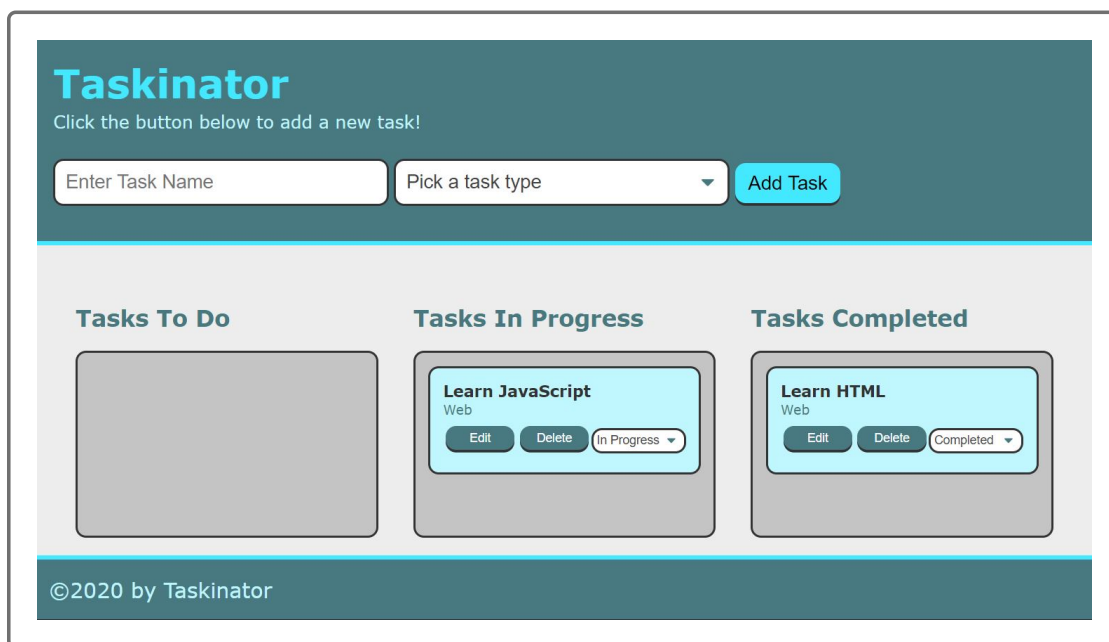


4.3.2 Preview

By the end of this lesson, the Taskinator app will have changed visually to accommodate two new columns, Tasks In Progress and Tasks Completed:



These changes relate to the following features:

- Tasks will now have a status that defaults to "to do."

- Changing the status dropdown will move the task to the appropriate column (To Do, In Progress, or Completed).
- Each task will have an Edit and Delete button.
- Clicking Edit will load the task's information in the form and change the button to say "Save Task".
 - Clicking Save Task will update the task's information in the corresponding column.
- Clicking Delete will remove the task entirely.

Here's the approach that we've outlined for this lesson:

1. Create a new Git branch. We'll create a feature branch that corresponds to the GitHub issue we're addressing in this lesson.
2. Add two new lists in the HTML. We'll add a "Tasks In Progress" and a "Tasks Completed" list to Taskmaster.
3. Apply unique IDs to the tasks. We'll create IDs that uniquely identify each task that's created.
4. Dynamically create task buttons for each task. Now that we have an ID for each task, we can start adding buttons and dropdowns that reference these IDs. Because the tasks and their IDs are dynamically created, these buttons and dropdowns will also be dynamic.
5. Add the ability to delete a task. We'll use event listeners to allow the user to delete tasks.
6. Load the task into the form to be edited. Users might want to edit existing tasks. We'll enable this by making sure the right task data loads in the editing form.
7. Save the edited task. Users will want to save their edits to a task; we'll make sure they can do that.

8. Move the task based on status. If the user changes the status of a task, we'll make sure it is moved to the appropriate list.
9. Save your progress with Git. You did it! It's time to merge your feature branch into the `develop` branch and commit your changes to GitHub.

Note that we added a step to "apply unique IDs to the tasks." Because we'll be updating and deleting tasks based on button clicks, we need some way to know which tasks are being edited. Giving each task an ID will make it easier to find and update or delete them.

We also chose to implement deleting a task before updating a task, because deleting entails fewer steps (straight-up deleting versus loading into the form and then saving). Figuring out the delete process will also give us a better foundation when it comes time to update, as a lot of the logic will be the same.

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.