

## 5.1.3 Set Up the Project

First you'll need access to the current Taskmaster codebase. Download the following zip file:

**Starter code** [\\_\(https://static-fullstack-bootcamp.s3-us-west-1.amazonaws.com/module-5/module\\_5\\_starter.zip\)](https://static-fullstack-bootcamp.s3-us-west-1.amazonaws.com/module-5/module_5_starter.zip)

On your computer, create a new folder called `taskmaster-pro` and copy the contents of the zip file into it. This should include the following folders and files:

- `index.html`
- `assets/css/style.css`
- `assets/js/script.js`

This folder is now the basis of the project, but every project should have its own repository. From the command line, navigate to the `taskmaster-pro` directory and initialize it as a Git repository.

You can use these commands for reference:

```
cd taskmaster-pro  
git init
```

Remember, `git init` only establishes a local folder as a Git repository. The repository doesn't exist on GitHub yet. Let's take care of that now:

1. In the browser, go to [GitHub.com](https://github.com/) (<https://github.com/>), make sure you're logged in, and create a new repository named `taskmaster-pro`.
2. Copy the repository URL, then return to the command line and run the following commands (replacing the example URL with yours):

```
git remote add origin git@github.com:<username>/taskmaster-pro.git  
git remote -v  
# IMPORTANT: You need to sync up with the remote GitHub repo first to  
git pull origin master
```

You just linked the local `taskmaster-pro` folder to your GitHub repository, a process that you're surely starting to become familiar with by now. Great job!

### IMPORTANT

Previously, you used `git clone` to set up some of your projects. You only need to use the command `git clone` if you don't have the files you want on your computer and the repository already exists on GitHub. Using `git clone` downloads the repo to your computer.

The command `git init`, however, is useful when files on your computer aren't yet part of a Git repository. Using `git init` will turn the folder into a repo that can then be linked to a remote repo (e.g., a GitHub repo).

In either case, you'd only ever use `git init` or `git clone` once, when first setting up a project.

Now that you've set up the repository, push the current codebase to GitHub with the following commands:

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

Because this is already a working app, you can publish it now with GitHub Pages. 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/taskmaster-pro/`.

You can play around with the live version of the app or open the local `index.html` file in Chrome. Either way, you'll want to see what the current app can do before jumping into the code. Make sure you create a few sample tasks!

### IMPORTANT

Before testing the app, you might need to clear `localStorage` in case any of the key names conflict with previous projects. To clear this data, run `localStorage.clear()` in the DevTools console.

Awesome—we're ready to map out the rest of the project!