Homework 6: Server-side Scripting

1. Objectives

- Get experience with Perl (or PHP, Python) programming language;
- Get experience with CGI programming;
- Get experience parsing text using regular expressions

2. Description

In this exercise, you are asked to create a webpage, which allows you to choose a hotel chain from a list and enter a city to look for the hotels of the chosen chain and display the result in the tabular format. The information of the hotels is grabbed from the website http://www.tripadvisor.com.

A user will first open the page, called **hotel_search.html**, where he/she can choose the hotel chain and enter the name of the city he/she wants to look for the hotel. The example is shown in Fig 1.

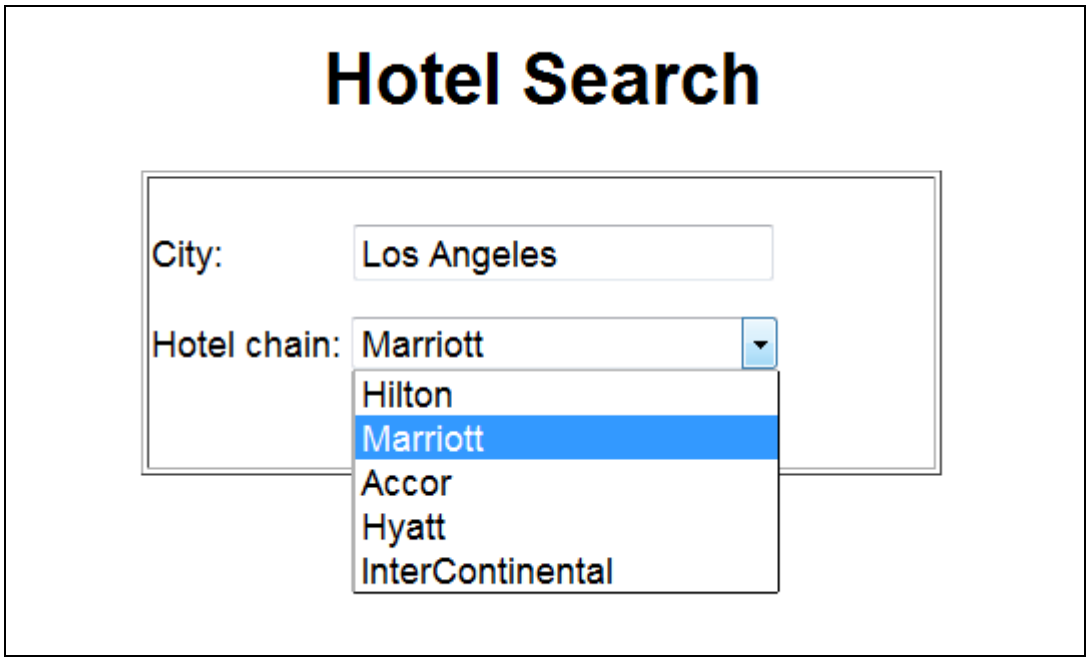

Figure 1: Enter city name and choose a hotel chain

After the user has entered the name for the city and the hotel chain and clicked on the search button, your program (typically a JavaScript function) will check if the user has entered the name for the city or not. If the user did not enter the city name, a message (alert) must be shown and ask the user to enter the data. The list can contain 5 to 7 different hotel chains. Make one of them to be default-selected. Example of the search button and the alert message is shown in Fig 2 and Fig 3 below:

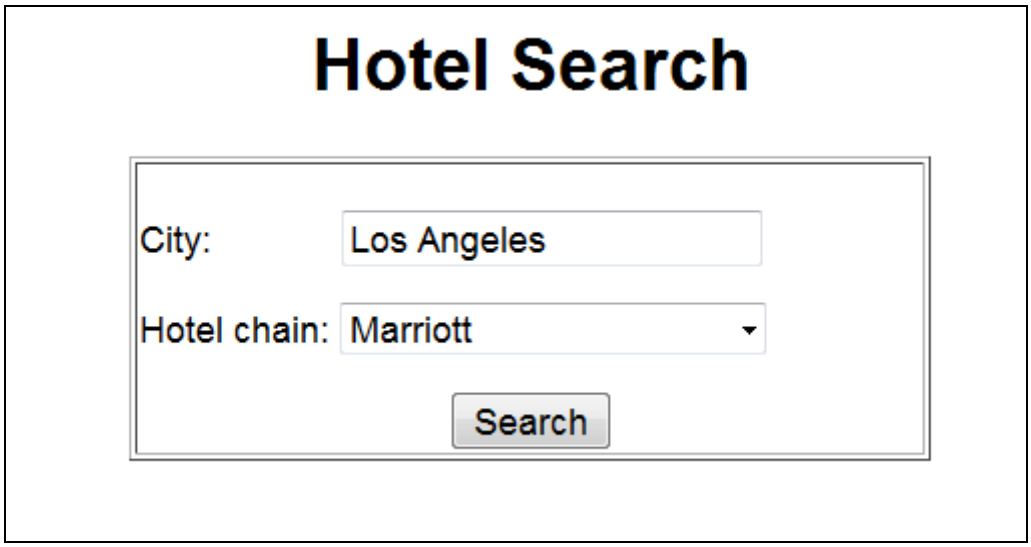

Figure 2: Search Button

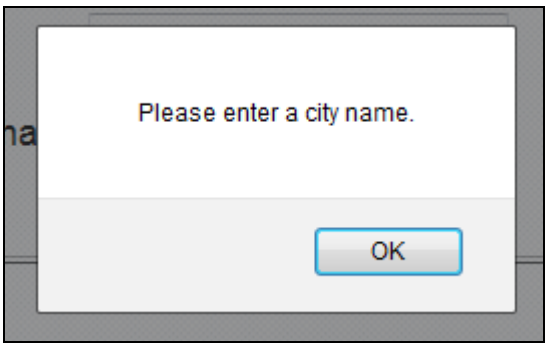

Figure 3: Alert message

When the city name is not empty, clicking on the Search button with call a Perl script, for example, **search_result.pl**. This script will grab the data sent from the html page (**hotel_search.html**) and receive the hotel information from Trip Advisor (http://www.tripadvisor.com). The way it works is: the perl script will use the data (hotel chain and city name) to construct a URL to the Trip Advisor website: For example, if you enter Los Angeles and choose Marriott, then the URL would lool like this: http://www.tripadvisor.com/Search?q=marriot+los+angeles

The perl script will use a library, called LWP, to retrieve the source code from Trip Advisor given the URL, use regular expressions to extract the following data:

1. The name of the hotel
2. The location
3. The image (small icon)
4. The rating (number of stars)
5. The number of reviews
6. The link to the reviews page

And display the data in the format of a table. The example of result is shown below:

Search Result

Marriott hotel in Los Angeles:

Image	Name	Location	Rating out of 5	Reviews
	Los Angeles Airport Marriott	Los Angeles, California	4	581
	Marriott Beverly Hills	Los Angeles, California	4	257
	Marriott Los Angeles Downtown	Los Angeles, California	3.5	233
				

Figure 4: Search Result

Keep in mind that there might be multiple pages of result for a search query. You only need to grab the first page returned by Trip Advisor.

Details are as follow.

- The first column of the table displays the small cover picture of the hotel.
- The second column displays the name of the hotel.
- The third column displays the location.
- The fourth column displays the rating of the hotel: number of stars out of 5.
- In the last column, the number is the number of reviews for that corresponding hotel; clicking on the number will you to the actual page of Trip Advisor where you can read the reviews. An example URL of the review page is: http://www.tripadvisor.com/Hotel_Review-g32655-d78037-Reviews-Los_Angeles_Airport_Marriott-Los_Angeles_California.html#REVIEWS
- If a non-sense city name is entered, for example, Bahama Mamaria, instead of the table, a message should be displayed to show that there is no result for the query. See Fig 5.

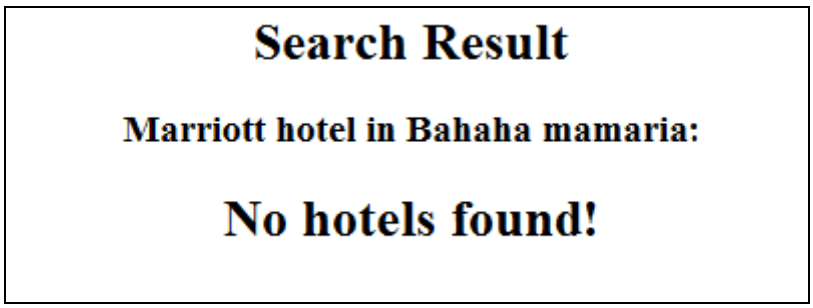

Figure 5: No result

Some notes:

- You are free to design your own html page and the structure of the table of result. However, the search result must be well displayed, at least as good as it's shown in Fig 4.
- You should not use any XML or HTML parsers in PERL. You should write your own parsing code. Also, all parsing (or scraping) of the web page (or any web page) should be done using PERL regular expressions, and NOT using any PERL libraries especially designed to simplify parsing web pages. This is not to be confused with Perl standard functions, which you are free to use. Example of such functions are **split**, **print**…
- The size of the table is not strict.
- Some of the figures in this file (Figs 1,2,3 and 5) are displayed with border lines to ease students in reading, and that does not mean that students have to make their HTML page display messages inside a rectangle.

3. Hints

To retrieve a web page you can use the Perl LWP module, and to show special characters correctly you can set Metadata indicating UTF-8.

```
#!/usr/bin/perl -w
...
# Add a header directive indicating that this is encoded in UTF-8
print "<meta http-equiv=\"Content-Type\" content=\"text/html;
charset=UTF-8\"/>\n";
...
# Check whether LWP module is installed
if(eval{require LWP::Simple;}){
}else{
print "You need to install the Perl LWP module!<br>";
exit;
}
...
# Retrieve the content of an URL
$url = "http://XXXXXX";

$content = LWP::Simple::get($url);
```

4. Options

We leave the choice of scripting languages to students, which means students are free to choose any scripting languages (Perl, PHP or Python) to implement this assignment. There are several important points that student must pay attention to:

- If you choose PHP or Python, we won't be able to provide full support, such as answering your detailed questions regarding those two languages. We will, however, try our best to help you finish the assignment. You can always post your questions on PIAZZA; instructors will be happy to assist you. Other students, who are experts in PHP or Python, can also help you on PIAZZA.
- Please, explain your implementation (script languages, files…) in the README file.

5. Files to submit

Submit your HTML/Javascripts, Perl scripts, XML 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. Also, you need to submit a README file that briefly describes the details of the organization of your program and how it works.