

# Code a Simple Website and Host it for Free

A step-by-step Guided Learning to build a simple website and host it for free using GitHub and Netlify.

6·8 min read



Who is this tutorial for?

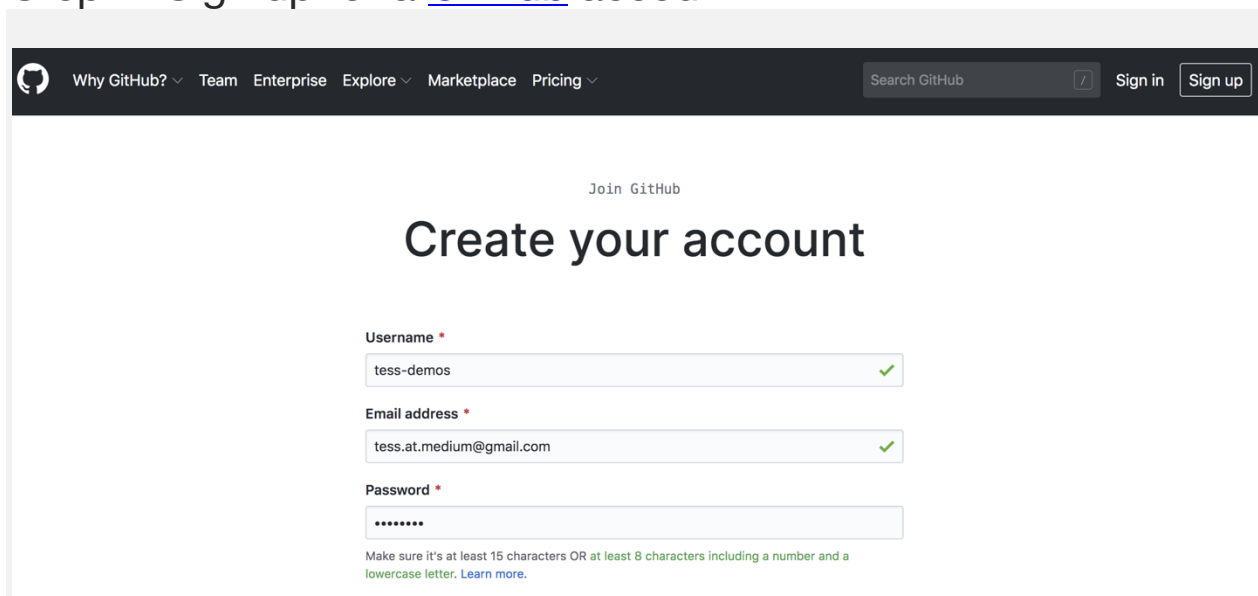
- *This tutorial assumes no prior knowledge and is suitable for complete beginners as a first project.*

## What you will need

- a *GitHub* account (if you already have one set up, skip step 1)
- a *Netlify* account
- a code editor ([Visual Studio Code](#))
- terminal app (*iTerm2*)
- approximately 1–2 hour

## Let's get started!

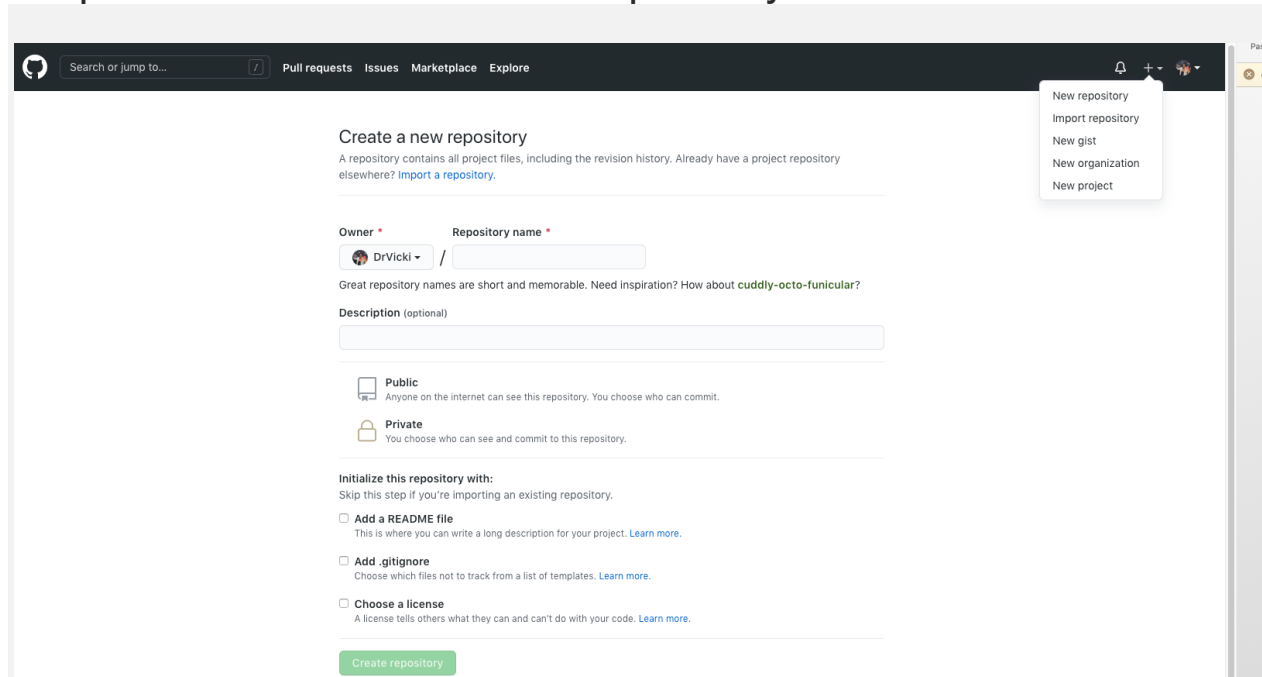
Step 1: Sign up for a [GitHub](#) account.

A screenshot of the GitHub website's account creation page. The page has a dark header with navigation links: 'Why GitHub?', 'Team', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing'. On the right of the header are a search bar, a 'Sign in' button, and a 'Sign up' button. The main content area is white and features the text 'Join GitHub' followed by 'Create your account' in a large, bold font. Below this, there are three input fields: 'Username' with the value 'tess-demos', 'Email address' with the value 'tess.at.medium@gmail.com', and 'Password' with masked characters. Each of the first two fields has a green checkmark to its right. Below the password field, there is a small note: 'Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. [Learn more.](#)'

*What exactly is GitHub? The Git in GitHub is a version control system, so that every time anything changes in our code, that change is tracked. This lets you trace everything you've ever written and*

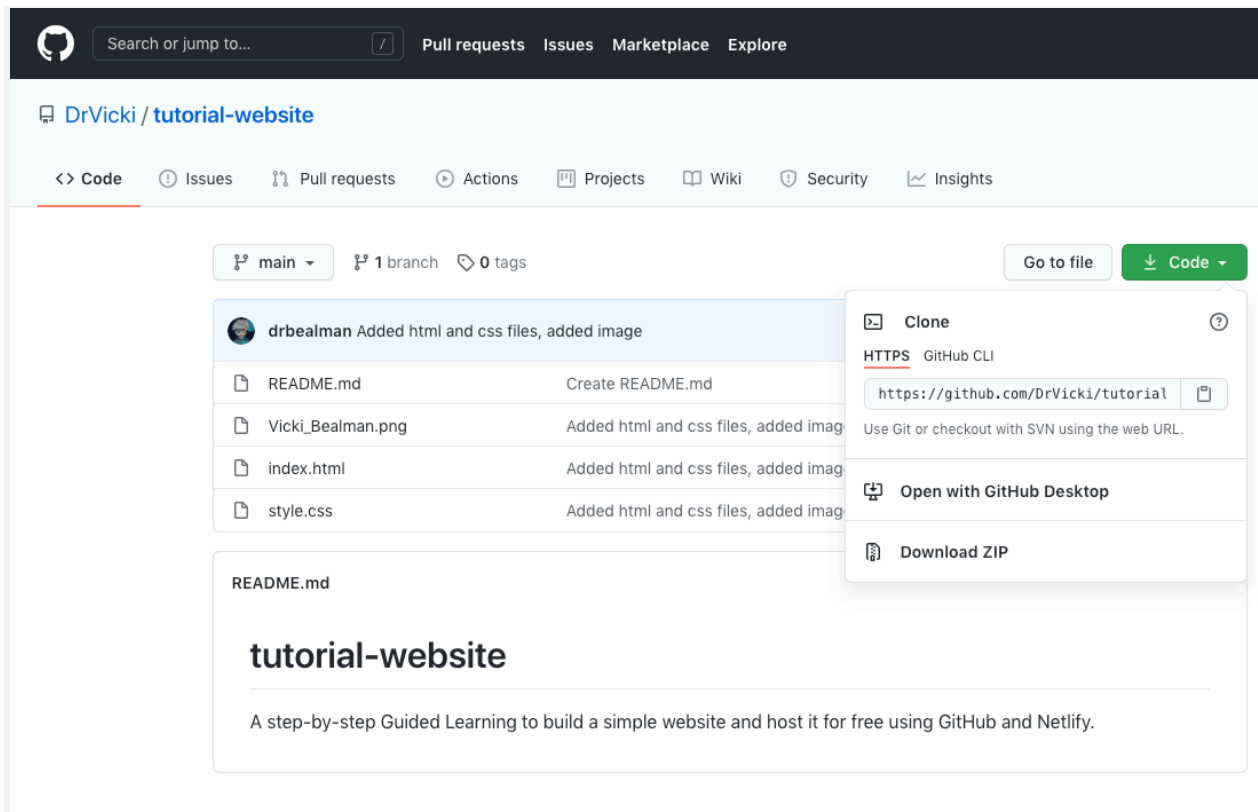
*changed within a project and revert back to an old version of your code if you need to. Git on its own is a command-line tool. GitHub is where it all comes together. It's where we can store our projects, track all our work and code changes, as well as network with other developers (and check out their projects!).*

## Step 2: Create a new GitHub repository

The screenshot shows the GitHub 'Create a new repository' page. At the top, there's a navigation bar with links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below this, the main heading is 'Create a new repository'. A subtext explains that a repository contains all project files, including revision history, and offers a link to 'Import a repository'. The form includes fields for 'Owner' (set to 'DrVicki') and 'Repository name'. A note suggests repository names should be short and memorable, with an example 'cuddly-octo-funicular?'. There's an optional 'Description' field. Under 'Visibility', 'Public' is selected, with a note that anyone can see and commit. 'Private' is also an option, where only chosen users can see and commit. The 'Initialize this repository with:' section has three checkboxes: 'Add a README file' (selected), 'Add .gitignore', and 'Choose a license'. Each has a brief description and a 'Learn more' link. A green 'Create repository' button is at the bottom.

*It's good practice to initialize your project with a README file. In this file, you can put some information about your project so that anyone who is interested can understand what the project is and how to make sense of the project files.*

## Step 3: Clone your repository on to your computer



- Click on “Clone or download” and copy the HTTPS URL
- Open up your terminal (on a mac just hit the search icon and type **Terminal**)

```

Last login: Tue Nov 10 08:48:23 on ttys000
vbealman-mbp:~ vbealman$ pwd
/Users/vbealman
vbealman-mbp:~ vbealman$ ls
AndroidStudioProjects  Documents          Justimind          Movies              Untitled.ipynb      git101              zipinfo
Applications           Downloads          LavinniaCatRunner  Music               android-sdks         iCloud Drive (Archive)
Creative Cloud Files    IDrive Downloads  Library            Pictures            eclipse              keystores:upload-keystore.vdb
Desktop                IDrive-Sync       MQMS               Public              eclipse-workspace    playservices_unity
vbealman-mbp:~ vbealman$

```

*The Terminal is the interface to the underlying operating system of our computer. It is a command line. Let's navigate to where we want*

*to keep our project files and make a “clone” of the repository that we just created on GitHub.*

*Give it a go with the following commands:*

```
$ pwd
```

*This stands for print working directory. As you can see above, it will show you where you are within your files.*

```
$ ls
```

*This one will give you a list of all the directories and files within your current directory. It stands for short listing as it only gives you the name of the file or directory.*

*To learn more terminal commands, check out this [cheat sheet](#).*

*Let's create a directory for our project*

```
vbealman-mbp:documents vbealman$ git clone https://github.com/DrVicki/tutorial-website.git
Cloning into 'tutorial-website'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), done.
vbealman-mbp:documents vbealman$ cd tutorial website
-bash: cd: tutorial: No such file or directory
vbealman-mbp:documents vbealman$ cd tutorial website
-bash: cd: tutorial: No such file or directory
vbealman-mbp:documents vbealman$ cd tutorial-website
vbealman-mbp:tutorial-website vbealman$ ls
README.md
vbealman-mbp:tutorial-website vbealman$
```

*Use the following commands to create a new directory and clone your project*

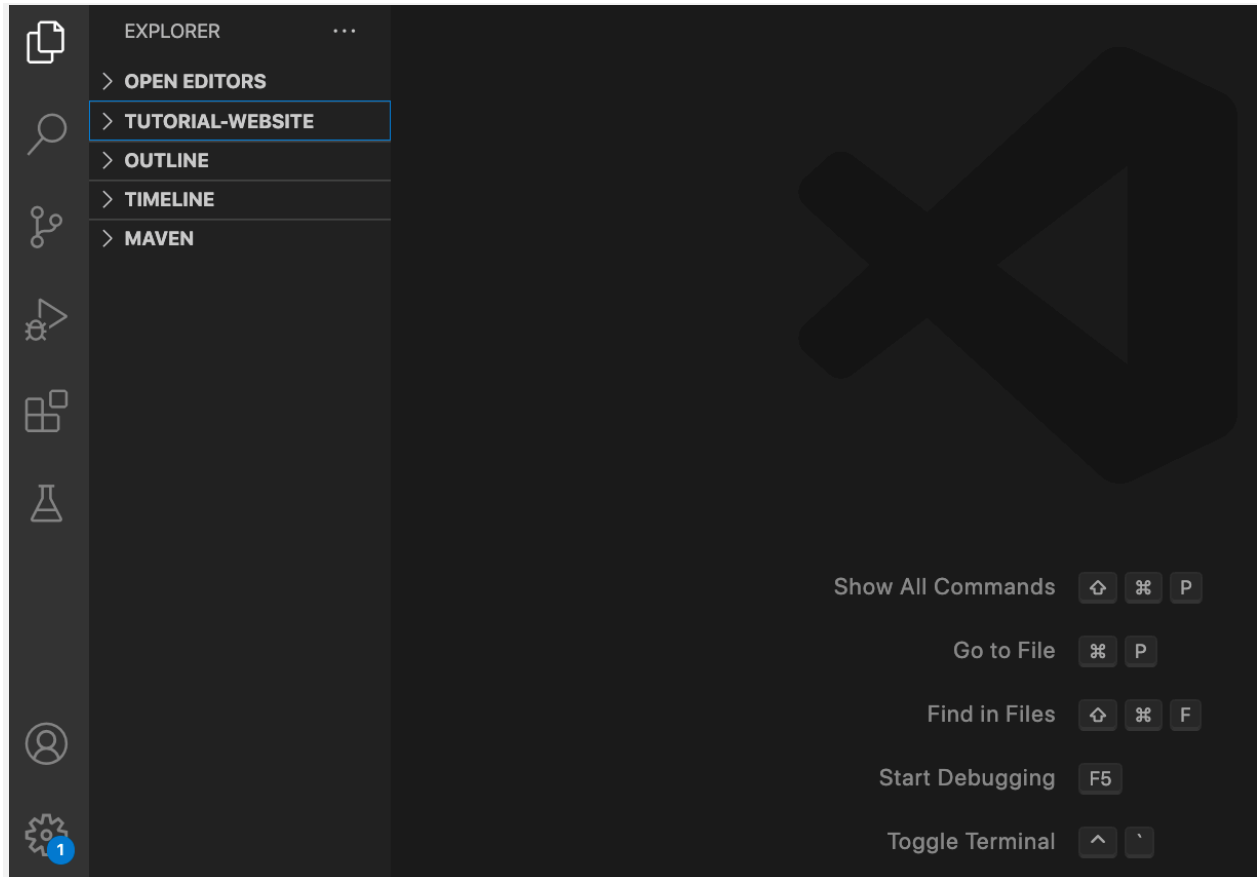
```
$ cd Documents  
$ git clone <HTTPS URL from your GitHub repository>  
$ cd <name of your project repository>  
$ ls
```

*Now we have a new directory; mine is called tutorial-website, and within this directory, we can type “ls” to see that we have our README file.*

**Step 4: Open your code editor and create your project files**

*There are lots of different code editors and everyone has their personal preferences. For DigitalCrafts, we will use Visual Studio Code. It's free, it's simple, it's good. You can download it [here](#).*

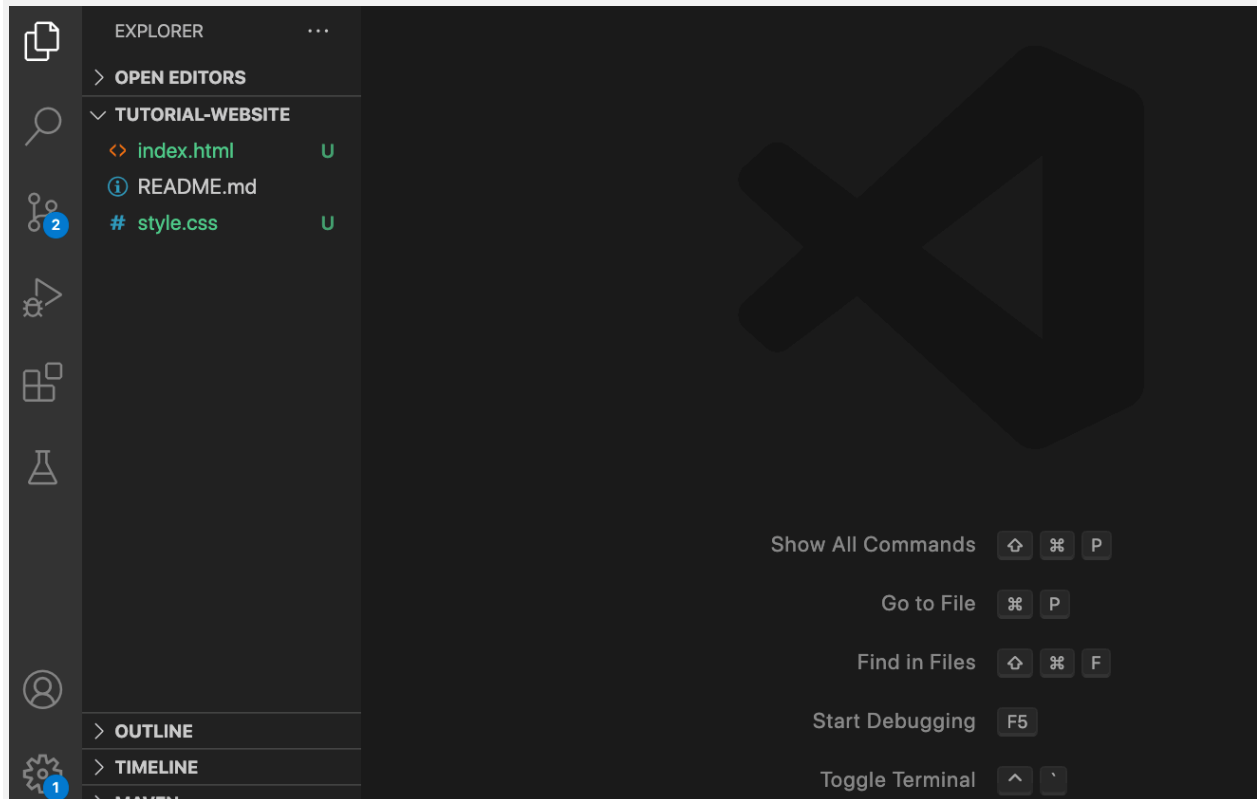
- *Let's launch VS code and open our project*



*So far we only have our README file. Let's go to our terminal and create 2 new files, one will be our HTML file and the other will be the CSS file.*

```
$ ls (to make sure you are still in the right directory)
$ touch index.html
$ touch style.css
```

*The touch command followed by the name of our file is how we can create a new file. Now let's go back to VS code and see our files.*



*Our **2 new files** have arrived in our project folder and they are **highlighted green** so that we know that they are new.*

## Step 5: Write some code!

- let's open our index.html file and write some code*

```
<!DOCTYPE html>
<html>
<head>
  <title>Tutorial Website</title>
</head>
<body></body>
</html>
```



- *This boilerplate code is the structure of our Html file. In the `<head>` tag, we have our meta-data, which is simply information about our website.*
- *In the `<body>` tag is where we will write our code to bring our website together*

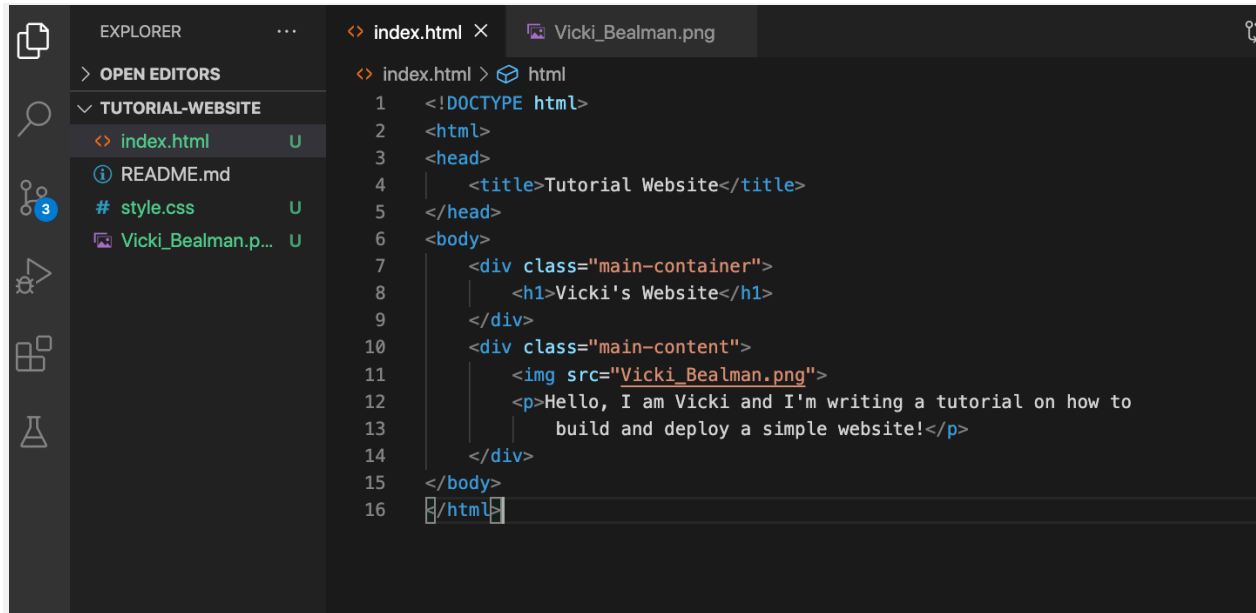
*Let's put some content in the `<body>` tag.*

*You can copy and paste the code below of course but I've found that the best way to learn how to code when you're just getting started, is to type everything out!*

```
<!DOCTYPE html>
<html>
<head>
  <title>Tutorial Website</title>
</head>
<body>
  <div class="main-container">
    <div class="header">
      <h1>Vicki's Website</h1>
    </div>
    <div class="main-content">
      
      <p>Hi, my name is Vicki and I'm writing a tutorial on how to
build and deploy a simple website!</p>
    </div>
  </div></body>
</html>
```

*I've decided to put an image on my website. Make sure to add your image to the project folder and put the file name in the `<img>` tag.*

*This is how everything should look at this point:*



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Tutorial Website</title>
5 </head>
6 <body>
7   <div class="main-container">
8     <h1>Vicki's Website</h1>
9   </div>
10  <div class="main-content">
11    
12    <p>Hello, I am Vicki and I'm writing a tutorial on how to
13      build and deploy a simple website!</p>
14  </div>
15 </body>
16 </html>
```

*How can we tell if it's working? Simply open your index.html file with your browser.*

*This is how it's looking so far:*



## Step 6: Adding CSS styling

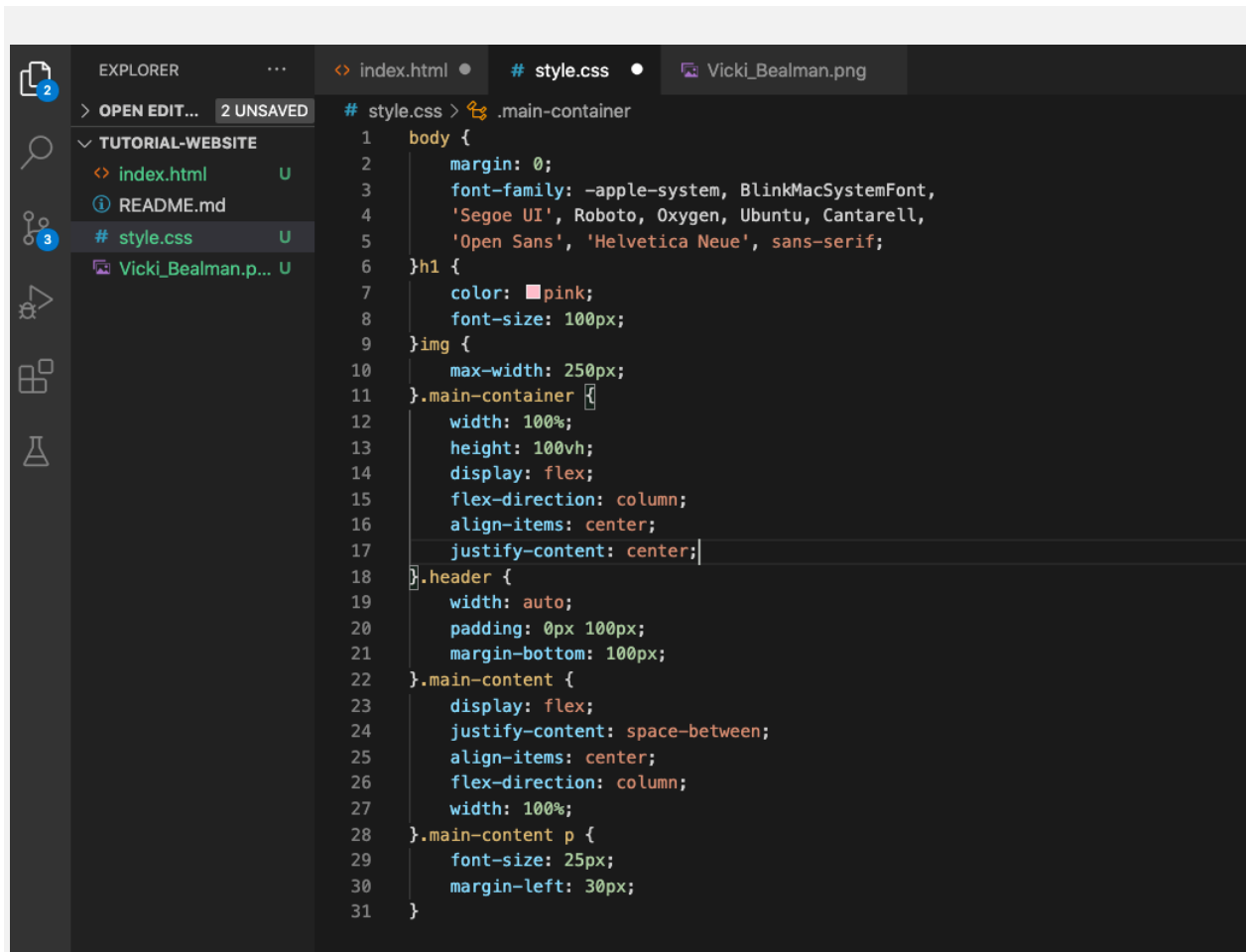
*Let's make it look good! Html is the structure of our page, but CSS is where the magic happens to make it look just how we want it.*

- *Open the style.css file*

*Let's write some code*

```
body {
  margin: 0;
  font-family: -apple-system, BlinkMacSystemFont,
    'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell,
    'Open Sans', 'Helvetica Neue', sans-serif;
}h1 {
  color: pink;
  font-size: 100px;
}img {
  max-width: 250px;
}.main-container {
  width: 100%;
  height: 100vh;
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
}.header {
  width: auto;
  padding: 0px 100px;
  margin-bottom: 100px;
}.main-content {
  display: flex;
  justify-content: space-between;
  align-items: center;
  flex-direction: column;
  width: 100%;
}.main-content p {
  font-size: 25px;
  margin-left: 30px;
}
```

*This is how our CSS file should look now:*



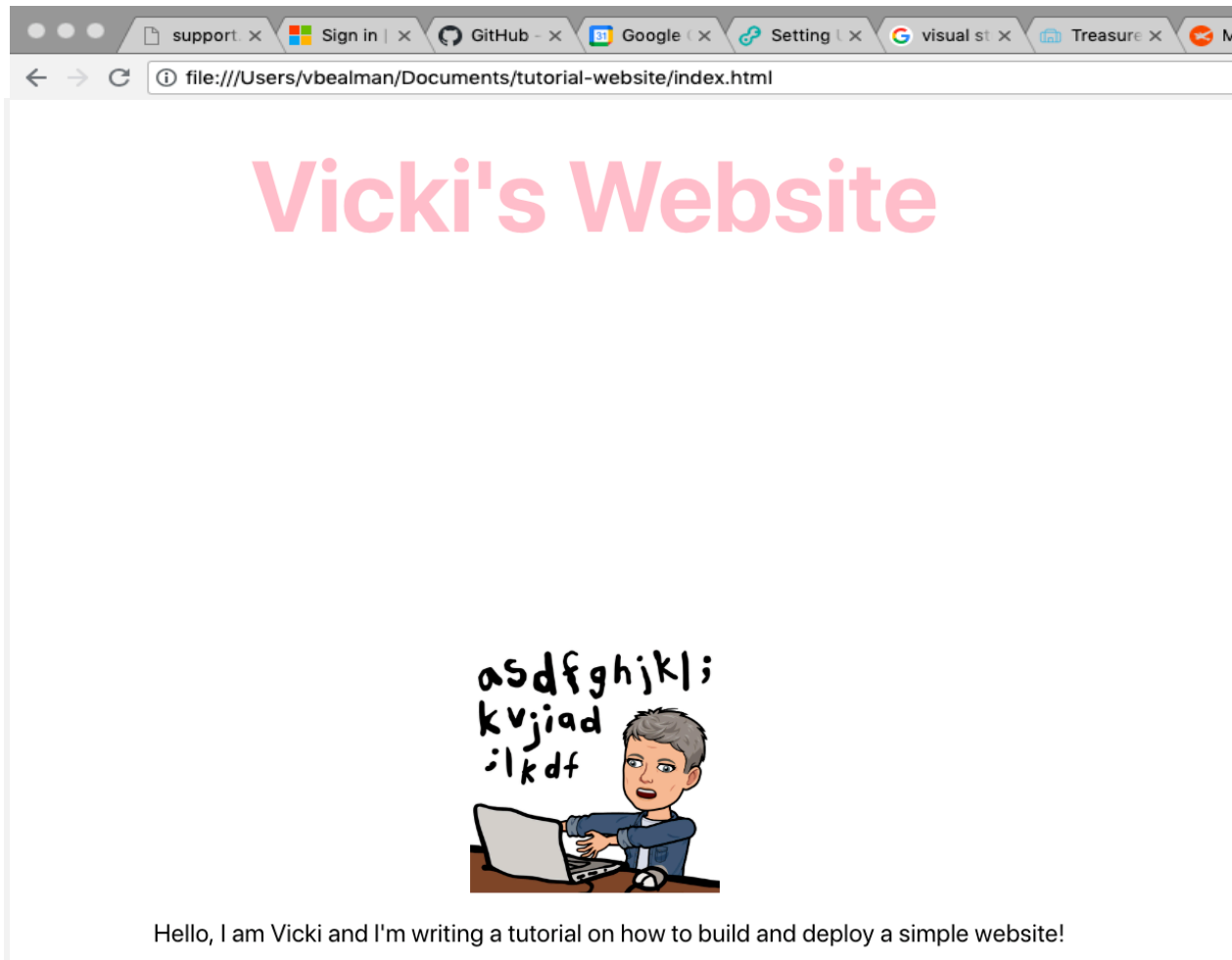
Now we need to make sure our CSS file is referenced in our Html file so that our website knows to read the styles we have defined in our CSS file.

In our Html file, let's add some code to our <head> tag.

```
<!DOCTYPE html>
<html>
<head>
  <title>Tutorial Website</title>
  <link rel="stylesheet" type="text/css" href="style.css">
</head>
<body>
  <div class="main-container">
    <div class="header">
      <h1>Tess's Website</h1>
```

```
</div>
<div class="main-content">
  
  <p>Hi, I am Vicki and I'm writing a tutorial on how to build
and deploy a simple website!</p>
</div>
</div></body>
</html>
```

*Let's go ahead and check how everything is looking now in our browser!*



*A bit of an improvement!*

## Step 7: Commit the code to GitHub

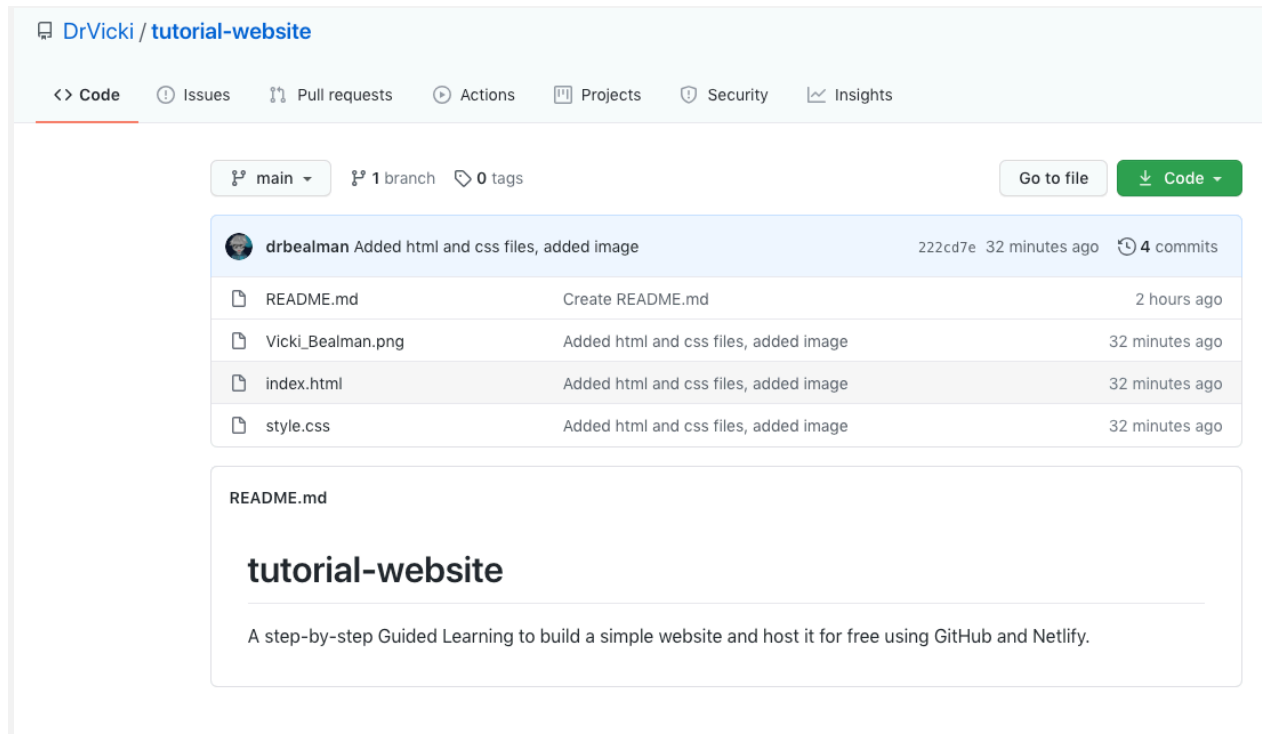
*As you work on your project, it's important to commit your code to your GitHub repository to make sure that you never lose any of your work. The most common way to work is to commit your code every time you finish a feature.*

*Let's go back to our terminal and commit our code.*

```
$ git add .  
$ git commit -m "write a message here that describes the change  
you've made to your project files"  
$ git push origin main:refs/heads/main
```

```
vbealman-mbp:tutorial-website vbealman$ git push origin main:refs/heads/main  
Username for 'https://github.com': DrVicki  
Password for 'https://DrVicki@github.com':  
Enumerating objects: 6, done.  
Counting objects: 100% (6/6), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (5/5), done.  
Writing objects: 100% (5/5), 95.29 KiB | 19.06 MiB/s, done.  
Total 5 (delta 0), reused 0 (delta 0)  
To https://github.com/DrVicki/tutorial-website.git  
    c7780dd..222cd7e  main -> main  
vbealman-mbp:tutorial-website vbealman$
```

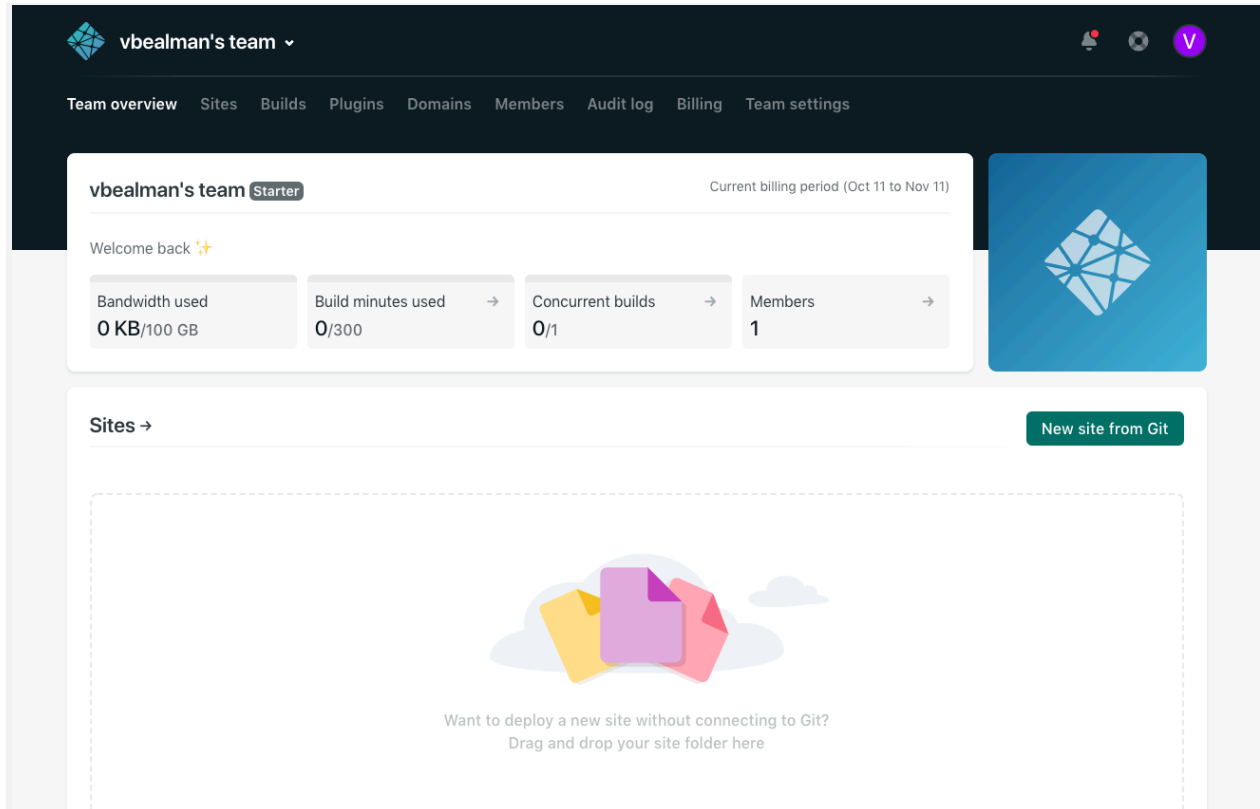
*Now we can go to our GitHub repository and our files will be there.*



*We can see our new files, and we can see the commit message which should include information about the changes we are committing to our project.*

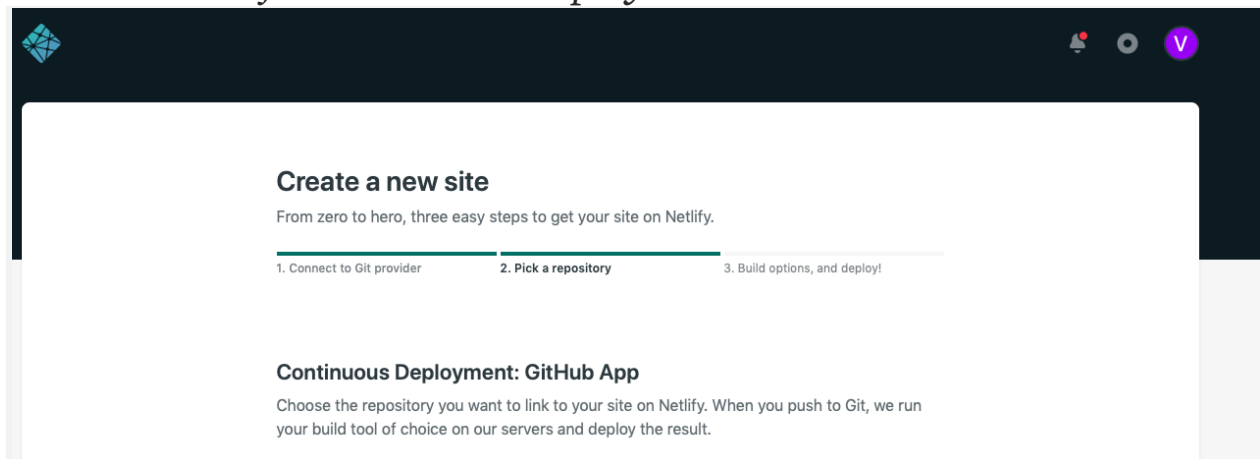
## Step 8: Make it live! Let's deploy our website

- *Sign up for a Netlify account [here](#)*
- *I used my GitHub login to sign-up, makes it easy and quick*



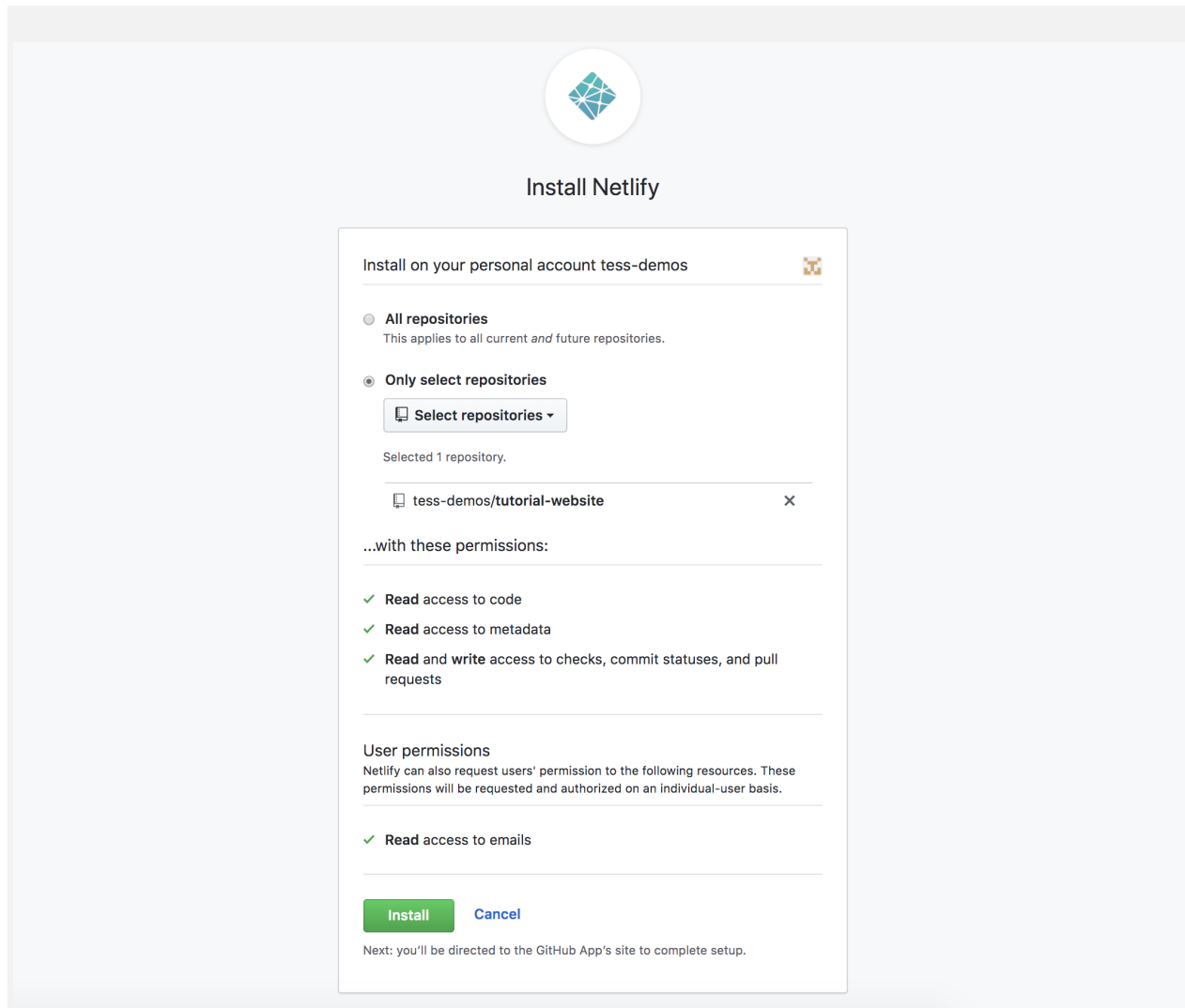
*From the landing dashboard, click on “New site from Git”*

*Select GitHub for continuous deployment.*



*Chose whether you want to allow Netlify to access all your repositories, or select only specific repositories, and click “install”.*



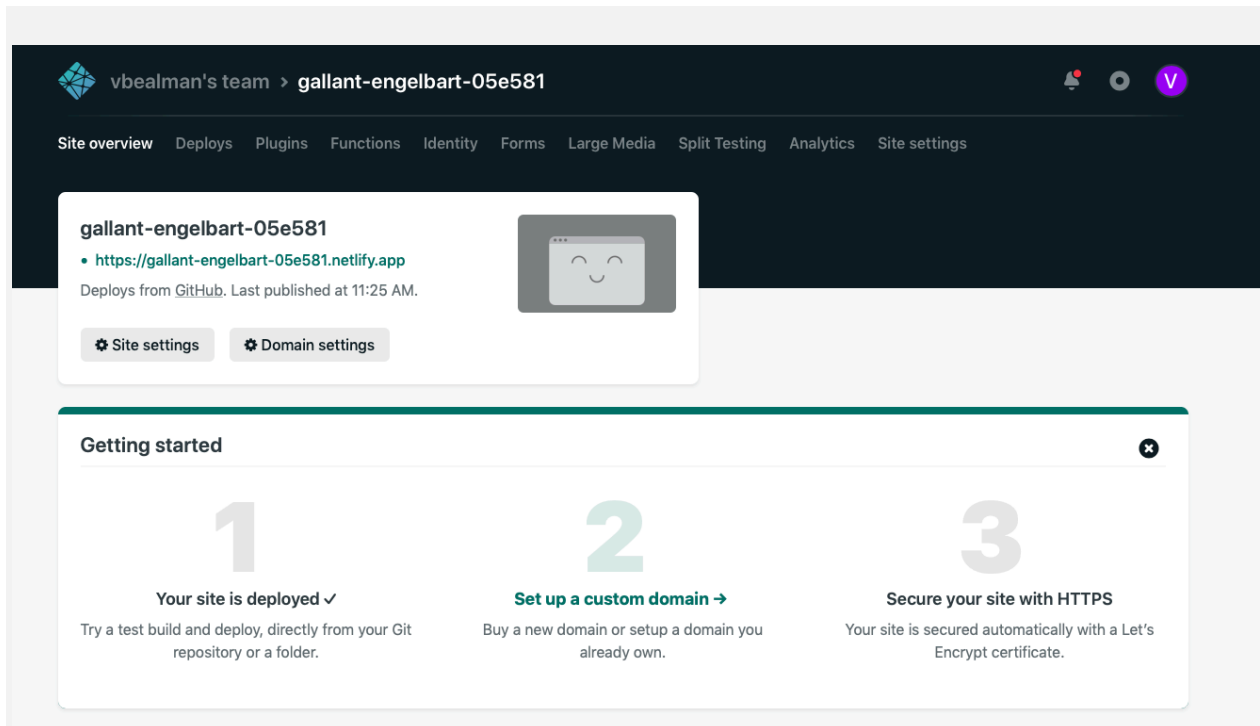


*Go back to the Netlify dashboard and select your project repository:*

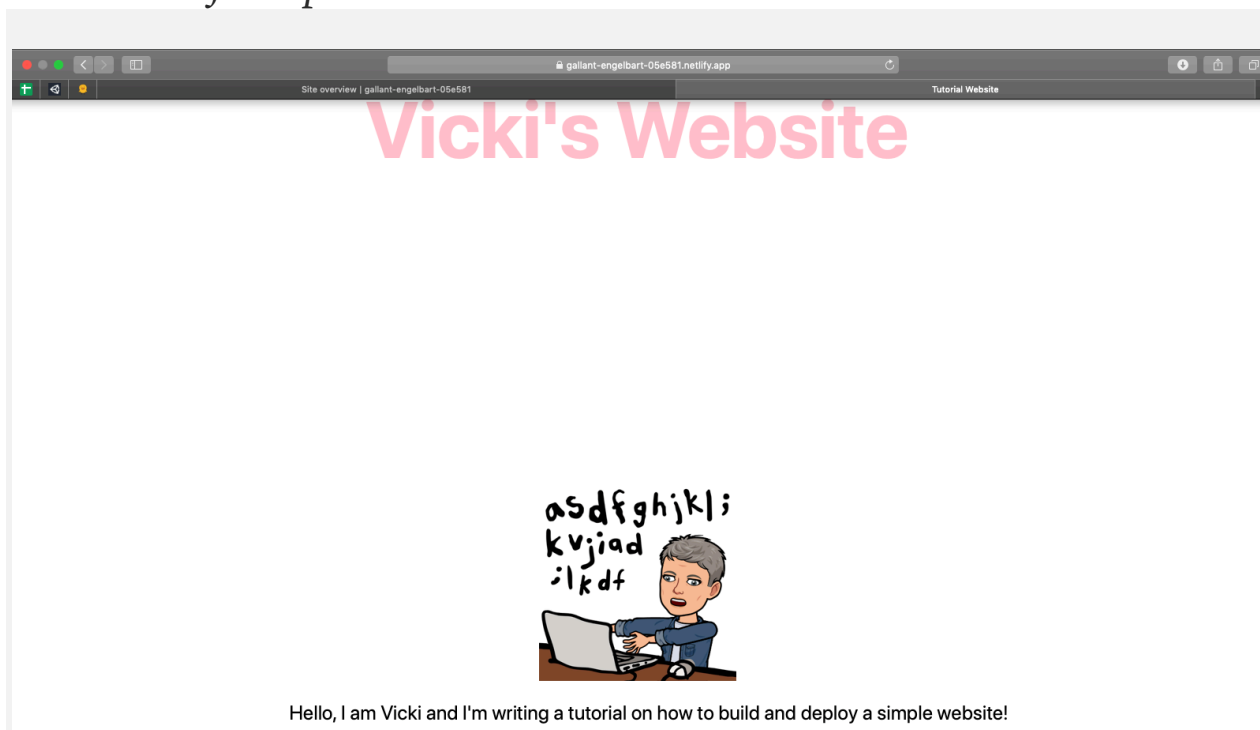
*Double-check the details, and click “Deploy site”.*

*We are live baby!*

*Your URL will consist of a strange generated Netlify domain. Here is the final product: <https://gallant-engelbart-05e581.netlify.app>*



*Here is the final product:*



*I hope you enjoyed following this guided learning. The fun part now is to play around with the code and make it your own!*

*Thanks for following along, I hope it was useful!*

WRITTEN BY

Vicki Bealman