**Assignment #5: Launch Checklist Form**

Using our knowledge of forms, the DOM, and HTTP, the commanders of our favorite space shuttle program asked us to create a quick launch checklist. We have four fields that need to be filled out with vital information: the pilot’s name, the co-pilot’s name, the fuel levels, and the mass of the cargo.

Our pilot, Chris, and the co-pilot, Blake, have been hard at work securing the cargo and filling the shuttle tank. All we need to do is use validation to ensure that we have all of the info for the space shuttle program and make sure no one prematurely launches the shuttle.

**Requirements**

Create a Launch Checklist Form for astronauts to fill out in preparation for launch. The form should do the following:

1. Validate the user responses with **preventDefault()** to ensure the following:
   1. The user entered something for every field.
   2. The user entered text for names and numbers for fuel and cargo levels.
2. With validation, update a list of what is currently ready or not ready for the shuttle launch.
3. Indicate what is good or bad about the shuttle and whether it is ready for launch by using the DOM to update the CSS.
4. Fetch some planetary JSON to update the mission destination with vital facts and figures about where the shuttle is headed.

**Setting Up Your Project Repository**

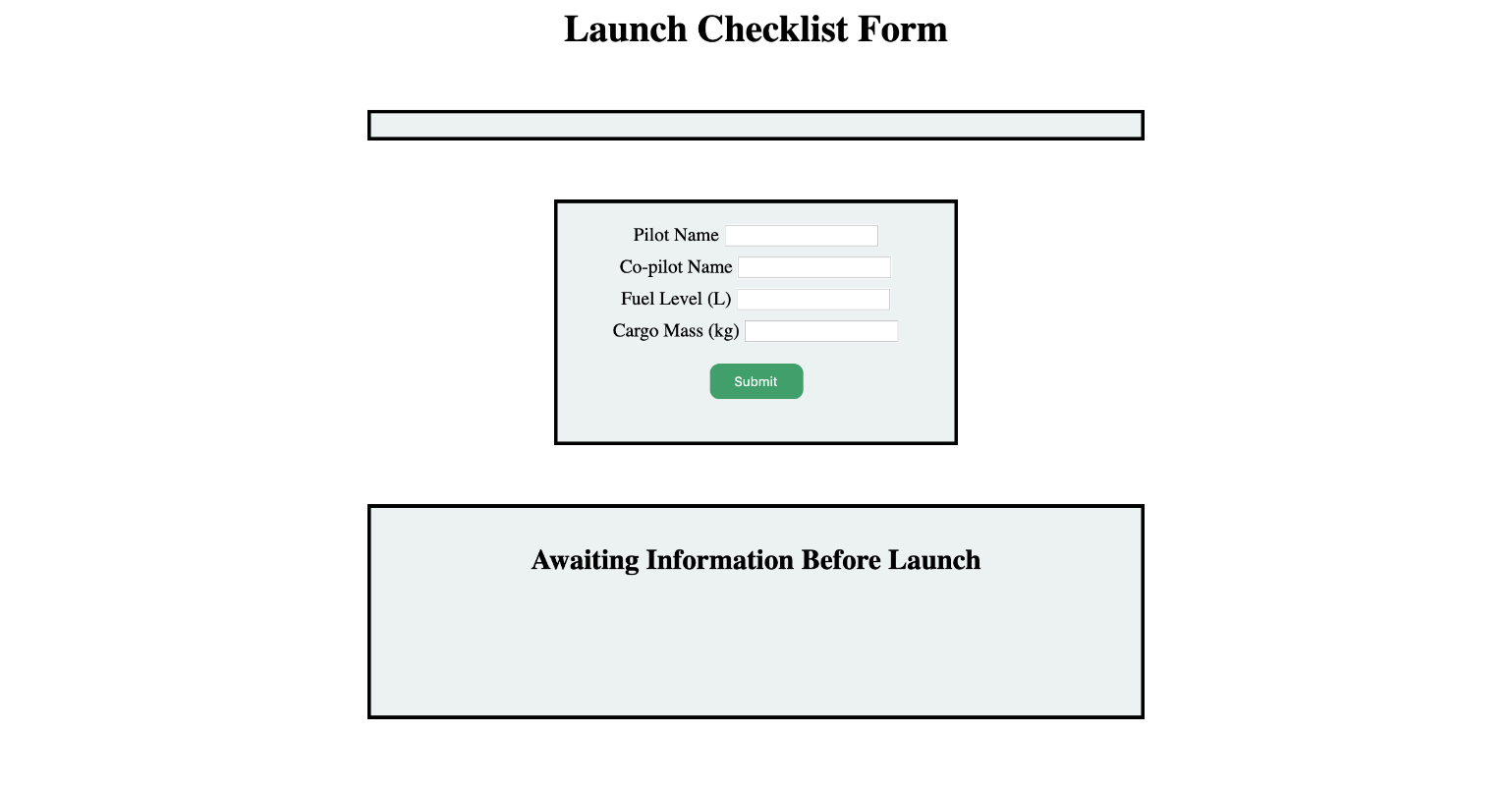
Fork the repository with the [starter code](https://github.com/LaunchCodeEducation/Launch-Checklist-Form/) to your personal Github profile and clone the repository to the directory where you are keeping your assignments for the class.

**Warning**

When updating styles to indicate whether the shuttle is ready for launch, do NOT modify **styles.css**!

To get started, navigate to the directory with your copy of the starter code. Open **index.html** with Firefox to verify that your starter code is working.

When you open **index.html**, you should see the Launch Checklist form with a rectangle above it for the mission destination and a rectangle below it that simply says “Awaiting Information Before Launch”.



**Adding Validation**

**Adding Alerts**

First, let’s add validation to notify the user if they forgot to enter a value for any one of the fields.

This process is going to look similar to the [validation section](https://education.launchcode.org/intro-to-professional-web-dev/chapters/forms/validation-with-javascript.html#javascript-validation) in the chapter on forms. Make sure to use **preventDefault()** and an alert to notify the user that all fields are required.

You also want to make sure that the user entered valid info for each of the fields. Valid information for the fields means that the user submits a value that is easily converted to the correct data type for our fellow engineers. The pilot and co-pilot names should be strings and the fuel level and cargo mass should be numbers.

**Note**

If you want to check if something is **NaN**, you cannot use **==** or **===**. Instead, JavaScript has a built-in method called **isNaN(value)** that returns **true** if **value** is **NaN** and **false** if **value** is not **NaN**.

**Updating Shuttle Requirements**

The list of shuttle requirements, the **div** with the id **faultyItems**, should be updated if something is not ready for launch. Using template literals, update the **li** elements **pilotStatus** and **copilotStatus** to include the pilot’s name and the co-pilot’s name.

If the user submits a fuel level that is too low (less than 10,000 liters), change **faultyItems** to **visible** with an updated fuel status stating that there is not enough fuel for the journey. The text of the **h2** element, **launchStatus**, should also change to “Shuttle not ready for launch” and the **color** should change to red.

If the user submits a cargo mass that is too large (more than 10,000 kilograms), change the list to **visible** with an updated cargo status stating that there is too much mass for the shuttle to take off. The text of **launchStatus** should also change to “Shuttle not ready for launch” and the **color** should change to red.

If the shuttle is ready to launch, change the text of **launchStatus** to green and display “Shuttle is ready for launch”.

**Fetching Planetary Data**

Finally, we need some JSON to fill in the crew on the mission destination. Our planetary data can be found in [JSON format](https://handlers.education.launchcode.org/static/planets.json). Review the list and decide which planet you want to send our intrepid crew to and make note of the index number.

**Note**

When fetching more than one JSON object, we get an array of all of the JSON objects. In this case, that means an array of our possible mission destinations. When picking the mission destination, just pick the item in the array you want and start counting at 0.

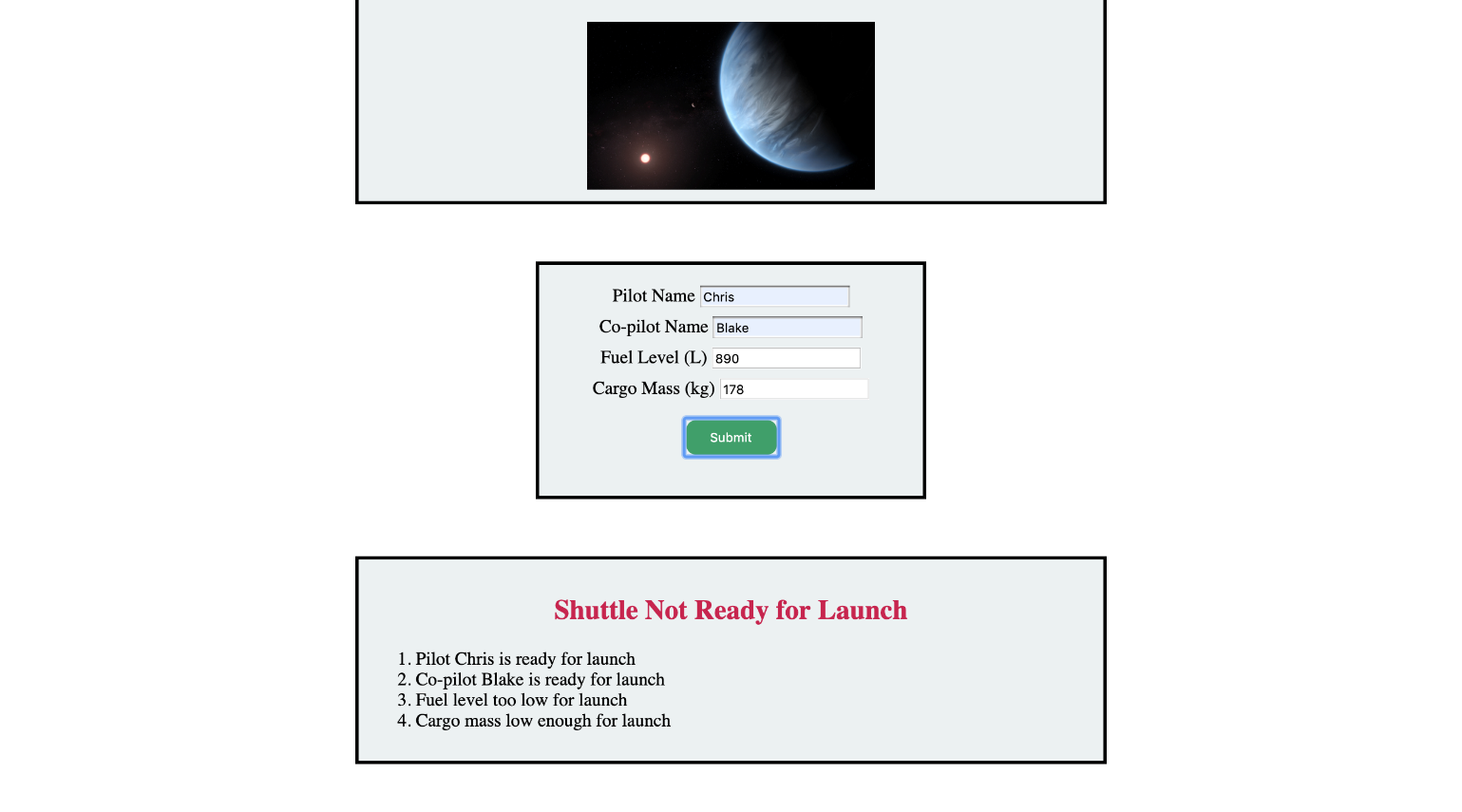
In **script.js**, we have a block of code commented out at the top. This is the format of the **innerHTML** for the **missionTarget** div. Be sure to include the appropriate variables in the template literals!

**The End Result**

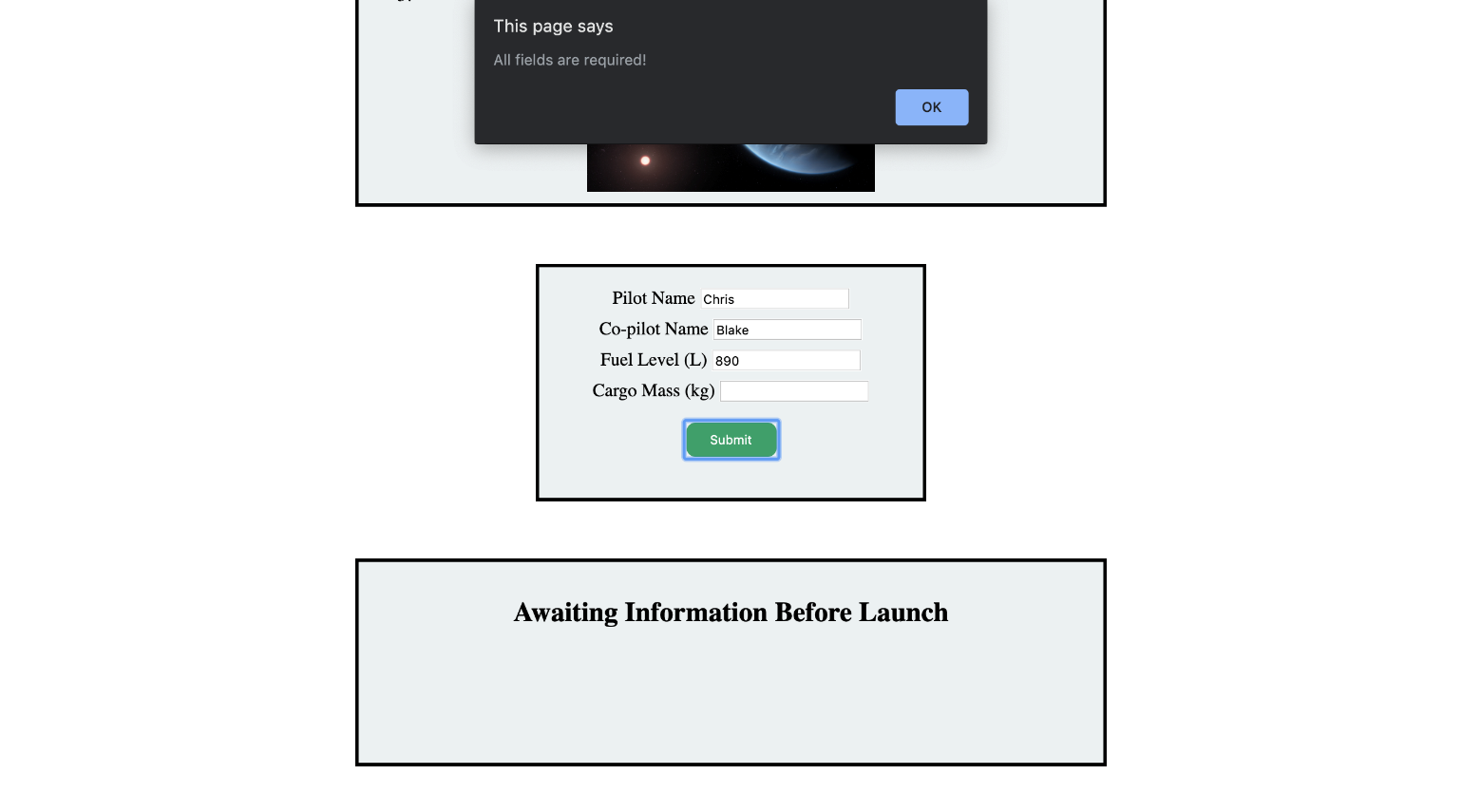
After you implement everything, the following form submission would result in the proper updates to the **launchStatus** and **faultyItems** list.



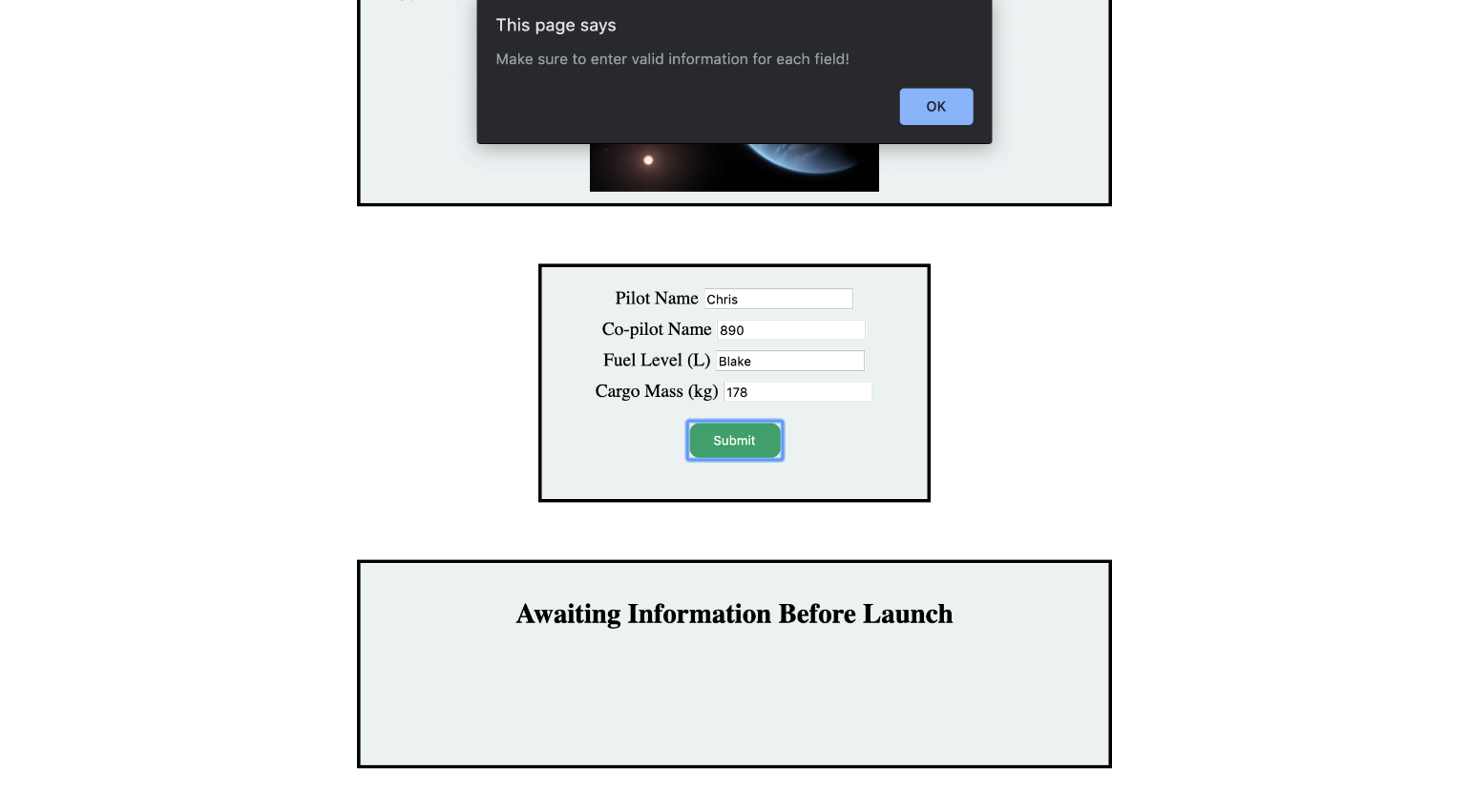
With only 890 liters of fuel, the status of the launch becomes red and states that the shuttle is not ready. The list has also updated to indicate that that is not enough fuel for the shuttle to launch.



If the user forgets to enter the cargo mass, then an alert pops up letting the user know that all fields are required.



If the user switches up the information that needs to go in the fields, then an alert pops up letting the user know that they have tried to enter invalid information.



**Bonus Mission**

Use whichever method you choose to randomly select the mission destination from the available options in the JSON file.