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 app for employees of a fishing charter business. There will be multiple types of fishing charters which will be the ‘organizations’, multiple charters (or groups) that can serve a max number of people who booked the charter (or members).**

**Overall, the app will allow employees of the business to manage all of the charters (add/edit charters, add/edit members to the charter). If time permits, I want there to be two types of users: admins and normal users. Admins would be the employees who will be able to manage everything, normal users will be able to book (add themselves) to available charters.**

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

**Once logged in you will land on the home page. From there you can find specific types of charters by selecting the panels, or go to Charters in the nav bar.**

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?

**Yes, a good test user is: username/password --- farmgirl/farmgirl respectively. Alternatively, you can register a new user, currently there are no restrictions on that, so anyone can create a login. There are no special privileges at the time of writing this.**

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

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

**Organizations: Type of fishing (ex: Deep Sea, River, Lake, etc.)**

**Groups: The different charters available for a given org (contains charter name, guide name/contact)**

**Members: Individuals that have booked a charter**

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

**I have made sure that it lines up with the amount of data there now.**

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)

**Nope, no changes to the BE were made.**

**GETTING ALL YOUR POINTS**

1) Did you implement each of the following?

\_\_X\_\_ A "Home Page” that includes a description of your "theme" and an attractive list of the "organizations" in the system

\_\_X\_\_ 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

\_\_X\_\_ The ability to register a "group"

In my project, this means: \_\_\_**Ability to add a charter to a specific type of fishing**\_\_

\_\_X\_\_ The ability to add an "item" to the "group"

In my project, this means: \_**A member registers for a specific charter (group)**\_

\_\_X\_\_ The ability to view a particular "group's" details and the "items" in that "group"

In my project, this means: \_\_\_**A group’s details consist of charter name/name of guide, items in my case would be members that have signed up for the specific charter under question**\_\_\_

\_\_X\_\_ The ability to edit an item's information (as appropriate)

In my project, this means: \_\_\_**Ability to edit info pertaining to any members who have signed up for a charter.** \_\_\_

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 implemented a component that handled showing toast messages, used that for some errors, but also on forms just used properties of the form to show errors during the process of filling out the form.**

**Not all of my requests have error handling though, so definitely a todo.**

3) Do you have:

\_\_X\_\_ Good HTML and CSS (readable, commented, no validation errors, etc.)

\_\_X\_\_ Linted server.js and all of your site's js files

\_\_X\_\_ Readable, commented JavaScript code with good names, good use of functions, etc

\*Just a note on this one, did not write comments out of force of habit. But I like to think it was clean code :)

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

\*Did not use different branches during development.

4) What innovative feature (new Bootstrap feature, cool JavaScript or jQuery function, custom API method, etc.) did you incorporate? How did you learn about it? What made you decide to use it?

**Made extensive use of RxJS’s Subjects. Had used them before, but never in this magnitude. Really opened my eyes to how to use them. That is how I wrote my own toast service and how I stored data so that the async pipe would update data as soon as it changed.**

**Async pipes were also something that I had not used outside of the brief intro to them in class. Super cool to learn about and threw a spin on how I understand using data in components.**

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?

**Seemed to be a bit of everyone in and out of breakout rooms. I tried to help the best I could when I was available if anyone had any issues.**

**Callout to a teammate on giving me the idea to pipe the async pipes to implement filter functionality**

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

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

In my project, this means: \_\_\_**Able to change the charter’s guide name, phone, email, and change the group name (really any property of the group)**\_\_\_\_

\_\_X\_\_ The ability to delete a "group"

In my project, this means: \_\_\_\_**Ability to delete a charter trip**\_\_\_\_\_\_\_\_

\_\_X\_\_ The ability to delete an "item"

In my project, this means: \_\_\_\_\_\_\_\_**Ability to delete and remove a member of a charter**\_\_\_\_\_

\_\_X\_\_ The ability to register a user / logon to the site

In my project, I use the users to: \_\_\_\_**Register and login to an account so that not just anyone can go and edit/add charters/members. (though currently anyone can register)** \_\_\_\_

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

**Was pretty proud of the Toast Message solution. Not the most elegant, but learned a ton about RxJS and really the power of it. At the end I could inject a MessageService that I wrote and then call a method to show a toast and it showed.**

**Also, again Async pipes on all of the data was eye opening to how I approach Angular components as a whole. You can eliminate a lot of code by using those pipes, once understood, much simpler to implement in my opinion.**

**SUMMARY**

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

I spent a lot of my free time on it because I enjoyed working / learning from it.

2) Comments?

Really enjoyed the whole process. I learn from experience and this being the first start to finish Angular project that I have created, I learned a TON. This whole bootcamp was super beneficial for me, I really feel like I am beginning to understand web dev at a deeper level. Getting rid of that imposter syndrome :)

Thanks a ton!