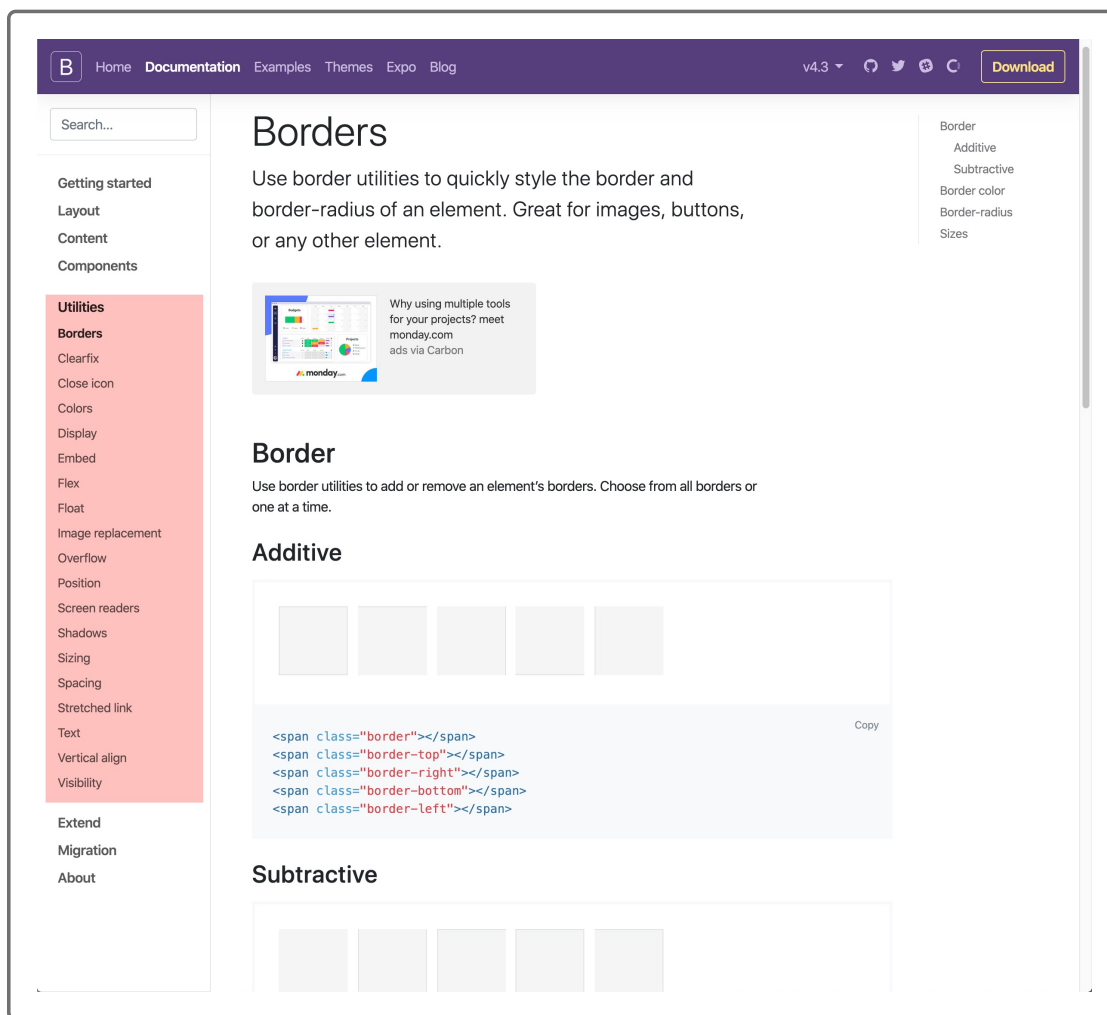


5.2.6 Use Bootstrap Utility Classes

If we look at Bootstrap's documentation, we can see a menu item called Utilities on the left-hand column. When we click on that, the left column will open a submenu with all the different types of utility classes we can use. This includes colors, display, sizing, spacing, and many more (including its own set of flexbox classes). For reference, the following image shows what that section of the documentation looks like:



Let's start in the left-hand column of the page and add some enhancements that will only affect how the app looks on tablets and mobile phones. Narrowing the browser window causes the left column to adjust and stack itself on top of the right column. This change provides the task lists with more space, but the following two complications arise:

- The paragraph in the `<header>` element takes up too much space.
- We lose track of the `<header>` element when we scroll down the page.

Let's use some Bootstrap utility classes to fix these issues, starting with the `<header>` element. Update it to include one more class, called `sticky-top`, so the opening tag looks like this:

```
<header class="col-12 col-lg-3 bg-dark text-light sticky-top d-flex fl
```

There are a lot of classes in place here, all of them from Bootstrap. Two pertain to the responsive grid, and the rest are scattered throughout the Utilities classes explained in the documentation. Here's where we can find them:

- `bg-dark` and `text-light` under [Colors](https://getbootstrap.com/docs/4.3/utilities/colors/)
(<https://getbootstrap.com/docs/4.3/utilities/colors/>)
- `sticky-top` under [Position](https://getbootstrap.com/docs/4.3/utilities/position/)
(<https://getbootstrap.com/docs/4.3/utilities/position/>)
- `d-flex` and `flex-column` under [Flex](https://getbootstrap.com/docs/4.3/utilities/flex/)
(<https://getbootstrap.com/docs/4.3/utilities/flex/>)
- `p-4` and `p-lg-3` under [Spacing](https://getbootstrap.com/docs/4.3/utilities/spacing/)
(<https://getbootstrap.com/docs/4.3/utilities/spacing/>)

Don't worry about memorizing these classes, as long as you know where to find information on how to use them.

Now let's set up that `<p>` element in the `<header>` to only display on larger screens and hide on smaller ones. We'll add classes to it to look like this:

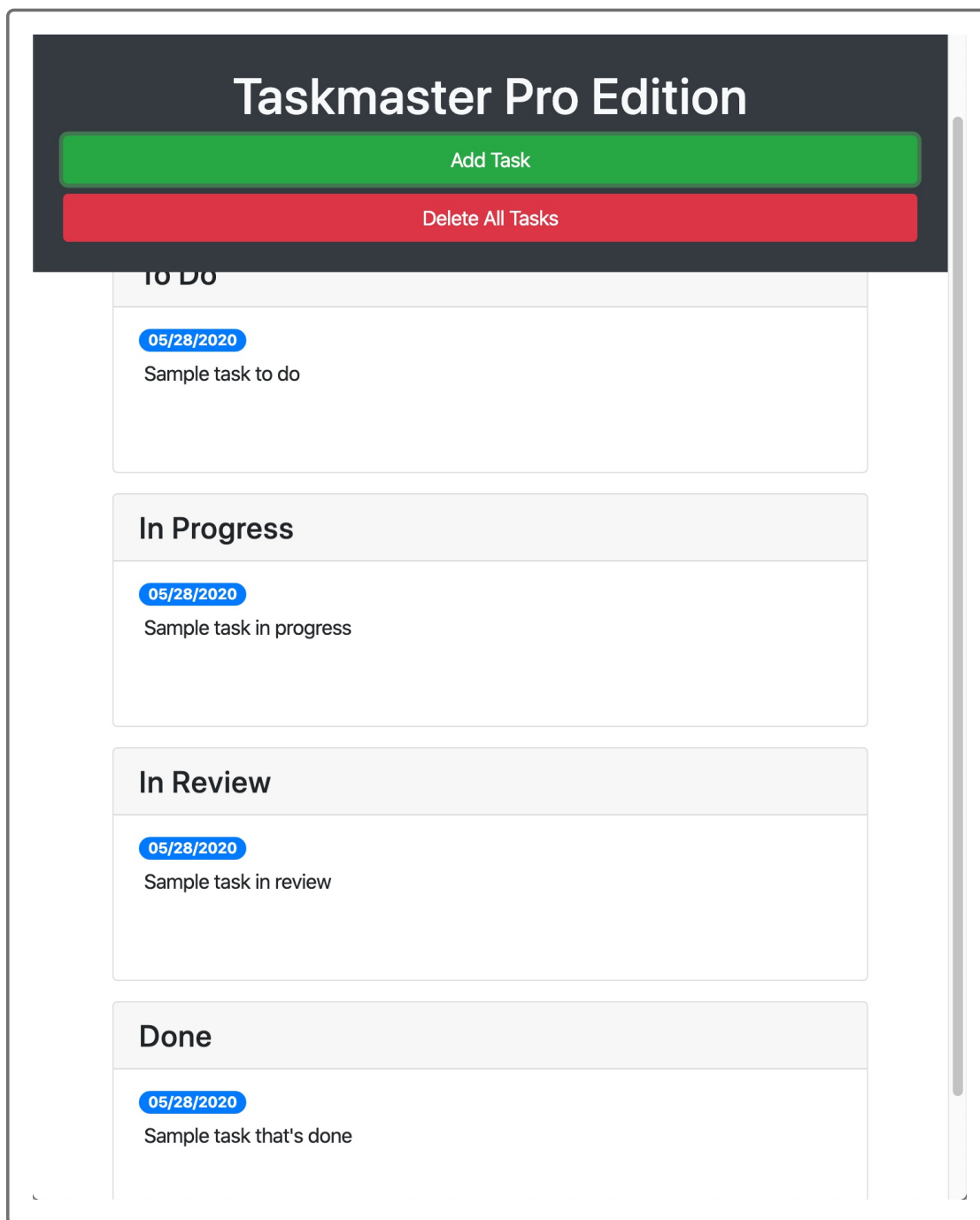
```
<p class="mt-auto d-none d-lg-block">
```

The `mt-auto` class was already in place, and it puts all of the unused space in the `<header>` element on top of the `<p>` element instead of keeping it below. This only works because the `<header>` is set up as a flexbox column.

Here's where we find information about these elements in the Bootstrap documentation:

- `mt-auto` under [Flex](#)
(<https://getbootstrap.com/docs/4.3/utilities/flex/#auto-margins>)
- `d-none d-lg-block` under [Display](#)
(<https://getbootstrap.com/docs/4.3/utilities/display/>)

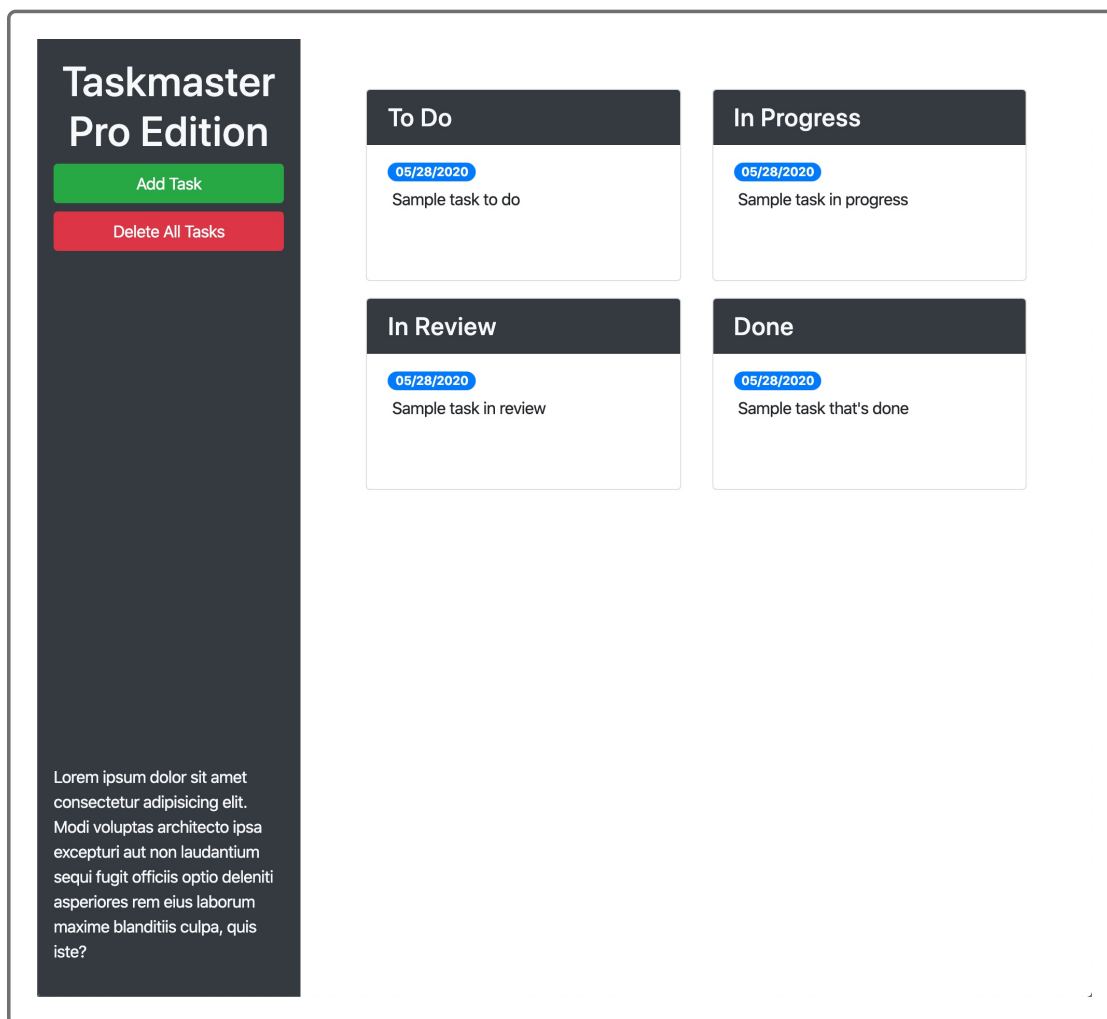
Now if we save `index.html`, refresh the page, and resize the browser, we should see something like this image:



With nothing but Bootstrap utility classes, we really transformed the application's design from desktop to mobile screen sizes!

Add Color to the Cards

Next thing we need to do in order to match the client's design is to add some color to the task list cards, like the mock-up shows us here:



This shouldn't be too hard; we just need to use [Bootstrap's color utility classes](https://getbootstrap.com/docs/4.3/utilities/colors/) [\(https://getbootstrap.com/docs/4.3/utilities/colors/\)](https://getbootstrap.com/docs/4.3/utilities/colors/). Take a moment to review the documentation and guess which classes we'll need to add.

HIDE HINT

We need a class for the background color and a class for the text color.

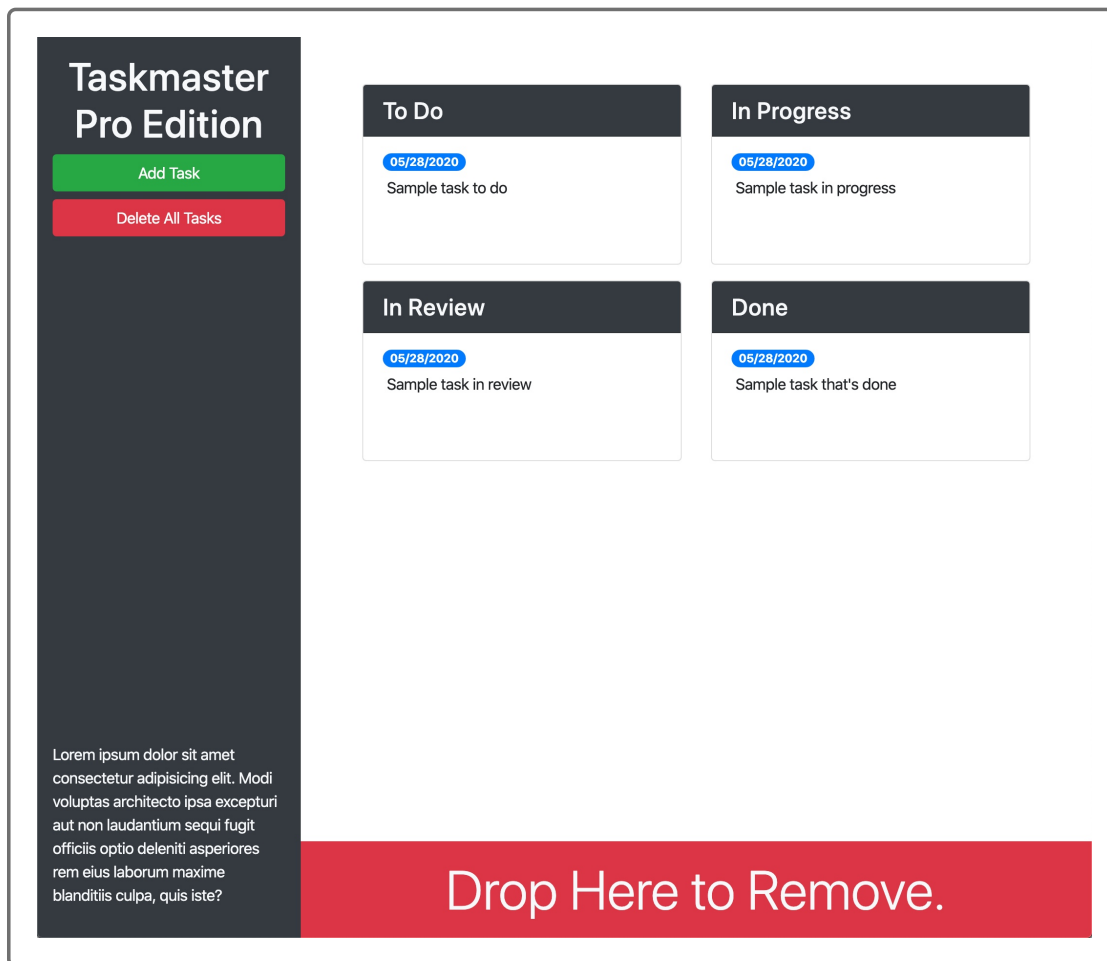
Once we've spotted what we want to use in the documentation, let's update all four of the card headers to look like this code:

```
<h4 class="card-header bg-dark text-light">
```

The cards should now match the mock-up the client provided us! Adding some contrast to the task lists makes them stand out a bit more and really enhances the overall design of the app.

Build a Trash Dropzone

Because the client has requested the ability to drag and drop task items from list to list, they'd also like to see another dropzone added for removing tasks. This is a lot like computer operating systems in which we drag items to a trash bin icon to delete them. The following image shows a mock-up of what they envision:



So it seems we need to create an element with a red background, flush to the bottom of the right column of the application. Creating an element with that type of background is simple, but how can we get this element to the bottom of the page?

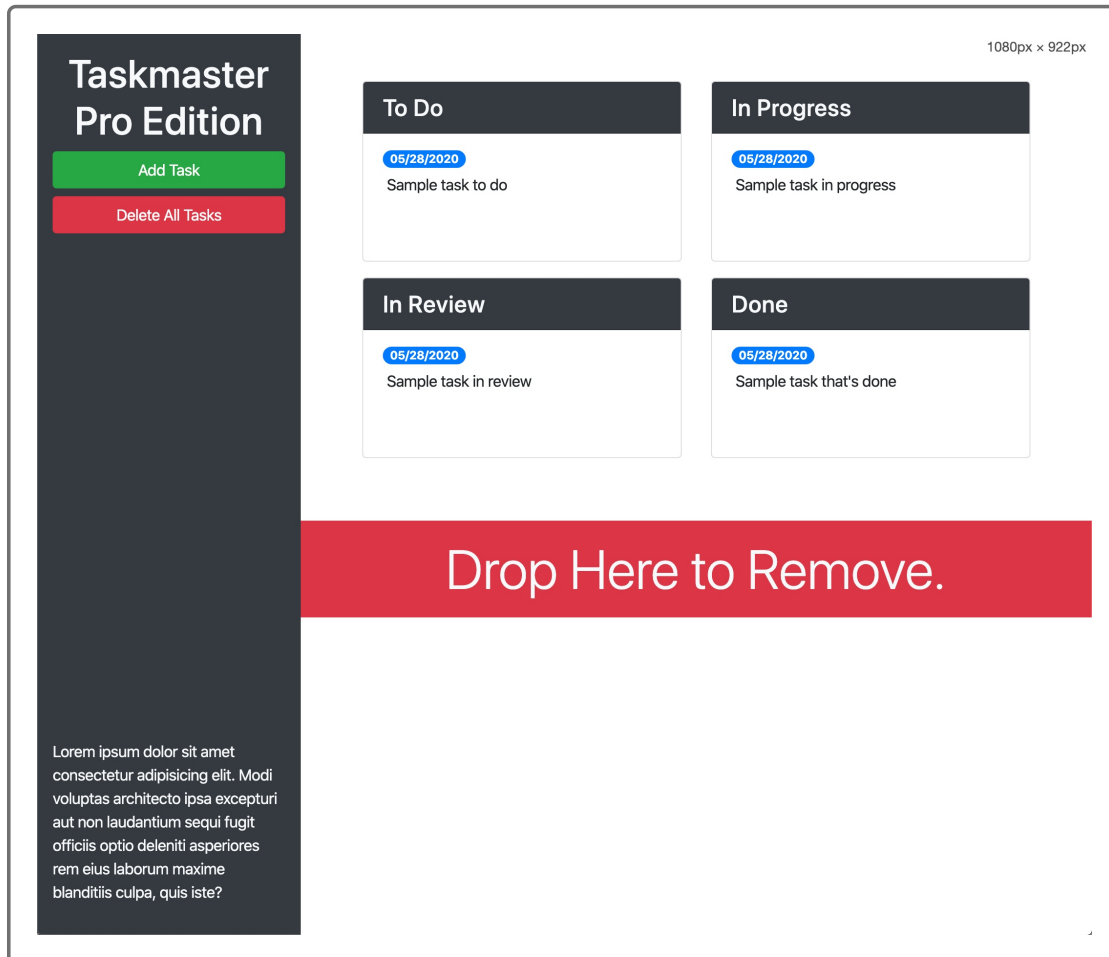
HIDE HINT

We already have something like this in place in the `<header>` element, since the `<p>` tag is flush to the bottom.

Let's start by building the element. After the closing `</div>` tag for the task list row and before the closing `</main>` tag, add the following code:


```
<div id="trash">
  <div class="w-100 bg-danger p-3 text-center text-light display-4">
    Drop Here to Remove.
  </div>
</div>
```

When we save and refresh, we should see something like this image:



Using just Bootstrap utility classes, we created an element that spans the entire width of the column, with a red background and centered text. Take a moment to look up some of the unfamiliar classes in the Bootstrap documentation.

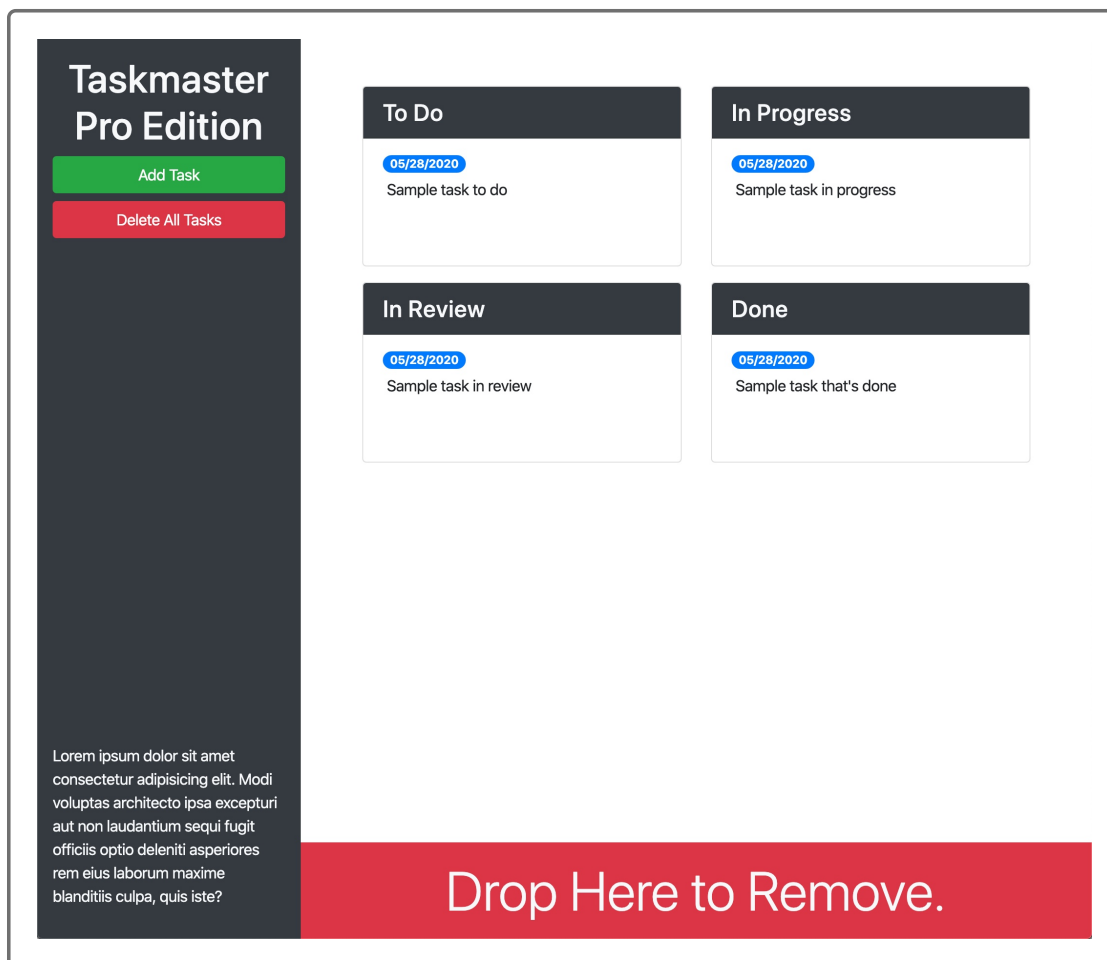
Now we need to push this trash area to the bottom of the page. Based on what classes the `<header>` and the `<p>` element in the header have, it

seems we can just convert the `<main>` element into a flex column. Let's add classes to the `<main>` element to look like this:

```
<main class="col-12 col-lg-9 d-flex flex-column">
```

Lastly, let's update the newly created trash dropzone element to have a class of `mt-auto`. This will push the trash element all the way to the bottom of the `<main>` element, now that it is a flex column.

When we save and refresh `index.html` in the browser, we should see that the client's mock-up looks like the following image:



Congratulations, you're now a Bootstrap expert! Bootstrap is a great tool for developers who don't have time to build a custom style sheet and need

to focus more on functionality. As you'll see later, you can still build custom styling over Bootstrap and have the two styles work in unison.

IMPORTANT

If you still have dummy task list data in `index.html`, don't forget to remove it.

Let's merge the feature branch and close the GitHub issue!

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