# Angular Lab, Part 1

In this lab, you will practice the Angular file structure, templates, and components. In future labs, you will build a Mission Planning Dashboard using your Angular skills.

## 1. Mission Planning Dashboard

A useful and common front end application is a dashboard. It shows a summary of information about a topic, helping users of the web app make informed decisions.

You will create a Space Mission Planning Dashboard.

## 2. Create Angular Project

1. Launch Visual Studio Code. Create an **angular\_practice** folder.
2. Open a terminal at the root of your **angular\_practice** folder.
3. Create a new Angular project by running **ng new angular-studio-part1**.

* When prompted about using routing, enter "N" for No.
* When prompted to select the stylesheet format, select CSS.

$ ng new angular-studio-part1

? Would you like to add Angular routing? No

? Which stylesheet format would you like to use? CSS

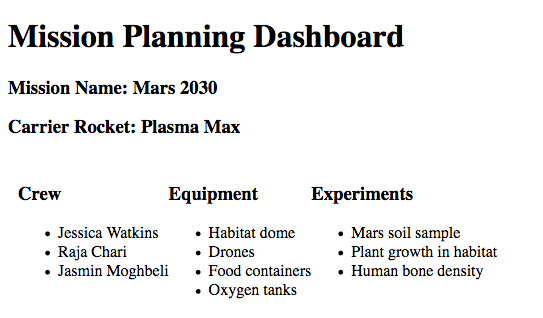
1. When the process finishes, use the File menu to open angular-studio-part1 in VS Code.
2. In the terminal, use **pwd** to check your position in the file tree. If necessary, navigate into the **angular-studio-part1** directory.
3. Install dependencies by running **npm install**
4. Verify that the application will run by running **ng serve**
5. View the site in your browser at [http://localhost:4200](http://localhost:4200/)
6. You should see a header that says "Welcome to angular-studio-part1!"
7. Stage and commit the files before starting on the features.

**Tip**

You will likely still have **ng serve** running in your terminal. You can stop **ng serve** by pressing the keys "control+c", or you can open an additional terminal window to run the git commands.

## 3. Requirements

The mission dashboard you are creating will eventually look like this.



### 3.1. Update Starter Page Content

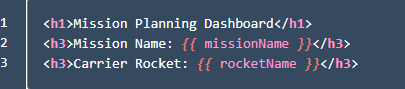
The default stater page created by Angular contains default text, images, and links. Your job is to remove the default content.

1. Clear out the contents of file **app.component.html**.
2. Type in the text **Add components here...** into **app.component.html**.
3. Run **ng serve** if it's not already running.
4. View the app in your browser to verify the words "Add components here..." is the only thing that appears on the page.

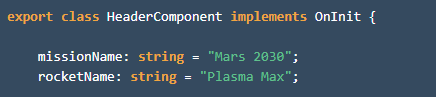
### 3.2. Header Component

You need to create a component that shows the title, mission name, and carrier rocket.

1. In terminal navigate to the folder **src/app** folder.
2. Create a header component using **ng g component header**.
3. In the file **header.component.html** add HTML:



1. Add the variables missionName and rocketName to the header component in header.component.ts



1. Add a reference to the header component in **app.component.html**.



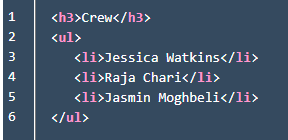
1. View the app in your browser to verify that the title, mission name, and rocket name are visible.



### 3.3. Crew Component

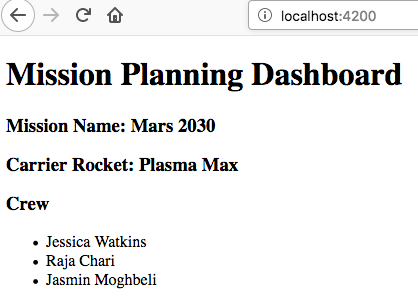
Next you need to make a component to show a list of crew members.

1. Create the component by running **ng g component crew**.
2. Set the contents of **crew.component.html** to be:



1. Add a reference to the header component in **app.component.html**.

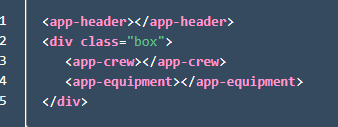




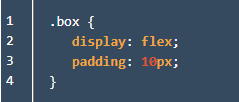
### 3.4. Equipment Component

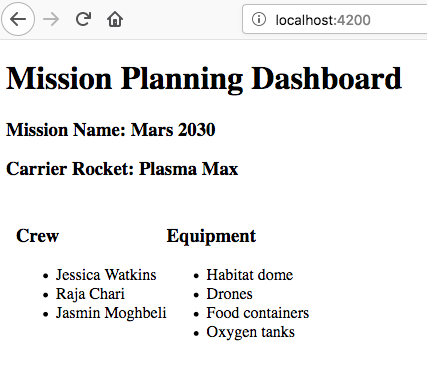
Now you need to create a component to show a list of equipment.

1. Create an equipment component named **equipment**.
2. The component should display the following:
   1. An **<h3>** that contains "Equipment"
   2. A **<ul>** that contains **<li>** for: Habitat dome, Drones, Food containers, Oxygen tanks
3. Add the equipment component to app.component.html using the HTML below. Notice the <div> surrounding the crew and equipment components.



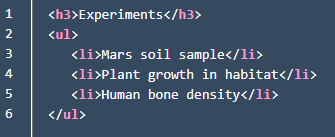
1. Add CSS to file **app.component.css** to horizontally align the crew and equipment lists. Without this CSS, the equipment list will appear below the crew list.





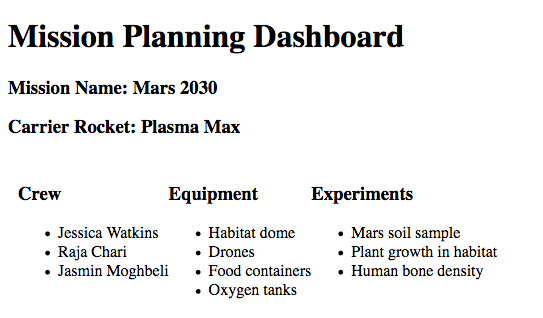
### 3.5. Experiments Component

1. Finally, add an experiments component that contains the HTML below:



1. Make the list of experiments show up to the right of equipment list.

When done your dashboard should look like this:



## 4. Commit Your Work

Be sure to stage and commit your changes!

1. Verify the branch and status of the files.
2. Commit your changes.
3. **Optional**: Create a new repository in your GitHub account, then push your commits to origin.

You will make different versions of the mission planning dashboard in the next two studios.

## 5. Bonus Mission

1. Display crew members by adding an array of crew names.
   1. In **crew.component.ts** add **crew: string[] = ["Jessica Watkins", "Raja Chari", "Jasmin Moghbeli"];**
   2. In **crew.component.html** use references like **<li>{{crew[0]}}</li>** to display the crew names.
2. Use CSS to add different colors, fonts, borders, etc. to your dashboard.
3. Move the components around to see how that affects the display of the data.

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