**Register Attendance button JavaScript and AJAX.**

Below is documentation of the steps I implemented to allow the user of the application to register the attendance of a guest within the Guest management tab within the Events Menu. Although this is not a finished solution I will walk through the steps I have taken to get to this point and the problems that still need to be addressed.

Table

Description automatically generated

Firstly the “Guest” view within the Events view folder will dynamically display “Register Attendance” checkboxes for the number of Guests that are currently assigned to an event. Note that the value of each checkbox will be assigned its respective GuestID. This HTML page has directions to a JavaSCript script within the wwwroot/js titled RegisterAttendance.js.

Text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

When the document is ready, and a checkbox is clicked, a function named validateCheckboxes will be called.

Text

Description automatically generated

This function store the checkboxes in a group, this will be iterated over and the values of each checkbox that should correspond to the GuestID are stored in an array.

I then use an ajax POST request to the web service and specifically the UpdateAttendance method.

Text

Description automatically generated

The UpdateAttendance method (Events/Controllers/GuestController)will take the array of checkbox values, the respective GuestID’s will be retrieved from the database and each guest entity will have their “Attended” value set to true.

Text

Description automatically generated

Unfortunately, as aforementioned I was unable to get this whole process to work. I feel an issue I have is the way the checkboxes are identified as the number of checkboxes required is dynamic, it would not be possible to give them unique names for an unknown number of boxes. One way to solve this may be to have a single button at the bottom of the HTML page that will collate all of the checkboxes on the page, validate each one and then send this array of values to the method. Given more time, this is a solution that I would investigate further.

Talk about the misunderstanding of implementing catering into the MVC but then also the choice of using the service layer, DTO on both sides.

**Catering API service layer.**

To implement the catering service API features into the MVC application I have created a service layer within the Events application that handles GET and POST requests to the API. The layer handles sending and receiving objects in their DTO form and also the transforming of DTO objects into a view model object that is suitable to be displayed in the MVC application. The idea of the service layer seeks to facilitate the slimming down of the methods within the Catering Controller in the Events side of the application.

Through research I have found that DTO’s on both client and server side may not be needed and outside of this assignment, using AutoMapper would be a much better solution to handling DTO’s.

<https://www.c-sharpcorner.com/article/using-automapper-in-mvc-project/>