6.1.3 Set Up the Project

We'll start setting up the project by creating a GitHub repository and cloning it on the local machine. Then we'll create GitHub issues to define the requirements for the remaining features. Finally, we'll create branches to work in.

1. Create a new repository in GitHub called `git-it-done`. Make it a public repository and check the box for "Initialize this repository with a README."

2. Go to the Issues tab and create the following issue to make feature branches from (you can cut and paste the following Markdown):

```
Title: Search repos by username

**Description**

- Users can enter a GitHub username and see a list of all repos fr

- Repos with no open issues will display a checkmark.

- Repos with open issues will display an X with the number of issu
```

```
Title: See list of all open issues

**Description**

- Users can click on a repository name, which will route them to

- The second HTML page will display a list of all open issues fo
```

```
Title: Search for featured repos

**Description**

- Users can click on a language button to see a list of featured
```

3. At the command line, use the `cd` command to navigate to the directory where you'll clone the repository.

4. Clone the repository using the `git clone` command.

5. Add Amiko's HTML and CSS files to the project by downloading them from here:

   Zip of module 6 starter code (https://static.fullstack-bootcamp.com/module-6/module-6-starter.zip)

6. Create a `.gitignore` file and add the line `.DS_Store` to it.

REWIND

As you learned earlier, a `.gitignore` file allows you to instruct Git to not track certain files and folders. Because you'll share this codebase with Amiko, you'll need to avoid unnecessary conflicts on system files, like the macOS `.DS_Store` file.

Your project should now consist of the following files:

- `assets/css/style.css`
- `.gitignore`
- `index.html`
- `README.md`
- `single-repo.html`

Push the current codebase to GitHub by using the following commands:

```
git add -A
git commit -m "initial files"
git push origin master
```

Let's go ahead and enable GitHub Pages for this project so that we can send a deployed version to Amiko. In GitHub, navigate to your repo's Settings tab, scroll down to the GitHub Pages section, and change the source to `master branch`. After a few minutes, the app will appear live at the following URL: `https://<username>.github.io/git-it-done/`.

Now that we've enabled GitHub Pages, we're almost ready to start on the first feature. But before we make any changes to the files, we should create the `develop` and feature branches by using the following commands:

```
git checkout -b develop
git checkout -b feature/user-repos
```

HIDE HINT

Don't forget to use the `git branch` command to confirm what the active branch is.

Now that the project is officially set up, let's get to work!