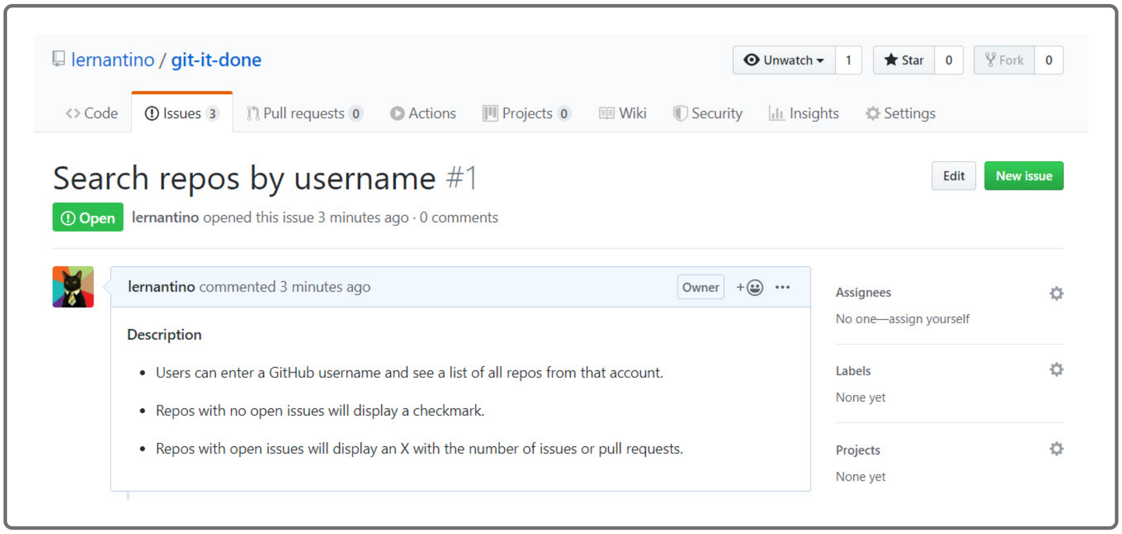
6.2.3 Add the Search Form

To begin, let's review the GitHub issue associated with this feature one more time:



Search repos by username #1

Open lernantino opened this issue 3 minutes ago · 0 comments

lernantino commented 3 minutes ago

Description

- Users can enter a GitHub username and see a list of all repos from that account.
- Repos with no open issues will display a checkmark.
- Repos with open issues will display an X with the number of issues or pull requests.

It may seem like we haven't accomplished much yet, because we haven't actually touched any of the bullet points listed for this feature. But in reality, we've made a lot of progress. The bullet points represent actionable items for the user to experience. However, they don't reflect all of the work that goes into the end goal. For example, we've already studied the GitHub API documentation and taken care of the HTTP request using `fetch()`. Now we just need to tie that functionality to the page!

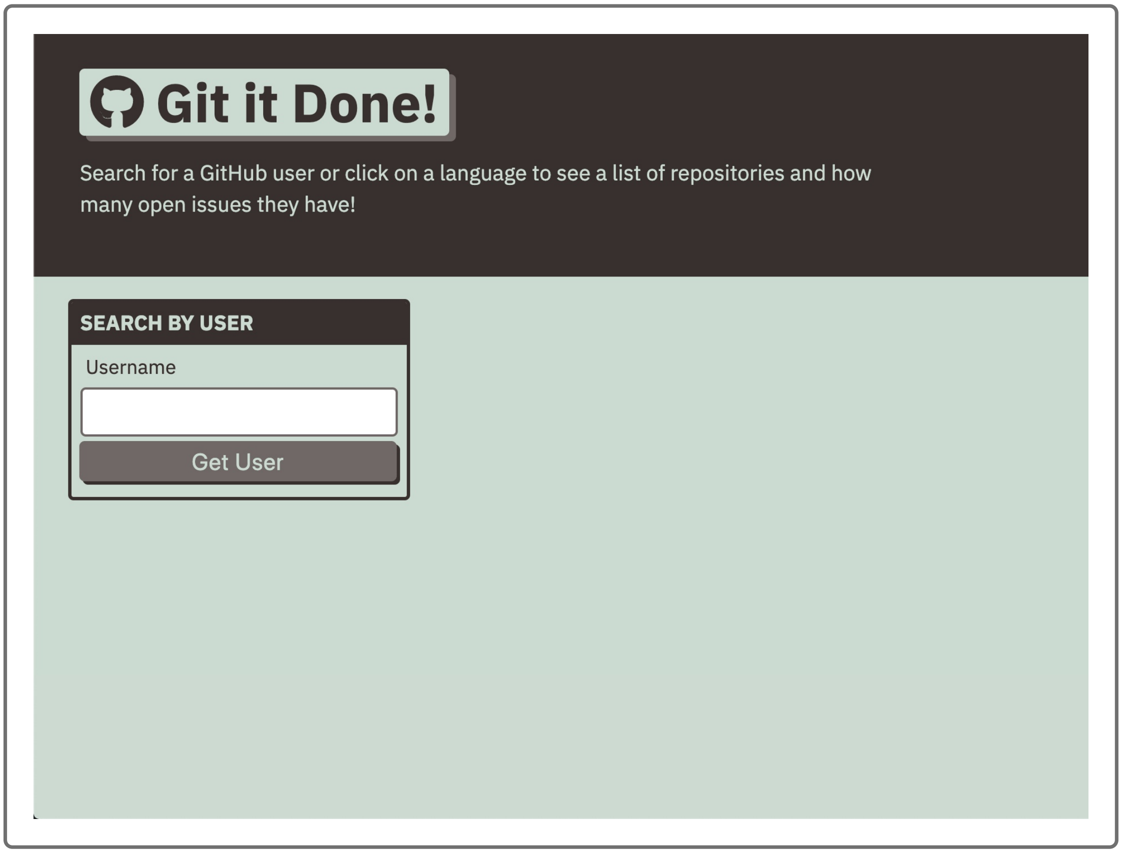
HIDE PRO TIP

Just as the GitHub issue for this feature doesn’t reflect all the work you’ve done so far, a client or project manager's requests may not accurately reflect the amount of work required for a project. Remember to allow for some setup time dedicated to reading documentation or learning a new tool to get the job done. You should always expect each job to take a little longer than estimated.

Now that we know how to use the Fetch API to create HTTP requests and handle their responses, let's tackle the first bullet point in this GitHub issue: "Users can enter a GitHub username and see a list of all repos from that account."

Build the Layout

To allow users to enter any GitHub username, we’ll need to add a form to to `index.html`. Amiko wants the form to look something like this image:


Git it Done!

Search for a GitHub user or click on a language to see a list of repositories and how many open issues they have!

The form doesn't need to take up much space on the page, so we'll create a two-column layout: we can place the form on the lefthand side, and we'll display the repositories to the right. Before we think about creating the form itself, let's create the two columns.

Update the `<main>` element so that it looks like this:

```html
<main class="flex-row justify-space-between">
  <div class="col-12 col-md-4"></div>

  <div class="col-12 col-md-8"></div>
</main>
```

We won't see a difference yet if we save `index.html` and refresh the page, but that's okay. At this point, we're just creating a responsive layout for the content to be placed into.

Did you notice that these classes look a bit like Bootstrap? Actually, Amiko created a custom style sheet that follows a lot of the same naming conventions. Take a moment to look over the CSS rules defined in `style.css`. You'll see that a lot of the classes leverage different uses of the `flex` property.

PAUSE

In a Bootstrap layout, what do the different `col-*` classes do for us?

They allow us to adjust the flexbox columns at different screen widths.

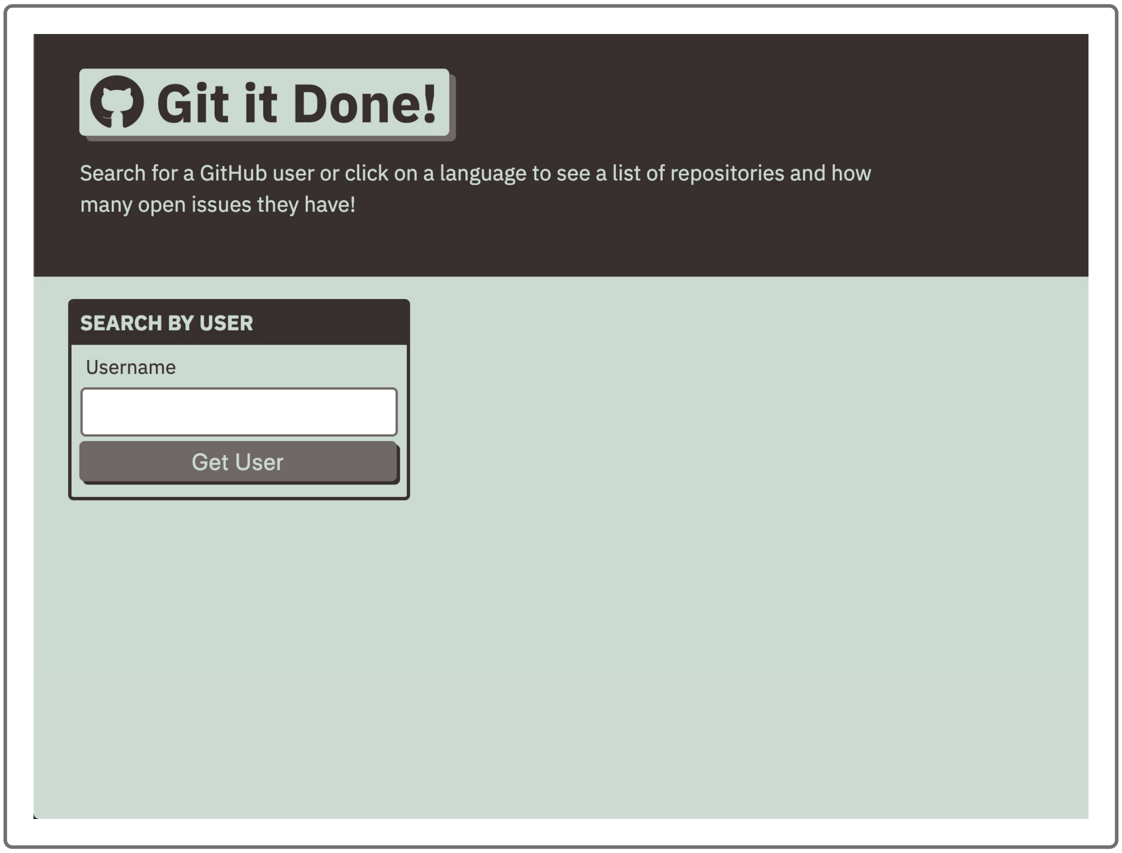
Hide Answer

Now that we've created the main layout, let's create the search form. We'll leverage that style sheet Amiko created.

Add the following code into the left column (the `<div class="col-12 col-md-4">` element):

```
<div class="card">
  <h3 class="card-header text-uppercase">Search By User</h3>
  <form id="user-form" class="card-body">
    <label class="form-label" for="username">Username</label>
    <input name="username" id="username" type="text" autofocus="true"
    <button type="submit" class="btn">Get User</button>
  </form>
</div>
```

Now if we save `index.html` and refresh the browser, we'll see something like this image:



Git it Done!

Search for a GitHub user or click on a language to see a list of repositories and how many open issues they have!

Great! Now we have a nice card component with a form inside it. Before moving on, study this HTML a bit more. We'll need some of this information if we want to capture the submitted form input.