

5.5.6 Implement Font Icons

Just like using custom fonts before web font libraries, web icons used to be implemented with images as well. This meant that those little icons for social networks, logging in and out of a site, and providing error feedback all used to be images. This led to a few frustrations for developers.

As sites began to scale for the increasingly different screen sizes, multiple versions of the same image with different sizes needed to be used.

Without multiple sizes of the image, bigger screens might cause the icon's image to lose resolution. Another thing that was frustrating is icon images had a hard time aligning with text content that the icon complemented.

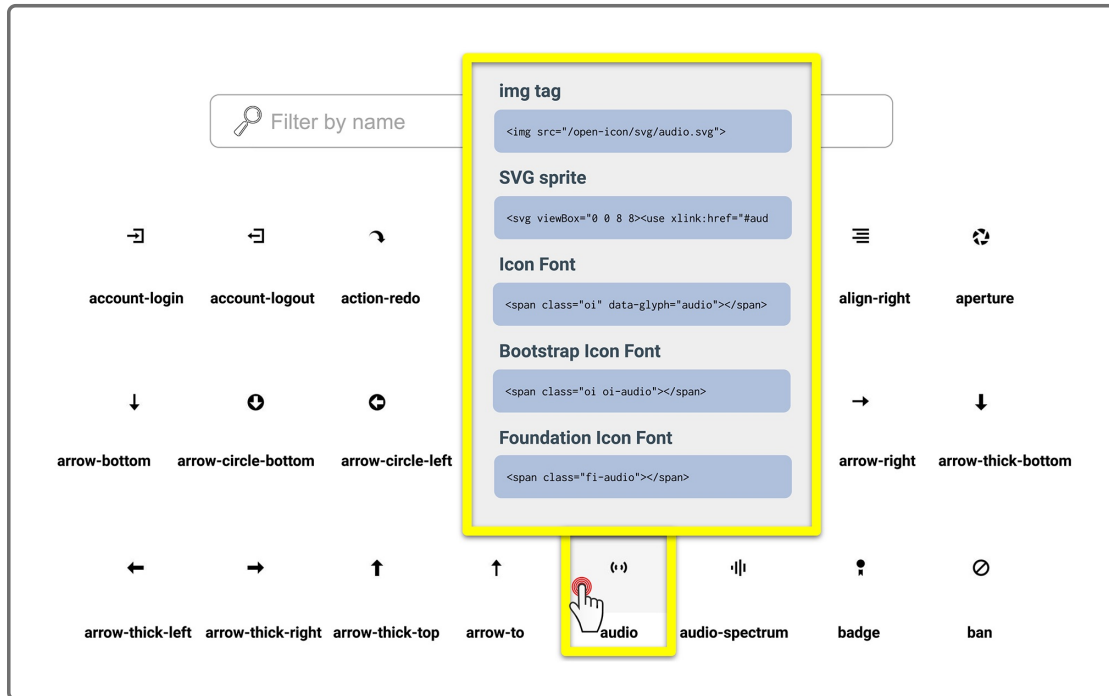
There are a lot more reasons why this was hard for developers to work with, and all you need to know is that it was annoying!

Today, however, a lot of these icons have been converted to SVG format and incorporated into what's known as a web font icon library. There are a number of services that offer web font icons, all having their own unique spin on popular icons, and we are going to use one that also happens to be recommended by the team at Bootstrap. This icon library is called Iconic.

Iconic has two variations, a professional (paid) version and a free (open) version. Let's visit the [Open Iconic documentation](https://useiconic.com/open) (<https://useiconic.com/open>) to see how we can use them in our application

and what icons they offer. Since this is the free version of Iconic, it's not as exhaustive of a list as the pro version, but we still get a ton of great icons to use.

If we click on one of the icons, a little box will appear that shows us how to use the icon, as shown in this screenshot:



As we can see, there are multiple options for how to use the icon. Since we're using Bootstrap already, we should probably use that version of the icons. All we have to do is copy and paste that `` element wherever we want the icon to appear. But there's one thing we need to do first.

PAUSE

Before we can use the Iconic icon library, what do we need to include in our `index.html`?

The CDN link for Iconic!

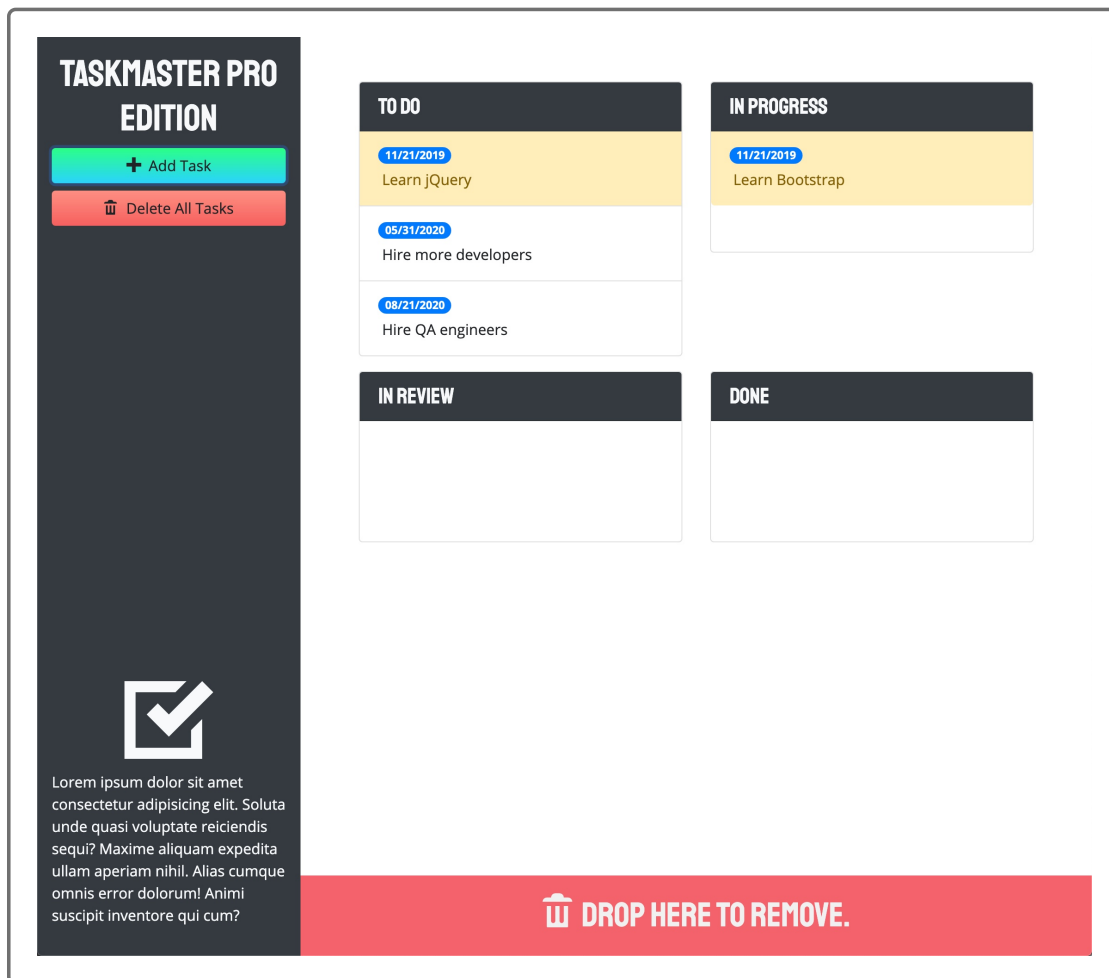
[Hide Answer](#)

In Open Iconic's documentation, there doesn't seem to be any mention of a CDN link for it, which is peculiar. But if we do a simple Google search for "Open Iconic CDN", we'll find the [link for Open Iconic on CDNJS](https://cdnjs.com/libraries/open-iconic).
(<https://cdnjs.com/libraries/open-iconic>)


Since we're using the Bootstrap version of it, let's select the link for the minified Bootstrap CSS and paste it into the `<head>` element in `index.html` so it's before our custom `style.css`. The `<link>` element should look like this code:

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/op
```

Once that's in, great! Let's take another look at the mock-up image from the client and identify the icons we need and where we need them:



So it seems we'll need the following icons:

- A plus sign  icon in the Add Task button
- A trash icon in the Delete All Tasks button
- A gigantic checkmark icon the bottom of the left column
- A trash icon in the trash dropzone

This doesn't seem like a lot, but it'll add a great deal of visual improvements for the app. Let's get to it!

In the [Open Iconic docs](https://useiconic.com/open) (<https://useiconic.com/open>), type the word "plus" into the search bar. The plus sign icon should be the only result. Click on it, copy the `` element under the Bootstrap option, and paste it into the left column's Add Task button so the code looks like the following:

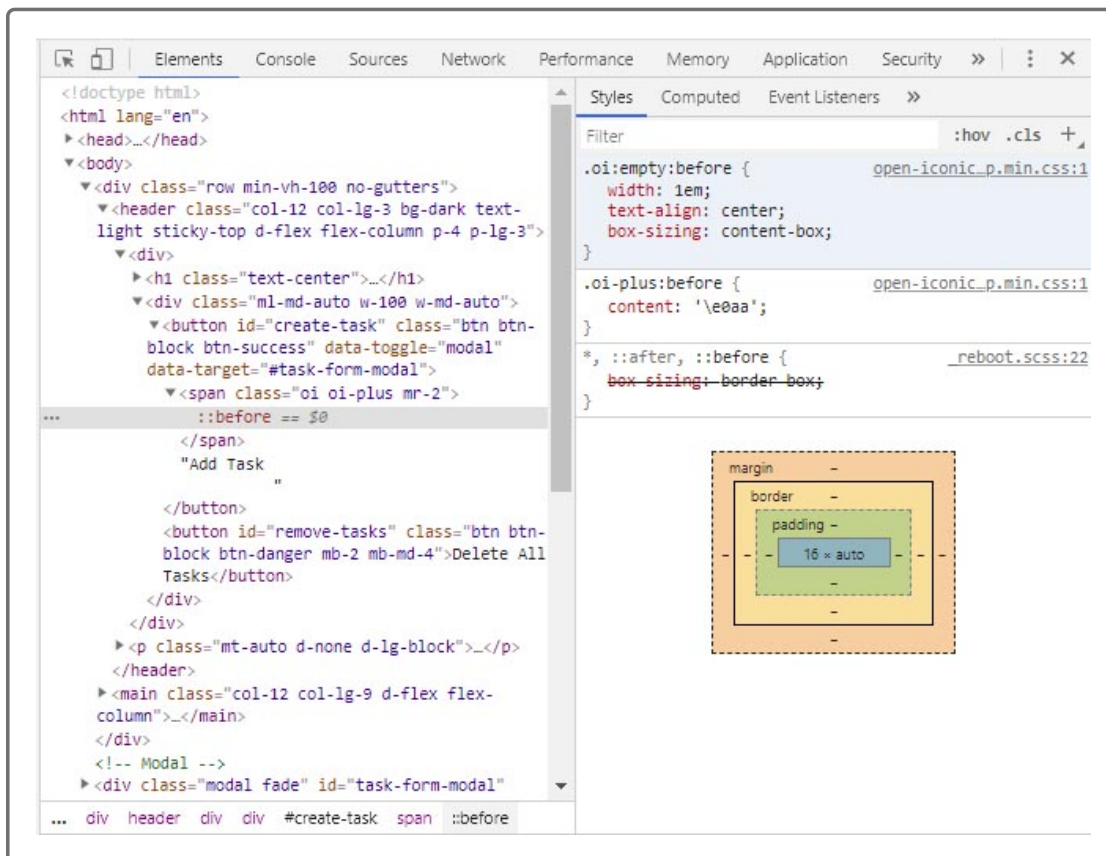
IMPORTANT

Also add a class of `mr-2` to the icon's `` for some right margin!

```
<button id="create-task" class="btn btn-block btn-success" data-toggle="collapse" -->  
  <!-- What does the mr-2 do? Hint: it's from Bootstrap! -->  
  <span class="oi oi-plus mr-2"></span>Add Task  
</button>
```

Save the file and refresh. You should see your icon show up in the button! But we didn't put anything between the `` tags, so how did the icon actually appear?

If we use the Chrome DevTools inspector to see what the `oi` and `oi-plus` classes bring to the table, we'll see standard CSS properties assigned to `oi`. The `oi-plus`, however, doesn't seem to have anything associated with it directly, but rather its `:before` pseudo-element, like this screenshot shows:



This is an interesting use of the `content` CSS property, as it has some weird string of characters set to its value. This string of characters actually represent the character for the plus sign `+` that we see on the page.

NERD NOTE

This type of code is what's known as **Unicode**.

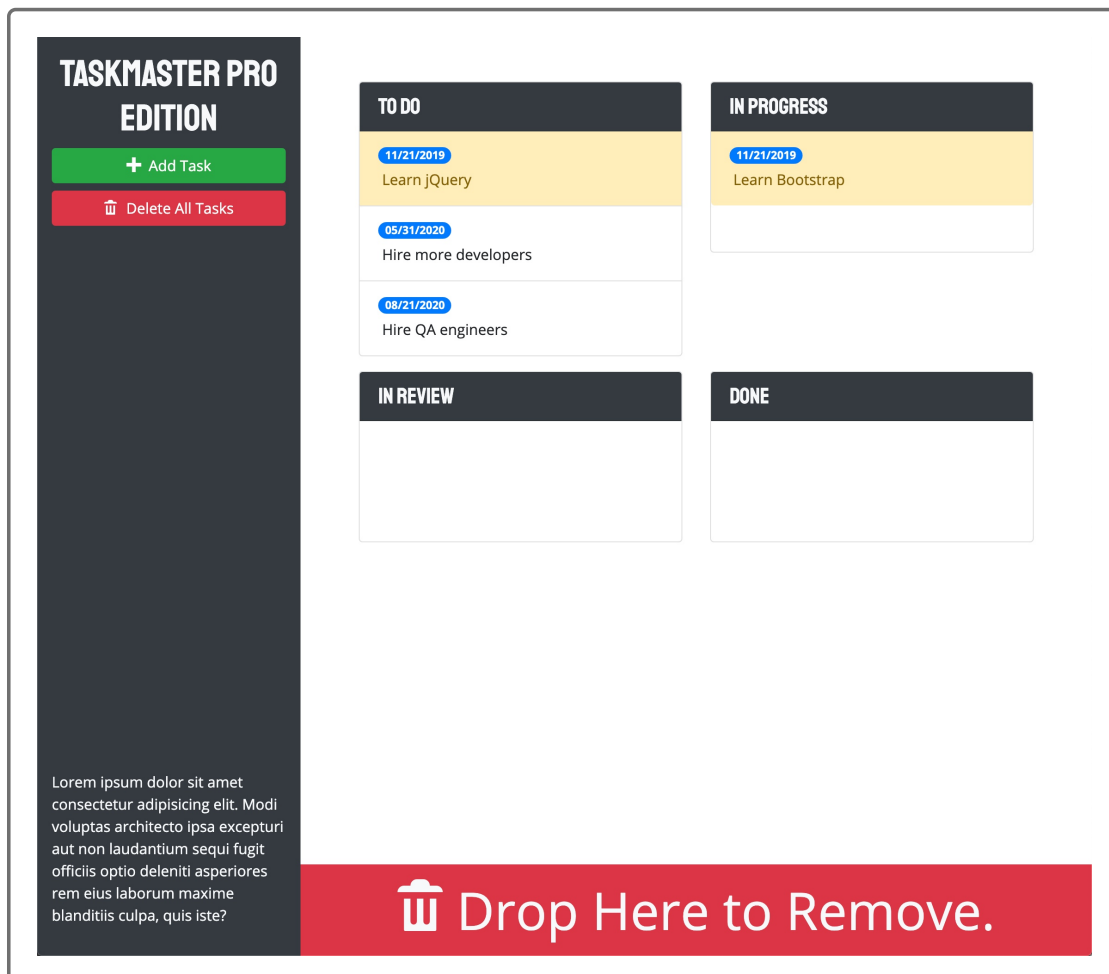
Add the Rest of the Icons

Now that we know how to use Open Iconic, let's add the other icons. Find the trash icon's code in the Iconic docs. Once you find it, copy the `` element for it and paste it in both the `<button>` element for deleting all tasks and the `<div>` element that holds our trash dropzone.

HIDE HINT

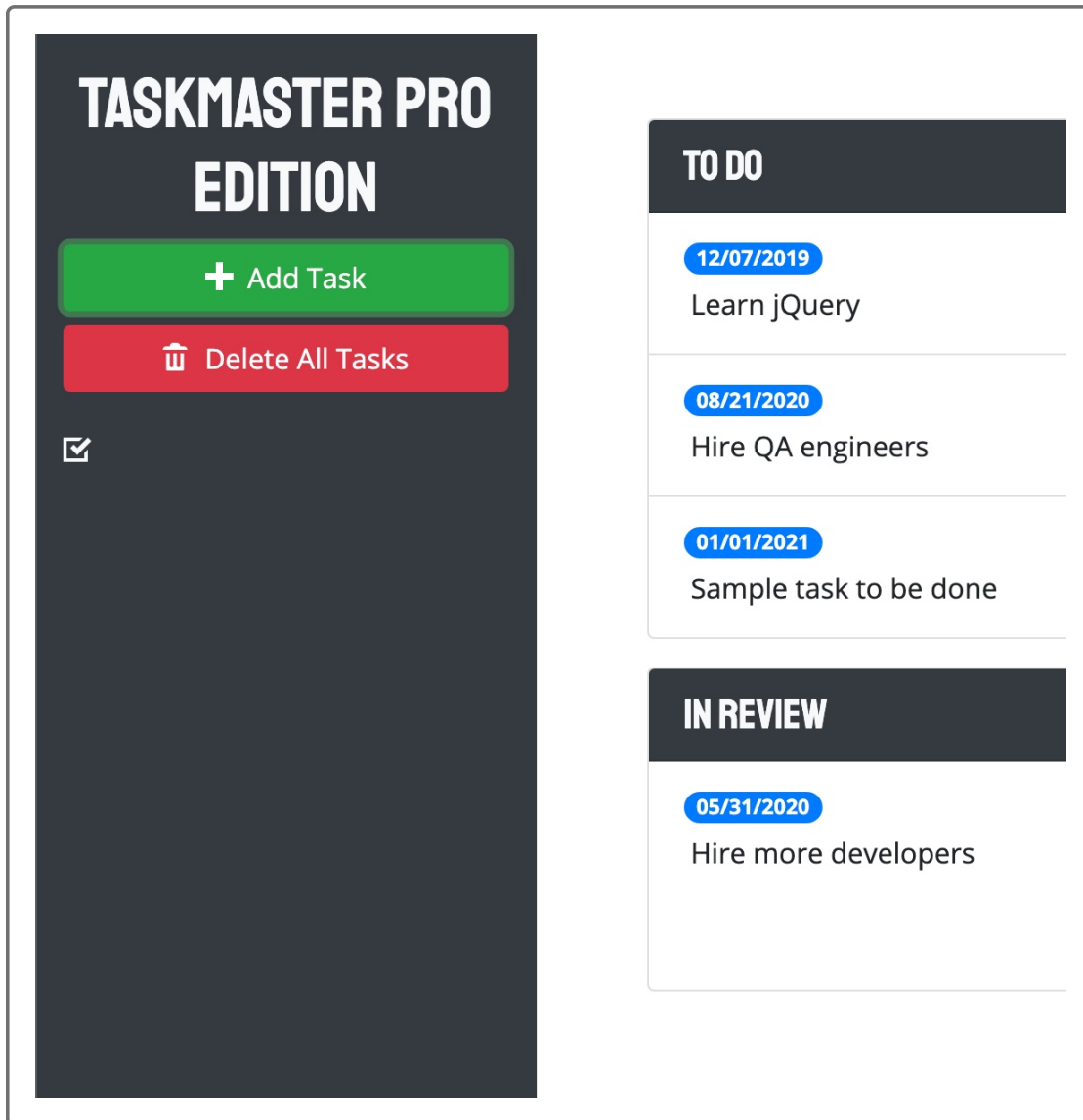
For the dropzone, put the trash icon right before the "Drop Here to Remove." text.

Save `index.html` and refresh the browser. You should see something like the following image:



Now let's worry about the icon in the bottom left of the page. If we search for "check" or "checkbox", we won't get the result we want. However, if we search for the word "task", we'll get the exact one we want! It's almost like this icon library knew we are building a task tracking application.

Let's copy that icon's code and paste it into the `<header>` element right before the `<p>` element. Once that's in place, we can see that it doesn't quite match the mock-up and looks more like the following image:

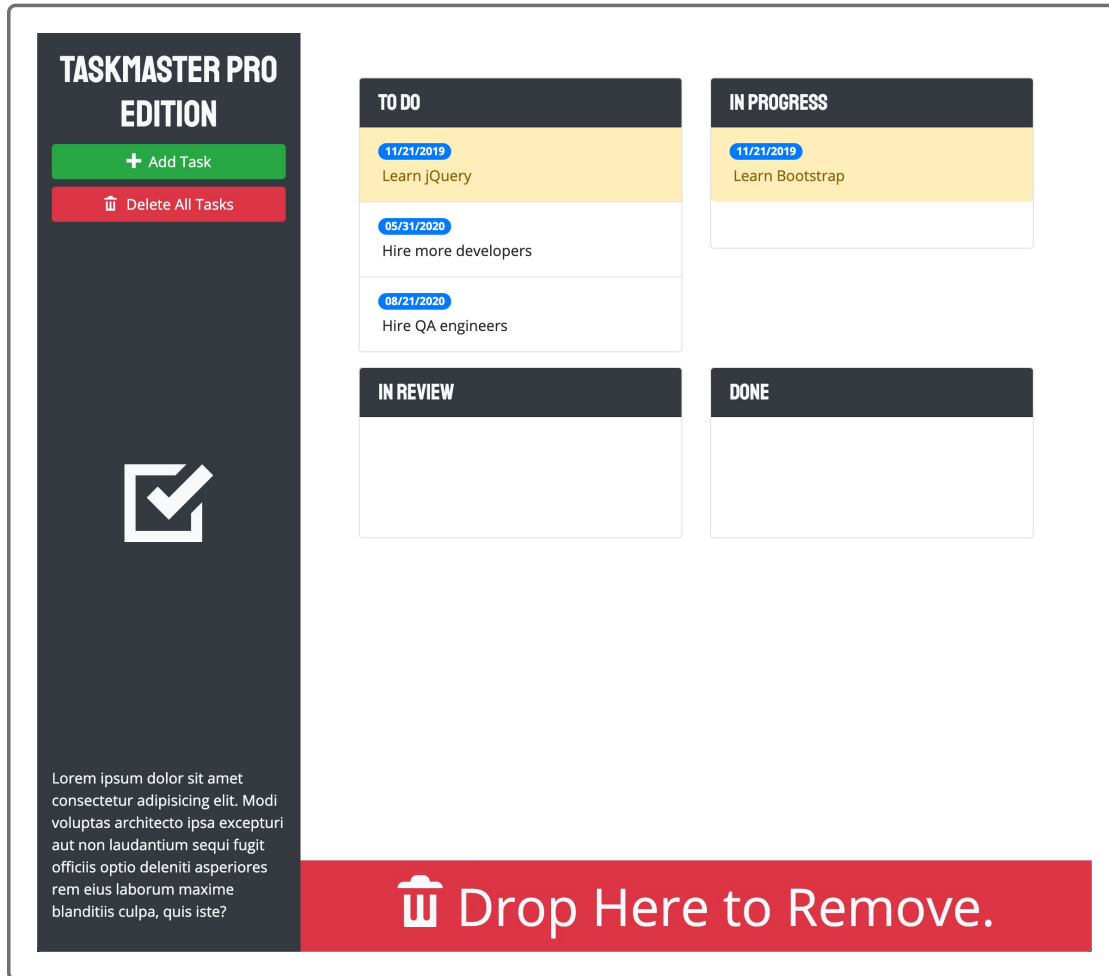


Let's add a few more classes to it so it looks like this when it's done:

```
<span class="oi oi-task display-1 text-center mb-2 d-none d-lg-block m
```

Now if we save `index.html` and refresh, we'll see that it looks good, but it's not bottom aligned! The three main parts of our `<header>` element are

now evenly spaced, as shown in the image below:



This is because both the icon and the paragraph have the Bootstrap class of `mt-auto`, which sets margin-top spacing to auto, but the `<header>` is a flexbox column: the two stylings are therefore in conflict. (See if you can conceptualize what each does and determine what the problem is.)

To fix this, simply remove the `mt-auto` class from the `<p>` element but leave it in the icon's `` element. The resulting code should look like this:

```
<span class="oi oi-task display-1 text-center mb-2 d-none d-lg-block m
<p class="d-none d-lg-block">
  Lorem ipsum dolor sit amet consectetur adipisicing elit. Soluta unde
</p>
```

HIDE PRO TIP

You may have noticed all of these Bootstrap classes added into the `` element. We haven't used all of these explicitly yet, so it's a good idea to look them up in the Bootstrap documentation or see what styles they add using Chrome DevTools.

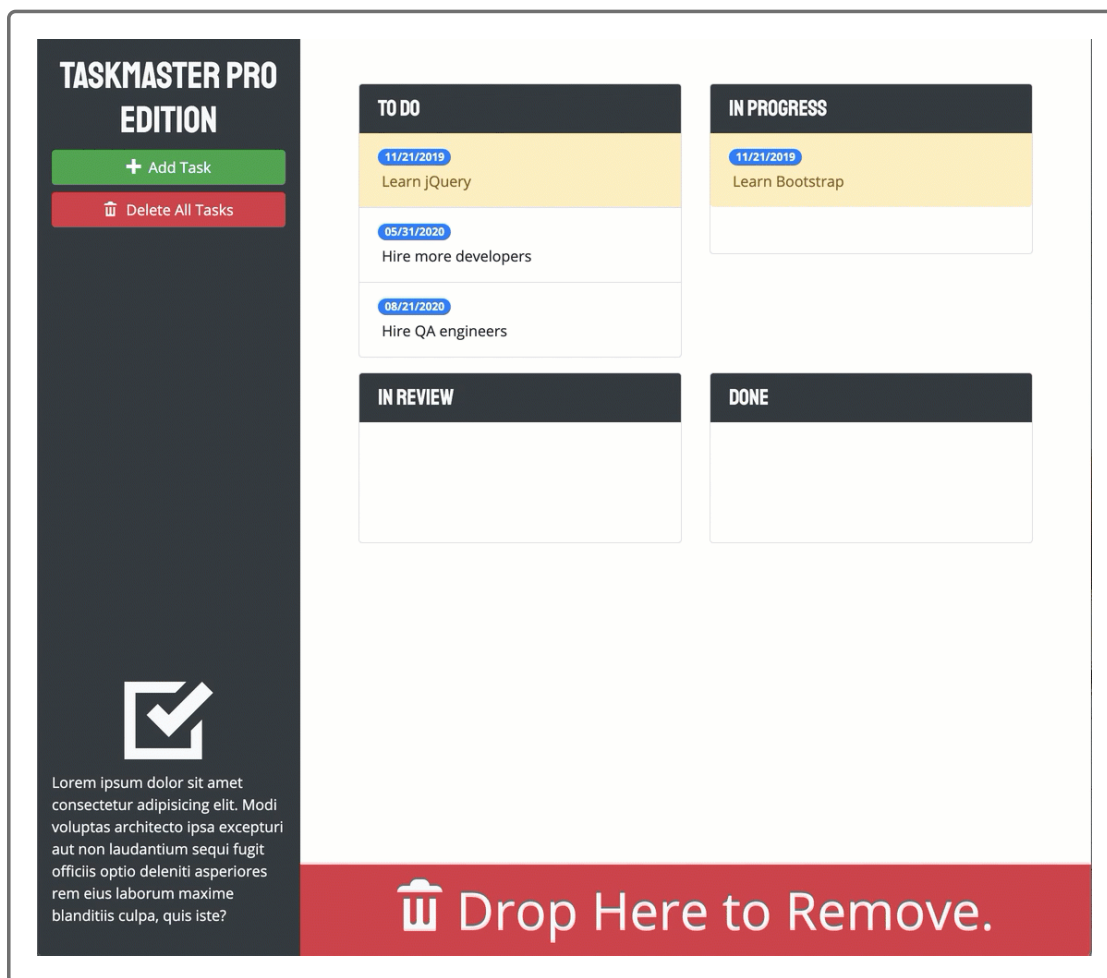
It pays to be curious!

There is one last thing to do, and then we'll be all set with our web icons! In that long list of Bootstrap classes added to the `` element, we're instructing it to not display at all when the screen gets smaller. This is good—we don't want the `<header>` to take up too much space on smaller devices, but the client still wants to see it *somewhere* on mobile.

Let's add this icon once more next to the application's name in the top of `<header>` so it looks like this:

```
<h1 class="text-center">  
  <span class="oi oi-task d-inline d-lg-none"></span>  
  Taskmaster Pro Edition  
</h1>
```

Now when we save, refresh the page, and resize the browser, we should see something like this animation shows:



Great! The client will be really happy that we found a way to keep the site's icon in the design for the mobile site.

So we've added automation, custom web fonts and now, custom web icons! We only have a few more updates to make, and it'll involve us putting our CSS skills to good use.

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