CAPSTONE SUBMISSION FORM

Please complete the form below when you submit your capstone. It will help your instructor understand your code and give you all the points you deserve! Email it back to your instructor Thanks!

**WHAT IS MY CAPSTONE ALL ABOUT**

1) Can you provide a brief overview of your project and its theme?

My project is an administrative tool that will manage groups of students that will learn piano, drums, or guitar at a musical arts school.

2) How would a user (me) use your project? Can you provide a short "user manual" or instructions to the features you have included?

The user will be able to organize students in classes within the categories of Piano, Drums, or Guitar. Almost everything is clickable so the user can select any category to view the available classes offered.

There is an ‘Add New Class’ button at the end of the list of available classes. If you click on a class, it will populate the ‘Class Details Page’ where you can view and edit all group information, including the currently enrolled students.

You can then click on any of them to view and edit that student’s information. Like the groups page, there is a ‘Add New Student’ button at the end of the list of students.

3) Do I need a username to interact with your program? If so, what username and password can I use to get started? Are there users with special privileges? How does that work?

There is no username required for this app at this time.

**WHAT DO I NEED TO UNDERSTAND ABOUT THE SERVER**

1) Can you explain how your data relates to "organizations", "groups" and "members"?

Organizations are the categories: Piano, Drums, and Guitar.

Groups are Music Classes.

Members are Students Enrolled in a Class.

2) Did you clean up your data and reset your counters.json file?

no

3) Did you make changes to server.js other than name mappings? If so, what were the changes? (include a list of any new API methods)

n/a

**GETTING ALL YOUR POINTS**

1) Did you implement each of the following?

yes— A "Home Page" that includes a description of your "theme" and an attractive list of the "organizations" in the system

no—The ability to search for and view a list of groups based on criteria that makes sense for your theme. Include a "view all groups" option

I did not implement any search functionality. I had a ‘view all groups’ option but then replaced it with ‘viewing groups by category’ for a better mobile experience.

yes—The ability to register a "group"

In my project, this means: Creating a new music class.

yes—The ability to add an "item" to the "group"

In my project, this means: Adding students to a class.

yes—The ability to view a particular "group's" details and the "items" in that "group"

In my project, this means: Displaying all class information including a list of the students within that class and their information.

yes—The ability to edit an item's information (as appropriate)

In my project, this means: Updating the student’s information.

2) What techniques did you use to display error messages that users enter bad data in forms or when your Ajax requests return errors?

I used form checking to show a message if a field is touched but still empty.

I used console.log(error) within observables to alert the developer of any issues.

I used alert boxes to notify the user of a bad request to the server.

3) Do you have:

yes—Good HTML and CSS (readable, commented, no validation errors, etc.)

n/a—Linted server.js and all of your site's script files

no—Readable, commented code with proper names, good use of functions, etc

yes—Use of GitHub Repo with appropriate branches, commit history / comments, and an   
***informative*** README.md file

4) What innovative feature that we didn't discuss in class (new Bootstrap or Angular feature, etc.) did you incorporate? How did you learn about it? What made you decide to use it?

Using local storage to make data survive a refresh.

I learned about it by doing some research on how to restore data after a refresh. I was having issues with my class/student information not surviving when the form was refreshed.

5) Who are some of the people you collaborated with on this project and how did they help you or how did you help them?

Chris and I worked together to figure out a type error we both were having. We got it to work with an ngIf but then figured out that we could use the elvis operator instead.

Jonathan and I worked together to think of a solution to the disappearing data in a form on refresh. We eventually decided that the issue was that the service properties and component are both wiped out on refresh. The solution was to store that data on local storage and retrieve the data before trying to reinitialize the component.

6) Did you implement any of the following features to earn bonus points? Elaborate!

yes—The ability to edit a "group's" details - for example, to change a sponsor's name and/or email

In my project, this means: Updating a class to have a new teacher.

yes—The ability to delete a "group"

In my project, this means: Deleting a class.

yes—The ability to delete an "item"

In my project, this means: removing a student from a class.

no—The ability to register a user / logon to the site

In my project, I use the users to: n/a

yes—Wild card feature - If so, tell me more about it! What does it do? Why is it cool? What did you learn implementing it?

My wild card feature is in the styling. Since this app is for an admin of a small music school, I created this app with a floating flexible feel for a good mobile experience.

**SUMMARY**

1) What else do I need to know about this project?

It uses a flat file structure to avoid complexity as suggested by the John Papa style guide.

2) Comments?

Thanks for all your instruction! You did an awesome job getting down on everyone’s level and carrying this whole class to new heights of knowledge!