# **Angular Lab - Part 2**

At the end of the first mission planner studio, multiple components display data about the mission. Your job is to allow the user to update the mission plan by adding user interaction.

## **1. Getting Started**

For this lab and the next, you will use some starter code from the GitHub repository.

1. Unzip the downloaded starter project in this folder and move it to the **angular\_practice** directory you are using for your Lab Assignments.
2. Run npm install on the root directory of the project.

**Warning**

Initializing a new Angular project inside of another one creates version control complications that are best avoided.

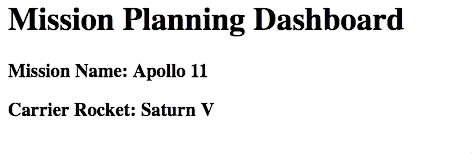
1. Before running the **npm install** command in the terminal to download dependencies, make sure you are NOT inside a current Angular project!
2. Ignore any errors regarding the python executable.
3. Run **ng serve** to build and serve the project.

## **2. Review the Starter Code**

The starter code for this studio is similar to the solution yoy coded for the first mission planner studio, but with a few notable changes.

### **2.1. Editable Mission Name**

The mission name can now be edited by clicking on the text, changing the text in the input box, and then updated by clicking save or pressing the enter key. Review the code in **src/app/header/header.component.html** and **src/app/header/header.component.ts** to see how this feature was implemented.



### **2.2. Crew Array of Objects**

Open **src/app/crew/crew.component.ts** in VSCode. Notice on line 10 that a crew array is defined. This array of objects will be used to display the crew. Each crew member has a **name** and **firstMission** property. If **firstMission** is **true**, it means this is the first mission for that person.



## **2.3 Requirements**

**Note**

All of these features only temporarily alter the data. If you refresh the page, the original data will reappear.

### **3.1. Edit Rocket Name**

The rocket name should be clickable and editable like the mission name. Alter **src/app/header/header.component.html** and **src/app/header/header.component.ts** to allow the user to edit the rocket name.

### **3.2. Use \*ngFor to Display Crew**

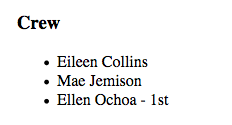
Replace the static list of **<li>** tags in **src/app/crew/crew.component.html** with an **\*ngFor** that loops over the **crew** array.

Add this code to **src/app/crew/crew.component.html**.



### **3.3 Display 1st Mission Status**

If a crew member's **firstMission** property is **true**, then display the text "- 1st" next to their name.



*Example of first mission status being shown.[¶](https://education.launchcode.org/intro-to-professional-web-dev/chapters/angular-lsn2/studio.html" \l "id2" \o "Permalink to this image)*

Add this code right after the member name in **src/app/crew/crew.component.html**.

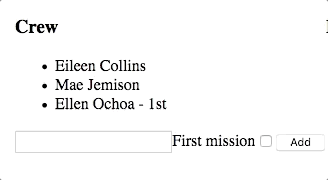


### **3.4. Add Crew Members**

Allow crew members to be added to the list. To create a new crew member, two pieces of information are required:

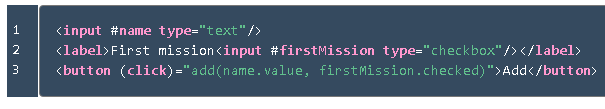
1. crew member's name
2. the first mission status

We will use an input box and a checkbox to collect the data.



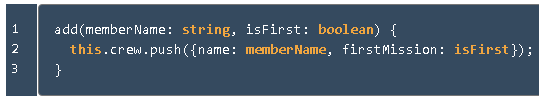
*Example of crew member being added.*

Add this code to the bottom of **src/app/crew/crew.component.html**.



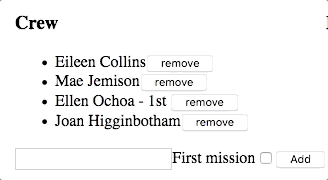
Line 1 creates an input that declares the local variable **name**. Line 2 defines a checkbox that declares the **firstMission** variable. Line 3 creates a button that, when clicked, sends the new **name** and **checkbox** value to the **add** function. This function adds the new crew member to the roster!

In the **src/app/crew/crew.component.ts** file, include this code for the **add** function:



### **Remove Crew Members**

Allow removing of crew members by adding a button next to each person in the crew list. When the remove button is clicked, the **remove** function in the crew component will be called, which deletes that person from the crew array.

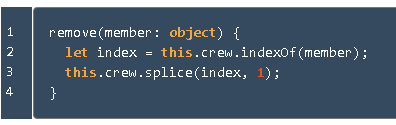


*Example of crew member being removed.*

Add line 4 to file **src/app/crew/crew.component.html**. Be sure to put it before the closing **</li>**, so that the button appears next to each item in the crew list.



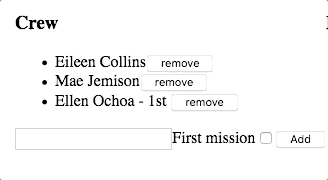
Add the **remove** function shown below to the crew component in the **src/app/crew/crew.component.ts** file.



### **3.6. Edit Crew Members**

Finally we are going to allow the user to edit crew members who have already been added.

1. If the crew member name is clicked, then their name should be replaced with a text input and a save button.
2. When save is clicked, the input and save button are replaced by the text-only version of the name.
3. Only one crew member can be edited at a time.



*Example of crew member name being edited.*

We need to add a click event to the member name.

1. Put **{{member.name}}** inside of a **<span>** that has a **(click)** handler.
2. Make the **<li>** in **src/app/crew/crew.component.html** look like the code below.

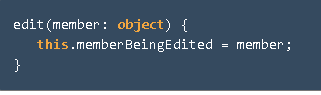


We need a way of knowing which crew is being edited.

1. Add this property to the crew component in file **src/app/crew/crew.component.ts**. The property **memberBeingEdited** represents the crew member who is currently being edited.



1. Next add a **edit** function to the crew component file **src/app/crew/crew.component.ts**. This function will set a **memberBeingEdited** variable to be equal to the crew member who was clicked.



Now we need to add an **\*ngIf** that will show the two versions of the member, the display state or the edit state.

1. In the edit state, an input box with a save button will appear, but for now the input and save won't have any functionality. Make your **src/app/crew/crew.component.html** file look like the below code.



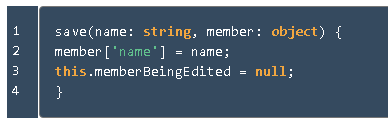
Finally, we are going to make the edit state update the member name when save is clicked.

1. Update the **<input>** and **<button>** tags to look like:



The last step is to add the **save** function to the crew component. This function will be called when the **<button>** is clicked or when the enter key is pressed and the **<input>** has focus.

1. Add the below **save** function to the crew component.



## **Bonus Missions**

1. Don't allow duplicate names to be added to the crew.
2. Allow user to add equipment.
3. Allow the user to edit equipment.
4. Allow the user to remove equipment.
5. Allow user to add experiments.
6. Allow the user to edit experiments.
7. Allow the user to remove experiments.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***