Project #6 (Data Validation/Drawing Device Recorder) CMSC389N Due Tue Jul 7 11:59 pm Summer 2015

Objective

To practice forms, data validation, data collection, events and object-oriented programming in JavaScript.

Grading

- (20 pts) **Form Implementation**
  - Text fields for firstname, lastname, phone
  - Use of type=number for age, height, and weight
  - Use of checkboxes for conditions
  - Use of radio buttons for time period
  - Use of combo box for type of study
  - Use of text fields for study id
  - Text area
  - Clear and Submit buttons
  - All text fields (except text area) marked as required
  - Use of type=email type for email address entry
  - Place holders for study id
  - Page structure similar to expected page
  - Page validates as HTML5
- (5 pts) **CSS Style**
  - Shadow for **Research Form** title
  - Brown color legend
  - Round borders for fieldset areas
  - Centralized **CMSC Research Institute** title
- (30 pts) **JavaScript Support**
  - Phone Validation
  - Conditions (Diabetes, etc.) Validation
  - Time Period Validation
  - Study Id Validation
  - At least two functions defined
- (5 pts) **Form generates expected php script confirmation**
- (40 pts) Drawing Device Recorder

Overview

This project has two parts. The first part is an exercise on data validation using JavaScript. The second is to define an interface that will allow us to record the activity associated with a drawing program discussed in class (DrawingPointer.html).

Specifications (Data Validation)

You must define a form with the form elements you can see in the "Sample Run (Data Validation)". The following provides additional information about this part of the project:

- **General Requirements**
  1. For this project you need to install a web server environment similar to XAMPP (http://www.apachefriends.org/en/xampp.html) so the php scripts provided can be executed.
  2. Download the zip file DataValidation.zip. This files has the php scripts you need.
  3. All the files for this part of the project must be placed in the DataValidation folder that has the php scripts. To grade your project, we will place that folder in a web server and open the research.html page.
  4. Your HTML must appear in a file named **research.html**.
  5. All the form text fields (e.g., first name) must be defined as required using the required="required" HTML5 feature.
  6. Use the type=email to define the e-mail field.
  7. Your program must verify (using JavaScript) that a valid phone number has been provided. A valid phone number has the format ### ### #### where # is a digit. The message "Invalid phone number" will be generated if an invalid phone number is provided.
  8. Use of type=number for age with a minimum value of 1 and a maximum value of 125.
  9. Use of type=number to represent the height feet and inches values. For feet, the minimum value is 2 and the maximum is 7. For inches, the minimum value is 0 and the maximum is 12.
  10. Use of type=number for weight with a minimum value of 1 and a maximum value of 600.
  11. Your program must verify (using JavaScript) that at least one condition ("High Blood Pressure","Diabetes", "Glaucoma", "Asthma", "None") has been selected. If no condition is provided, your program must generate the error message "No conditions selected". If the user selects any condition and also "None", your program will generate the error message "Invalid conditions selection".
  12. Your program must verify (using JavaScript) that at least one time period ("Never", "Less than a year", "One to two years", "More than two years") has been selected. Your program will generate the message "No time period selected" if none has been selected.
  13. The "Study Information" section includes a combo box with two options: "Long Term" (default) and "Short Term".
  14. Your program must verify (using JavaScript) that a valid study id has been provided. Valid ids have the following format A### B### where # is a digit. Your program will generate the message "Invalid study id" if an invalid id is provided.
  15. If after evaluating the data, any errors are found, your program will not submit the form and will display a single message (using alert) where the fields with invalid data are identified. If the data is correct, a confirmation message ("Do you want to submit the form data?") will be displayed asking for a submission confirmation.
  16. The color associated with the legend is #A0522D
  17. Keep in mind that Komodo Edit can have problems displaying HTML5 elements. Look at your projects using Chrome.
  18. **For this project you may not use regular expressions.**
- **Functions Requirement**
  1. It is up to you to define any functions you need but you must have at least two functions. All functions and all the javascript code should be in a file named **research.js**.
  2. You should avoid code duplication.
- **Style Requirement**
  1. The style and appearance should be close to what you see in the "Sample Run".
  2. Style information should be in a css file named **research.css**.
- **Server Processing**
  1. The form data will be sent to the researchProcessing.php script.
  2. You will use **get** as the submission method.
  3. The names to use for form fields are: firstname, lastname, phoneFirstPart, phoneSecondPart, phoneThirdPart, email, age, heightFeet, heightInches, weight, highBloodPressure, diabetes,

glaucoma, asthma, none, period, studyType, firstFourDigits, secondFourDigits (these last two correspond to the study id), and comments.

4. The link php script request provides an example of submitting data to the script. From this example you can verify the names to use for form entries. Notice that to execute this script you need your web server running and you need to place the DataValidation folder in htdocs.
5. Make sure that you define the action field in your form as follows: **action="researchProcessing.php"**
6. **Do not post your project in Terpconnect.**

Specifications (Drawing Device Recorder)

You need to define an interface that allow us to record and play the activity associated with the drawing application presented in class (DrawingPointer.html). For example, someone using your interface can select a record button, draw a circle, stop the recording process, and then playback the drawing of that circle. For this part of the project:

- You must define a JavaScript "class" that represents the recorder. The class will define functions that control each of the following activities:
  - Start recording
  - Stop recording
  - Play
  - Clear Screen

  Feel free to add any other methods or instance variables/variables you understand you need.
- Your interface must provide access to the functionality specified above (e.g., start recording, stop recording, etc.) It is up to you do decide how to provide access to the functionality. One approach is to used buttons or by using the keyboard.
- Your interface must define a help section that describes how to use the recorder.
- As mentioned in class, you can keep track of mouse activity by using an array that stores the x,y coordinates visited.
- You may not use any program/code from the web in order to complete your recorder.
- You may not use JQuery.
- Feel free to add any other extra functionality. Surprise us!
- Your HMTL code must validate.
- Place all the files associated with the drawing device recorder in a folder called Recorder.
- Use Recorder.html as the name of your application.
- **Do not post your recorder on the web (even after the class is over).**

Additional Requirements

- Your project must work using Chrome.
- You must use good indentation.
- You must use meaningful variable names.
- Your html must validate according to http://validator.w3.org/#validate-by-upload
- You must use the **"use strict"** pragma directive for your JavaScript code.

Sample Run (Data Validation)

The data submission video illustrates the functionality associated with the data validation form.

Submission

Submit your project using the appropriate submit server entry.

Academic Integrity

Please make sure you read the academic integrity section of the syllabus so you understand what is permissible in our assignments. We want to remind you that we check your assignment against other students' assignments and any case of academic dishonesty will be referred to the University's Office of Student Conduct.