CST8259 Web Programming Language II

Lab 5

Objective

- Create Ajax request and consume Ajax response using JQuery and JSON.
- Process Ajax request and generate response in PHP

Due Date

See Brightspace posting for the due date of this lab. To earn 5 points, you are required:

1. Complete the lab as required.
2. Submit your lab work to the Blackboard before the due date.
3. Demo your lab work during the following week's lab sessions.

Preparation

1. Download CST8259Lab5.zip from the Brightspace. Unzip the file to a HTML file **RestaurantReviews.html**.

2. If you use AMPPS/NetBeans, create a PHP Web Application project inside AMPPS' www folder. You can name your project **CST8259Lab5** or **Lab5**

   If you use other Apache/PHP distribution, you may need to place your project inside some different folder, e.g., **hdoc**

3. Copy the HTML file **RestaurantReviews.html** into the project folder. You should **NOT** modify this HTML file, instead, attach the required functionality to this page using JavaScript and jQuery.

   This web application development approach is call **unobtrusive JavaScript**. The main advantage of unobtrusive JavaScript is the complete separation of HTML page, JavaScript and sever side scripting so that they can be developed and/or maintained separately by different specialized developers without accidently breaking each other's code.

4. Create a folder named **js** inside the project folder for JavaScript files you will create.

5. Create a folder named **Date** inside the project folder and copy your **Restaurant_Review.xml** file into this folder.

6. Create a folder named **php** inside the project folder for PHP files you will create.

Requirements

Create two JavaScript files **RestaurantReviews.js** and **Config.js** in the project's **js** folder and two PHP files in the project's php folder to accomplish the following functionality;.

- When HTML page complete loading, the JavaScript in **RestaurantReviews.js** makes an Ajax request to the server to get the names of the restaurants in the **restaurant_review.xml** file.

- Upon receiving the above request, the php scripts in **RestaurantReviews**.**php** loads **restaurant_review.xml** file and responses with a JSON string of the restaurant names.

- Upon receiving the response from the server, the JavaScript in **RestaurantReviews**.**js** populates the dropdown list for restaurant names.

- When the user selects a restaurant from the dropdown list, the JavaScript **RestaurantReviews.js** makes an Ajax request with the id of the restaurant to the server to get the details of the selected restaurant.

  You can use the sequence number of the restaurant in **restaurant_review.xml** file as its id.

- Upon receiving the above request, **RestaurantReviews.php** retrieves the details of the restaurant identified by the id and responses with the restaurant's address (street address, city, province, and postal code), the summary and the rating (including max, min and current rating) of the specified restaurant in a JSON string.

- Upon receiving the response from the server, **RestaurantReviews.js** displays the selected restaurant's address (street address, city, province, and postal code), summary, and rating.

  The rating should be a dropdown list whose selections should be determined by max and min rating received from the server.

- The user can change the restaurant’s address, summary and selects a different rating from the dropdown list.

- When the user clicks the **Save** button, **RestaurantReviews.js** makes an Ajax request to the server sending the revised restaurant data to a server in a JSON string.

- Upon receiving the **save** request, **RestaurantReviews.php** updates the restaurant data and save the updated restaurant back in the **restaurant_review.xml** file.

  RestaurantReviews.php sends a message in a JSON string to info the user whether the saving is successful and not.

- Upon receiving the message from the sever, **RestaurantReviews.js** displays the message to the user.

  Note:

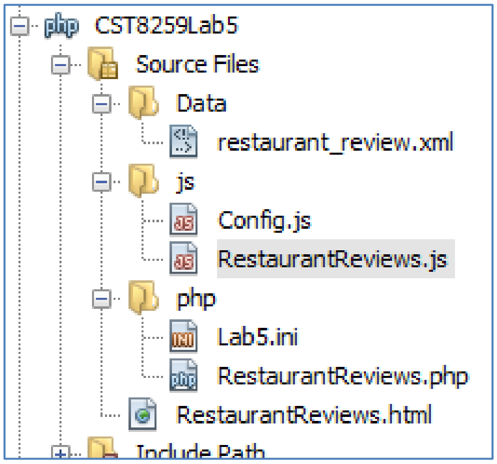
  1. The functionality of this lab is the same as Lab 4. But throughout the above-described client (browser) server interactions, no page loading should happen, that is, the user should not experience page-refresh during his/her interaction with the web application.

  2. Re-user your PHP code in Lab 4 for accessing **restaurant_review.xml** file.

  3. The path to **restaurant_review.xml** file should be maintained in Lab5.ini file

  4. The URLs of the Ajax requests should be maintained inside **Config.js** file

  5. If you use NetBeans as your IDE, your project should look like:



```
CST8259Lab5
  Source Files
    Data
      restaurant_review.xml
    js
      Config.js
      RestaurantReviews.js
    php
      Lab5.ini
      RestaurantReviews.php
    RestaurantReviews.html
  Include Path
```