Ross Studtman

CIS131 – Web Development 2

Project 4: Team Roster

Shane May

12/06/2012

Problem Scope

Program provides for users to select a sport (baseball, basketball, or football), enter a team name, enter a player name, and select the player’s position from a drop down menu. User’s selections for team name, player, and player’s position should be displayed.

Functionality

User clicks “sport” button which dynamically populates the select drop down options with sport appropriate positions. Dynamic population of select element’s options occurs via “.getJSON()” and loads a .json file.

User enters team name (form text field) which is logged to an HTML display when “Set” button clicked.

User enters player name (form text field) and selects player position (form select options) which are logged to HTML display when “Add” button clicked.

Each click of the “Add” button should remove the player name from text field and selected player position from select options. Page focus should return to player text field.

User can click a “Reset” button to reset all fields to original states and clear all displays.

Pseudo-Code

Program loads:

* Focus moved to first form text field
* Event handlers created for buttons (baseball, basketball, football, Offensive Line, Defensive Line, and Reset).
* Default sport selected (baseball) and form’s select options populated.

User clicks a sport button:

* If user doesn’t click a sport button: default to “baseball” as sport selection.
* If baseball or basketball, load .json file to populate select options.
* If football, reveal buttons “Offensive Line” and “Defensive Line”.
* User clicks Offensive Line or Defensive Line button:
  + Load appropriate .json file to populate select options.

User enters team name into text field and clicks “Set” button:

* Gather data from text field and display in HTML.

User enters player name into text field. User selects player position from form’s select options. User clicks “Add” button:

* Gather data from text field and chosen select option and display in HTML.
* Remove selected options from form select.
* Return focus to player name text field.
* If select options become depleted hide or remove the select element and “Add” button from being displayed.
  + Display message to user that options are gone.

User clicks any button:

* Redisplay select and “Add” button.
* Hide display message that indicated no more options.
* Reset HTML displays.
* Return focus to appropriate form field.



Operational Requirements

Javascript written with jQuery library 1.8.3 minified. Program requires modern browser capable of correctly evaluating JQuery. Program’s Javascript, HTML, CSS, and jQuery tested in Mozilla Firefox version 16.0.2.

Program requires the following to exist in the same folder:

* baseball.json
* basketball.json
* footballDL.json
* footballOL.json
* jquery-1.8.3.min.js
* Project4\_teamRoster.css
* Project4\_teamRoster.html
* Project4\_teamRoster.js

Known Bugs or Limitations

Program runs as expected. Note, for programmers seeking to add functionality to this program, the “selectOptions” function should not be invoked after the initial invocation by jQuery’s .ready function, doing so will result in duplicate event handlers being created. A possible solution to this issue would be to create a Boolean that allows only one run through the code that sets the event handlers; or, the event handers could be moved back into the jQuery .ready function.

Currently the user does not have to enter a player name and is allowed to select a position and add this to the HTML display, sans player name. This could be viewed as a feature or a limitation depending on interpretation.

Testing

Testing involved selecting each sport button to determine if the appropriate .json file was loaded into the select’s options. Initially there was an error where the option’s list would double in size after “football” and either “Offensive Line” or “Defensive Line” were clicked; this error was tracked down to erroneous duplication of event handlers and was fixed.

Alphanumerics were added to text fields and buttons were clicked in a variety of orders to determine functionality.

Alerts (alert(‘hey’)) were added in several functions to test the number of times the function was being ran with each click. This testing revealed that even after the duplicate options bug in the select element was fixed some functions were being run twice. The problem was fixed when the duplication of event handlers was discovered.