather forecast. The big city name should be clickable and link to rss feed page. The underlined temperature unit should be clickable and perform the unit changing function (see Figure 3). For this homework, display the weather information of only the first location (woeid) returned by Yahoo! Geolocation service;

- On the right side of the weather information, there should be a Facebook button, which allows the user to select to post either the current weather condition or the weather forecast to his/her Facebook wall;

- Authorize the user to login to Facebook;

- Post a feed of the weather information to the user's Facebook wall using the Facebook Connect API.

  The format of the feed to be posted is as follows:

  For current weather:

  "**area**

  The current condition for **city** is **weather**.

  Temperature is **current temperature**.

  Look at details: **here**"

  Enter appropriate values for **area**, **city**, **weather**, and **current temperature**. The **area** should be the city name with state and/or country. The **city** should be the city name only. The **area** (first line) and **here** should hyperlink to the weather feed page and full Yahoo!

Weather service page, respectively. The weather icon should be displayed as well.

For weather forecast:

“**area**

Weather Forecast for **city**.

day[0]: **weather**, **high**/**low**;……; **day[n]**: **weather**, **high**/**low**.

Look at details: **here**”

Enter appropriate values for **area**, **city**, **day[i]**, **weather**, **high**, and **low**. The **area** should be the city name with state and/or country. The **city** should be the city name only. The forecast of each day should be listed using the format above. The **area** (first line) and **here** should hyperlink to the weather feed page and full Yahoo! Weather service page, respectively. Use the default weather icon from http://www-scf.usc.edu/~csci571/2013Fall/hw8/weather.jpg.

In both of the above cases, users can enter the message of their choice in the message box. (Refer to Figure 5 for more details on the format of the message to be displayed)

A snapshot of the initial user interface is shown in Figure 1.

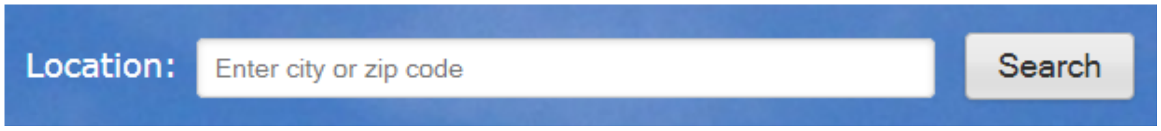
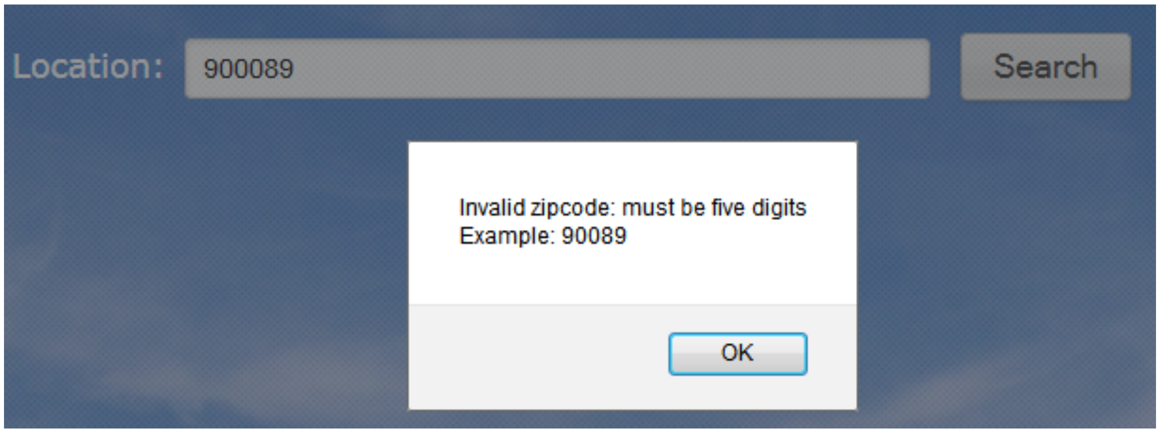
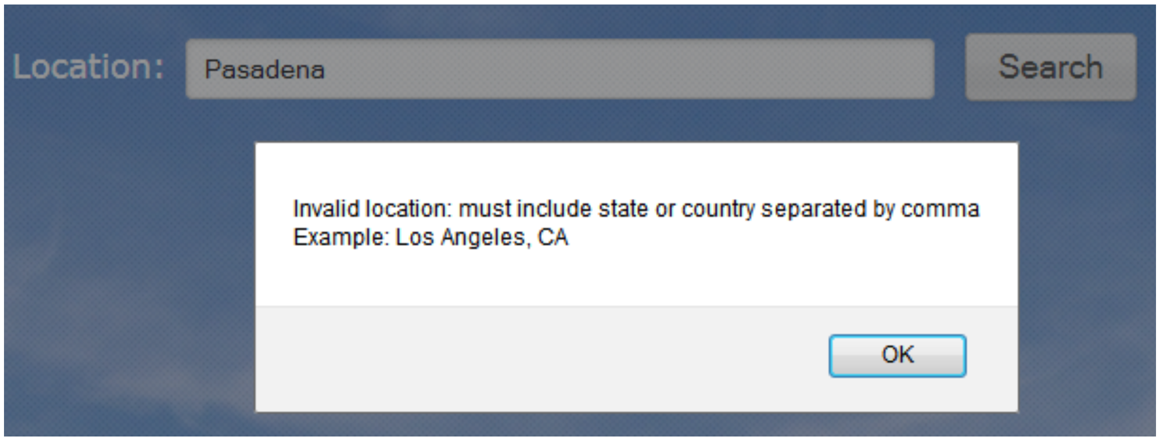


Figure 1 – Initial User Interface

Your program should check whether the user inputs a valid value in the location edit box. If the zip code is malformed or the city name is accompanied by neither state nor country, the request should not be forwarded to the Java Servlet and you should display an alert message, which is shown in Figure 2.



Invalid zipcode: must be five digits
Example: 90089



Invalid location: must include state or country separated by comma
Example: Los Angeles, CA

Figure 2 – Malformed Input

To implement this exercise you are required to write a combination of HTML, PHP, JavaScript and Java Servlet programs. The top-level interface consists of three areas:

- A form including a text area to enter the location and a "Search" button;
- A dynamic area that displays the current weather and a table with the five-day forecast;
- A "Facebook" button that posts the details, corresponding to the selected entry, to a user's feed page.
- A Yahoo! Weather Service icon displayed in the bottom right corner of the page, which links to the full Yahoo! Weather site.

Once the location has been entered and the "Search" button has been pressed, the form calls a JavaScript function. This function first performs basic validation of the input as stated above. When doing the validation, your program should also prepare the location type and temperature unit (default is Fahrenheit) to be sent. Once the validation is successful, the JavaScript function executes XMLHttpRequest to start an asynchronous transaction with a Java Servlet running under Tomcat, and passing the "query string" including location, type and temperature unit as parameters of the transaction.

The Java Servlet in turn performs the following three steps:

First, the Java Servlet extracts the query string and then it calls the PHP script on AWS, modified after Homework #6, to retrieve data from Yahoo!. For example, if your AWS server domain is called default-environment-randomstring.elasticbeanstalk.com, a query of the following type will be generated:

If location = 90089, the generated URL will be

http://default-environment-randomstring.elasticbeanstalk.com/?location=90089&type=zip&tempUnit=f

Secondly, the PHP script running on AWS would respond by returning the XML of the following format:

```
<weather>
       <feed>http://weather.yahooapis.com/forecastrss?w=2468964&u=f</feed>
       <link>http://us.rd.yahoo.com/dailynews/rss/weather/Pasadena__CA/*http://weathe
r.yahoo.com/forecast/USCA0840_f.html
       </link>
       <location city="Pasadena" region="CA" country="United States"/>
       <units temperature="F"/>
       <condition text="Fair" temp="85"/>
       <img>http://l.yimg.com/a/i/us/we/52/33.gif</img>
       <forecast day="Sat" low="61" high="92" text="Clear"/>
       <forecast day="Sun" low="60" high="95" text="Sunny"/>
       <forecast day="Mon" low="58" high="87" text="Partly Cloudy"/>
       <forecast day="Tue" low="57" high="78" text="Sunny"/>
       <forecast day="Wed" low="53" high="68" text="Few Showers"/>
</weather>
```

Notice that in Homework #6 your PHP script produced HTML. In this exercise, the output must be changed to XML and the PHP code must run on AWS.

Now, the Java Servlet extracts the **appropriate information** from this XML.

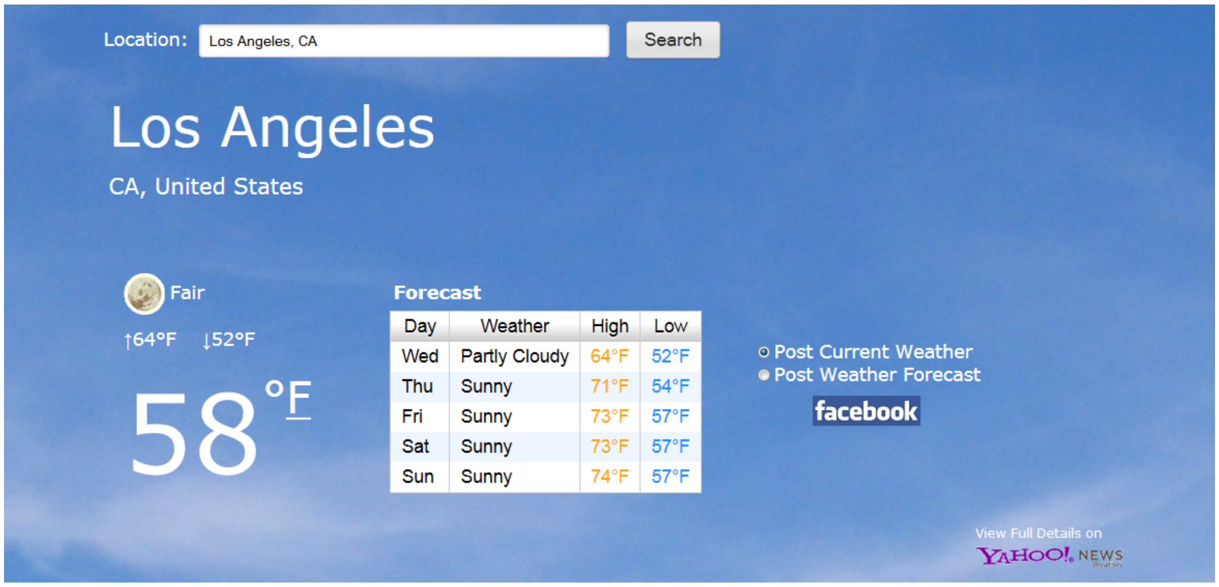
Lastly, the Java Servlet converts the XML to JSON and produces a JSON string that is returned asynchronously to the original XMLHttpRequest.

The format of the JSON string that needs to be generated is as follows:

```
{
   "weather":{
      "forecast":[
         {
            "text":"Clear",
            "high":92,
            "day":"Sat",
            "low":61
         },
         {
            "text":"Sunny",
            "high":95,
            "day":"Sun",
            "low":60
         },
         {
            "text":"Partly Cloudy",
            "high":87,
            "day":"Mon",
            "low":58
         },
         {
            "text":"Sunny",
            "high":78,
            "day":"Tue",
            "low":57
         },
         {
            "text":"Few Showers",
            "high":68,
            "day":"Wed",
            "low":53
         }
      ],
      "condition":{
         "text":"Fair",
         "temp":85
      },
      "location":{
         "region":"CA",
         "country":"United States",
         "city":"Pasadena"
      },
      "link":"http://us.rd.yahoo.com/dailynews/rss/weather/Pasadena__CA/*http://weath
er.yahoo.com/forecast/USCA0840_f.html",
      "img":"http://l.yimg.com/a/i/us/we/52/33.gif",
      "feed":"http://weather.yahooapis.com/forecastrss?w=2468964&u=f",
      "units":{
         "temperature":"F"
      }
   }
}
```

After obtaining the query results from the callback of XMLHttpRequest, the JavaScript program displays the appropriate table in the "dynamic" area of the web page. **Also note that successive queries will clear the data of the dynamic area and overwrite it with new data.**

The "dynamic" area of the page is shown in Figure 3, below the search area.



Location: Los Angeles, CA

Los Angeles

CA, United States

Fair

↑64°F ↓52°F

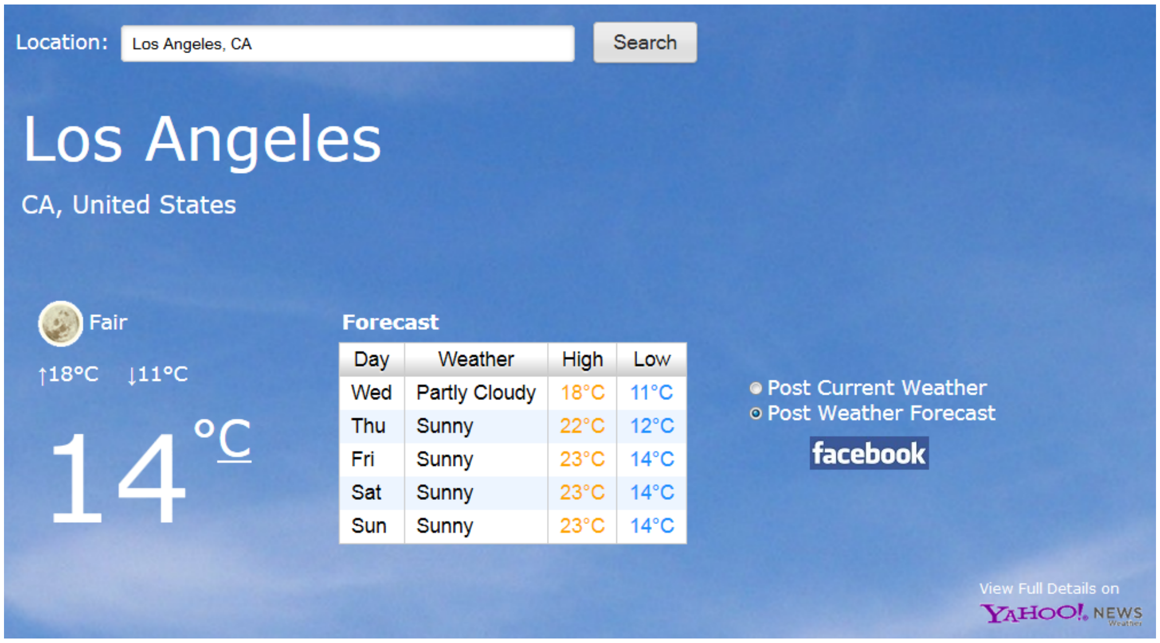
58 °F

Forecast

Day	Weather	High	Low
Wed	Partly Cloudy	64°F	52°F
Thu	Sunny	71°F	54°F
Fri	Sunny	73°F	57°F
Sat	Sunny	73°F	57°F
Sun	Sunny	74°F	57°F

Figure 3 – Result Page

After clicking the underlined default temperature unit (F for Fahrenheit), all the displayed units should be changed to C (Centigrade) as shown in the following snapshot.



Los Angeles

CA, United States

Fair

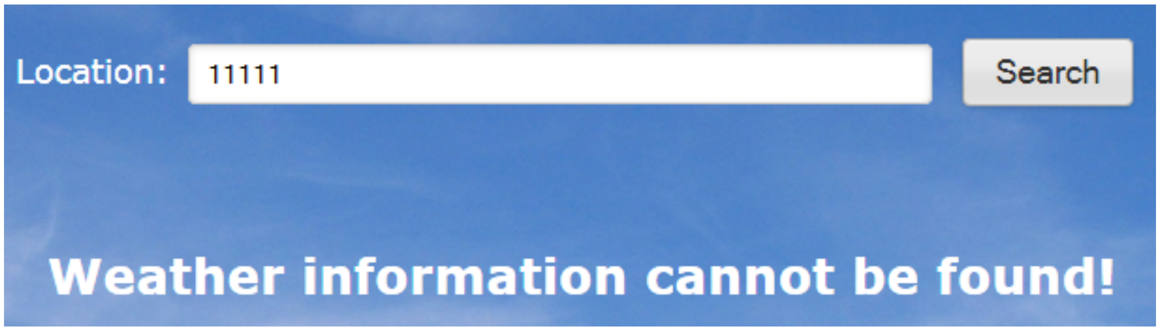
↑18°C ↓11°C

14 °C

Forecast

Day	Weather	High	Low
Wed	Partly Cloudy	18°C	11°C
Thu	Sunny	22°C	12°C
Fri	Sunny	23°C	14°C
Sat	Sunny	23°C	14°C
Sun	Sunny	23°C	14°C

If the weather information cannot be found, an appropriate message should be shown as in the following snapshot.

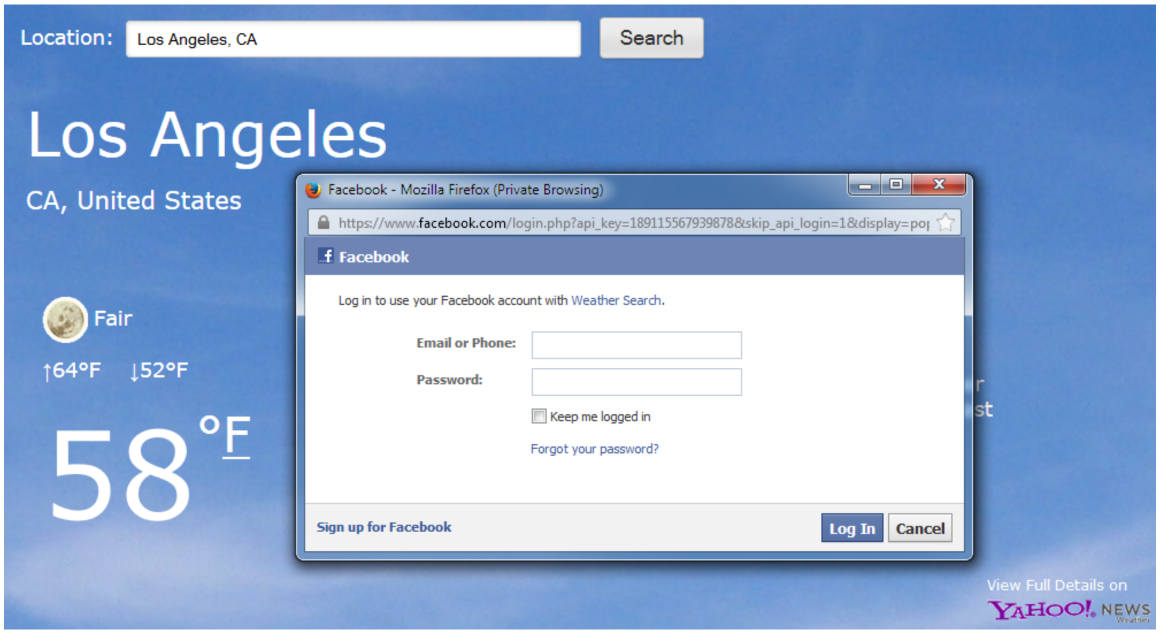


Weather information cannot be found!

Next, the user is allowed to click on the “Facebook” button. 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;
- Posts an Update Status message to the feed (described at the beginning of section 3).
- 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/



Location: Los Angeles, CA

Los Angeles

CA, United States

Fair

↑64°F ↓52°F

58 °F

Facebook - Mozilla Firefox (Private Browsing)

https://www.facebook.com/login.php?api_key=189115567939878&skip_api_login=1&display=pop

Log in to use your Facebook account with Weather Search.

Email or Phone:

Password:

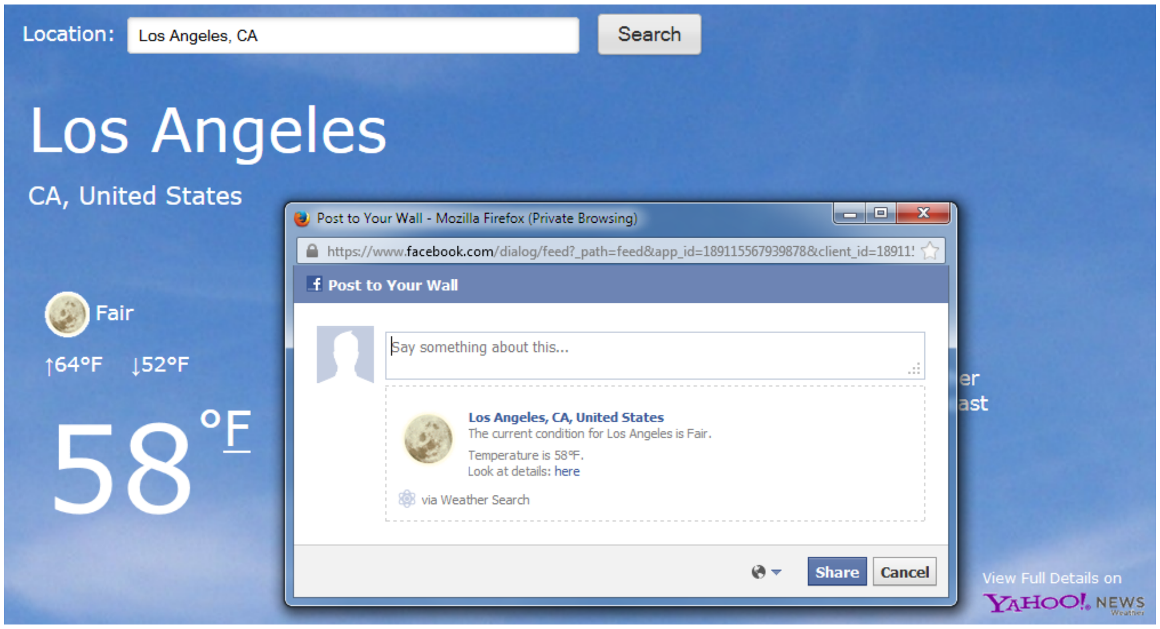
Keep me logged in

Forgot your password?

Sign up for Facebook

View Full Details on YAHOO! NEWS

Figure 4 – Facebook Login



Location: Los Angeles, CA

Los Angeles

CA, United States

Fair

↑64°F ↓52°F

58 °F

Post to Your Wall - Mozilla Firefox (Private Browsing)

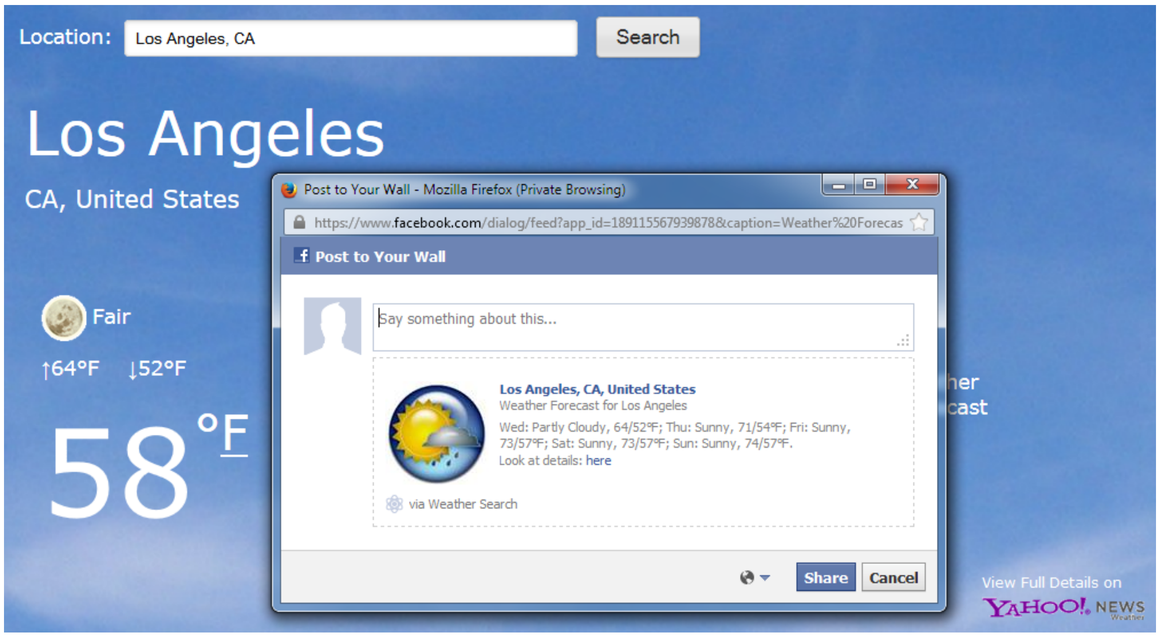
https://www.facebook.com/dialog/feed?_path=feed&app_id=189115567939878&client_id=18911!

Say something about this...

Los Angeles, CA, United States
The current condition for Los Angeles is Fair.
Temperature is 58°F.
Look at details: here

via Weather Search

View Full Details on YAHOO! NEWS



https://www.facebook.com/dialog/feed?app_id=189115567939878&caption=Weather%20Forecas

Say something about this...

Los Angeles, CA, United States
Weather Forecast for Los Angeles
Wed: Partly Cloudy, 64/52°F; Thu: Sunny, 71/54°F; Fri: Sunny, 73/57°F; Sat: Sunny, 73/57°F; Sun: Sunny, 74/57°F.
Look at details: here

via Weather Search

Figure 5 – Step 1-3 to post the current condition or weather forecast to user's feed

4. Implementation Hints

- *Step 1: Writing your JavaScript Program – set up an Ajax transaction*

  The JavaScript invoked by the Search button click event should do all of the following:

  - Assign the "callback" function;
  - Assemble the "url" parameter of the GET request as a reference to the Java Servlet to be invoked:

    Example url :http://cs-server.usc.edu:XXXXX/servlet/MyServlet?parameters

  - Call the XMLHttpRequest method (see Ajax Slide 24) and create the request object.
  - Prepare the GET XMLHttpRequest using the setRequestHeader method:

```
req.open("GET", url, true);

req.onreadystatechange = myCallback;

req.setRequestHeader("Connection", "Close");

req.setRequestHeader("Method", "GET" + url + "HTTP/1.1");
```

- *Step 2: Writing your JavaScript Program – Execute Ajax Transaction*

  The JavaScript should finally invoke the XMLHttpRequest **send** method (see Ajax slide

24).

The "callback" function should check for completion of the transaction (request **readyState** equal to **4** and **status** equal to **200** (see AJAX slide 27 and JSON slide 5); use **eval()** and the **responseText** to retrieve the resulting JSON data (see JSON slide 5), and display the returned information properties to the "dynamic" area.

- *Step 3: Use the Java Servlet to respond to XMLHttpRequest and retrieve the information listings from Yahoo!*

  The Java Servlet referred above (in step 1) as **/servlet/MyServlet** (see 1.c above) should be invoked using **doGet()**.

  The Java Servlet should do all of the following:

  - Initiating a connection with the modified PHP script from Homework #6, using AWS, to retrieve the appropriate information from Yahoo!.

- *Step 4: Use the Java Servlet to retrieve the XML file content*

You may have to use an XML parser (for example, JAXP or JDOM). The steps to retrieve XML file contents are as follows:

Step a: Get the XML content based on the URL above in Step 3.a.

- You need to open a URL connection to get the file you want.

  To create a URL connection to AWS:

```
URL url = new URL(urlString);

URLConnectionurlConnection = url.openConnection();

urlConnection.setAllowUserInteraction(false);

InputStreamurlStream = url.openStream();

//read content
```

Step b: Parse the XML file using an XML parser

- Any XML parser can be used to parse the XML file. You can use methods like **getNodeName()** to access these elements. A good choice might be the JDOM library, which you get from:http://www.jdom.org/downloads/index.html

- *Step 5: Use the Java Servlet to process the XML data*

As you parse the data, you will build an output string, converting the XML data into JSON format, as described in section 3.

Finally you will return the JSON as a single string to the calling JavaScript program. To easily create a JSON string, you might find useful the JSON-RPC library available at:

http://mirrors.ibiblio.org/pub/mirrors/maven/com.metaparadigm/jars/json-rpc-1.0.jar

The Java Servlet should handle exceptions such as MalformedURLException and IOException. The Java Servlet should also handle error responses sent from the PHP code running on AWS and reply with an appropriate error, a JSON message to the initial JavaScript XMLHttpRequest. This way, the JavaScript callback function will be able to inform the user that an error has occurred.

- *Step 6: Writing your JavaScript Program – Generate dynamic area*

After you retrieve the JSON result from Java Servlet, you need to generate the dynamic area. The weather forecast table is a YUI widget. Please refer to http://yuilibrary.com/yui/docs/datatable/ to learn about how to populate data and set the style of the table.

- *Step 7: Writing your JavaScript Program – Post media details to Facebook*

Once the media properties are displayed in the dynamic area, the user should be able to click on the "Facebook" button to populate the post with either the current weather of the forecast.

- *Step 8: Writing your JavaScript Program – Authorize Facebook User*

Once the user clicks on the Facebook button, the program invokes the Facebook Connect API and authorizes the user. The recommended API to use is FB.Init, which is documented at: http://developers.facebook.com/docs/reference/javascript/FB.init/

Also look at the code listed under "Loading" in the JavaScript SDK page at:

https://developers.facebook.com/docs/reference/javascript/

- *Step 9: Writing your JavaScript Program – Post media information to Facebook News Feed*

There are several methods to post a message to the user's feed page (the "wall").

One such method uses the Fb.ui() API with the feed dialog, documented at:

http://developers.facebook.com/docs/reference/javascript/FB.ui/

The feed dialog is documented at:

https://developers.facebook.com/docs/reference/dialogs/feed/

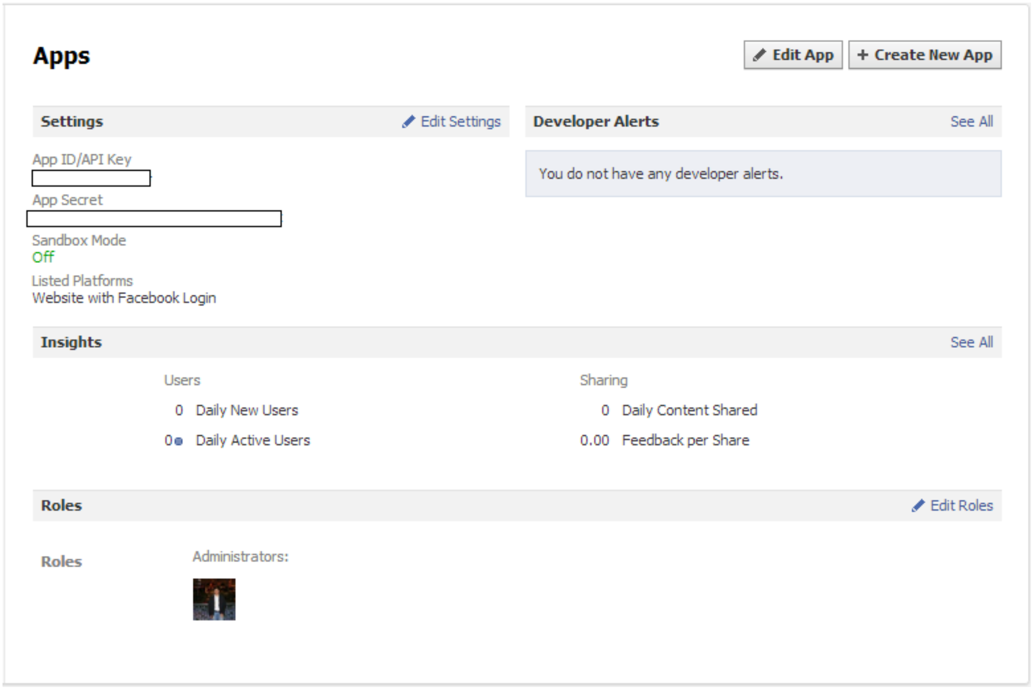
Once the user is authorized, and an appropriate session token has been obtained, the text of the selected entry is posted to the user's news feed page (the "wall"). A subsequent posting will not require the user to log in again.

Additional information that may be useful to you is available at "Facebook for Websites" web page: https://developers.facebook.com/docs/guides/web/

5. Prerequisites

This homework requires the use of the following components:

- A servlet-based web server, Tomcat 4.1.27. Instructions on how to load Tomcat 4.1.27 can be found here:
  http://www-scf.usc.edu/~csci571/2013Spring/tomcatinstall.html.
  A tar version of Tomcat 4.1.27 can be found here:
  http://www-scf.usc.edu/~csci571/download/jakarta-tomcat-4.1.27.tar.
- The Java Servlet library (servlet-api.jar) to perform HTTP transactions using methods such as doGet() or doPost() from Java.
  You may find it here: http://repo1.maven.org/maven2/javax/servlet/servlet-api/2.5/
- A Java XML parser library. You may use the JDOM, an object model that uses XML parsers to build documents, available in the Download section of the jdom website
  http://www.jdom.org/downloads/index.html
- You may also use JAXP, the Java API for XML Parsing. A good tutorial on JAXP is available at http://www-106.ibm.com/developerworks/xml/library/x-jaxp1.html
  or http://docs.oracle.com/javase/tutorial/jaxp/intro/index.html
- The YUI library, which can be downloaded from http://yuilibrary.com/.
- An AWS account. Signup and installation of your Apache+PHP server on AWS is available at http://www-scf.usc.edu/~csci571/2013Fall/hw7/aws.pdf. A $100 credit for AWS services will be provided to you by e-mail.
- You need to create a Facebook Platform application:
  To do that you will need to add the Facebook Developer application: go to https://developers.facebook.com/ and in the apps section, click **Create New App**. Once you've added the Facebook Developer application to your account, you can begin creating your application for Facebook. You should be getting an **API Key** and **Application Secret** that you will have to use with the JavaScript Client Library FB.init API.


Apps

Settings

App ID/API Key

App Secret

Sandbox Mode
Off

Listed Platforms
Website with Facebook Login

Developer Alerts

You do not have any developer alerts.

Insights

Users

0 Daily New Users

0 Daily Active Users

Sharing

0 Daily Content Shared

0.00 Feedback per Share

Roles

Roles

Administrators:

Figure 6 – Sample Facebook Developers App Page

6. Deployment Structure

To write your own Java Servlets program using Tomcat 4.1.27, you need to:

- Successfully install Tomcat 4.1.27 on your machine.
- Go to **$CATALINA_HOME/webapps/examples** directory.
- Place the HTML, CSS and JavaScript (.js) files in the Tomcat **servlets** subdirectory.
- Place your Java Servlets file (.java) in the/**WEB_INF/classes** folder. So the path of your Servlets file is **http://server_name:port/examples/servlet/your_servlet_name**
- Add appropriate sections to the **WEB-INF/web.xml** file, as in:

```
<servlet>
<servlet-name>weather_search</servlet-name>
<servlet-class>WeatherSearch</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>weather_search</servlet-name>
<url-pattern>/servlet/weathersearch</url-pattern>
</servlet-mapping>
```

- To avoid UTFDataFormatException during file IO operation, you have to use JDK 1.3 or later for Tomcat. In the **.cshrc** file under your home directory, add the entries:

```
setenv JAVA_HOME /usr/j2se
setenv PATH /usr/j2se/bin:${PATH}
```

- Before you issue a request to your Java Servlet file, you need to compile it. You will need a Java Servlet class to compile your code, so open the .cshrcfile, and add the full path to the Tomcat file that implements the Servlet class APIs located in **../jakarta-tomcat-4.1.27/common/lib/servlet.jar** to your CLASSPATH variable, and use the variable with the classpathswitch of the javac compiler.

- Then run "**source .cshrc**" and you are ready to compile your Java files.

7. Material You Need to Submit

On your course homework page, your link for this homework should go to a page that includes your JavaScript/HTML program. You should submit all source code files including HTML (.html), Images (.png, .jpg or .gif), Cascading Style Sheets (.CSS), JavaScript (.js), PHP scripts (.php) and Java Servlets (.java) electronically to the csci571 account so that it can be graded and compared to all other students' code via the MOSS code comparison tool.

8. Partial Credit

If you complete Step 1 through 7 as listed in Section 4, you will obtain 6 points. If you also complete Steps 8 and 9 (authorizing a user to Facebook and posting the message to the user's Facebook news feed), you will obtain the full 10 points for the exercise. Please see the grading guidelines for additional notes regarding required components (i.e., Java Servlet proxy and AWS).