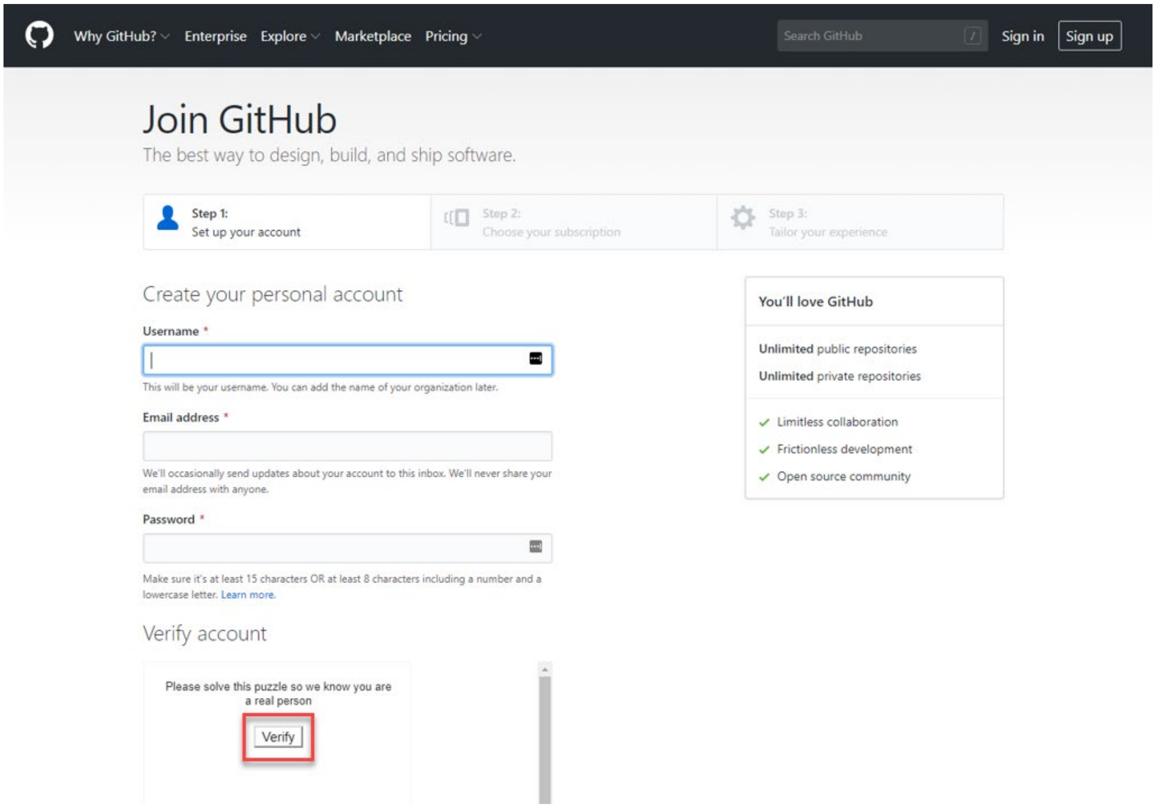
Southern
New Hampshire
University

CS 465 GitHub Repository Tutorial

Setting Up Your Account

If you have not done so already, go to GitHub and set up an account. **Note**: You must register for a GitHub account using your SNHU email address. Using your SNHU email will configure your account with the academic policy automatically. GitHub will ask you to create a username, add your email address, and create a password. When finished, click **Verify**.



Join GitHub

The best way to design, build, and ship software.

Step 1: Set up your account

Step 2: Choose your subscription

Step 3: Tailor your experience

Create your personal account

Username *

This will be your username. You can add the name of your organization later.

Email address *

We'll occasionally send updates about your account to this inbox. We'll never share your email address with anyone.

Password *

Make sure it's at least 15 characters OR at least 8 characters including a number and a lowercase letter. Learn more.

Verify account

Please solve this puzzle so we know you are a real person

You'll love GitHub

Unlimited public repositories

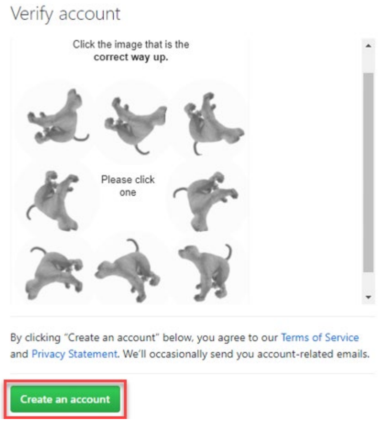
Unlimited private repositories

✓ Limitless collaboration

✓ Frictionless development

✓ Open source community

When you click **Verify**, a puzzle will appear, and you will be asked to solve it. The puzzle will likely be different from the puzzle in this example. After you answer the question correctly, you will be able to click on the **Create an account** button.



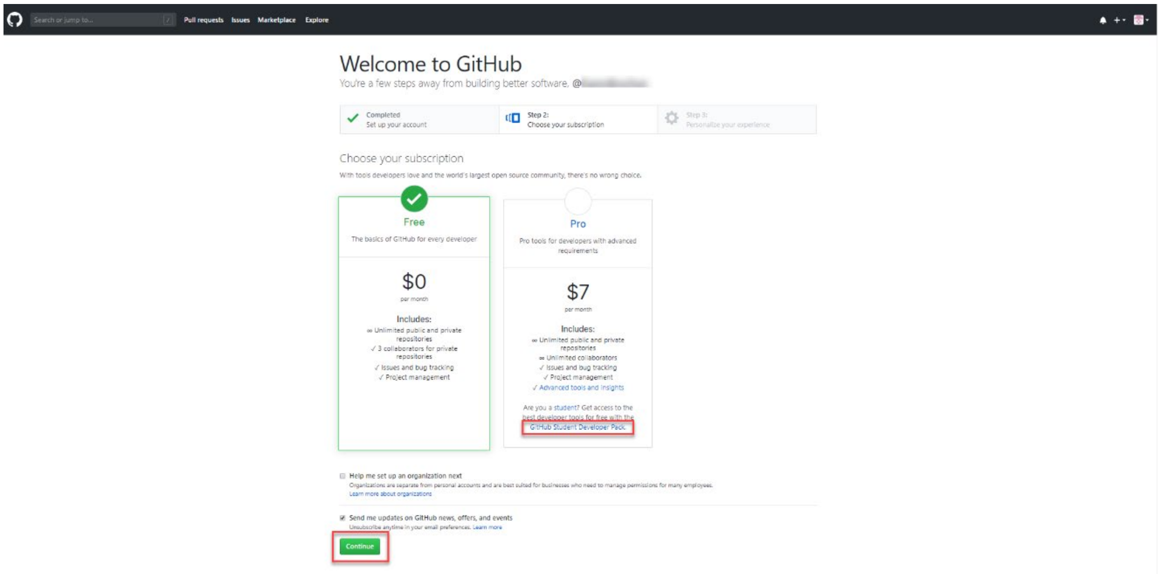
Verify account

Click the image that is the **correct way up.**

Please click one

By clicking "Create an account" below, you agree to our Terms of Service and Privacy Statement. We'll occasionally send you account-related emails.

You will be taken to a welcome page, where you can select your account type. For the purposes of this class, you only need the free option. After you make your selection, click **Continue**.



Welcome to GitHub

You're a few steps away from building better software, @[redacted]

✓ Completed
Set up your account

Step 2:
Choose your subscription

Step 3:
Personalize your experience

Choose your subscription

With tools developers love and the world's largest open source community, there's no wrong choice.

Free
The basics of GitHub for every developer
$0
per month
Includes:
∞ Unlimited public and private repositories
✓ 3 collaborators for private repositories
✓ Issues and bug tracking
✓ Project management

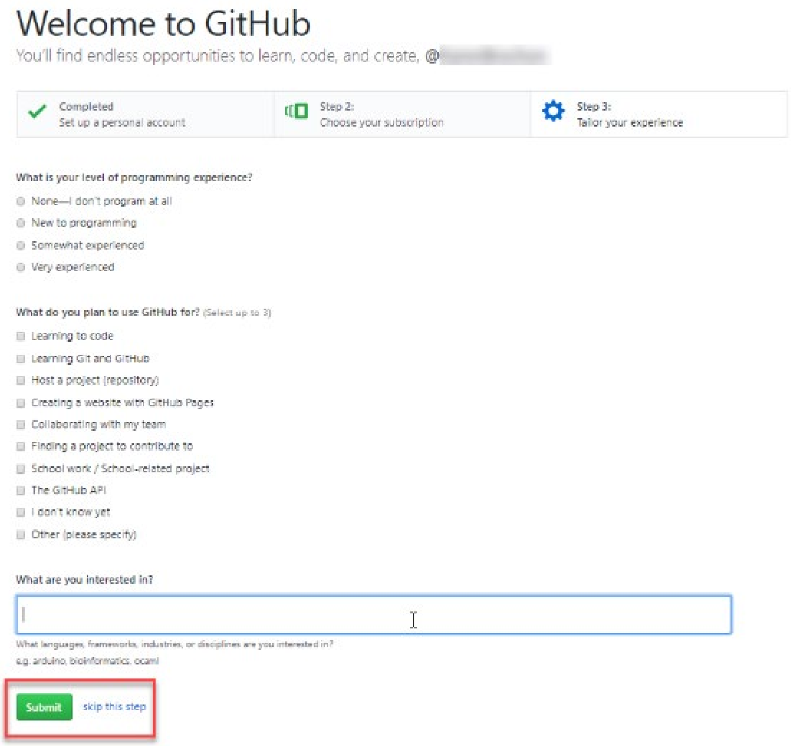
Pro
Pro tools for developers with advanced requirements
$7
per month
Includes:
∞ Unlimited public and private repositories
∞ Unlimited collaborators
✓ Issues and bug tracking
✓ Project management
✓ Advanced tools and insights
Are you a student? Get access to the best developer tools for free with the GitHub Student Developer Pack.

□ Help me set up an organization next
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.
Learn more about organizations

☑ Send me updates on GitHub news, offers, and events
Unsubscribe anytime in your email preferences. Learn more

Continue

GitHub will send an email verification. Make sure to verify your account. On the next screen, you will be asked questions about how you intend to use GitHub. You may choose to fill out the form and click **Submit**, or you may choose **Skip this step**.



Welcome to GitHub

You'll find endless opportunities to learn, code, and create, @[redacted]

✓ Completed
Set up a personal account

Step 2:
Choose your subscription

Step 3:
Tailor your experience

What is your level of programming experience?
○ None—I don't program at all
○ New to programming
○ Somewhat experienced
○ Very experienced

What do you plan to use GitHub for? (Select up to 3)
□ Learning to code
□ Learning Git and GitHub
□ Host a project (repository)
□ Creating a website with GitHub Pages
□ Collaborating with my team
□ Finding a project to contribute to
□ School work / School-related project
□ The GitHub API
□ I don't know yet
□ Other (please specify)

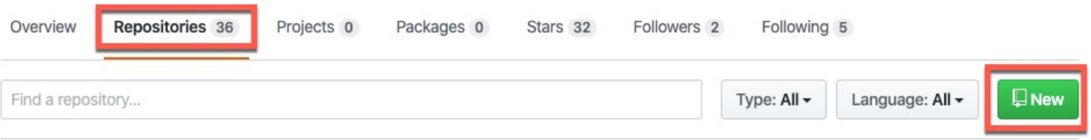
What are you interested in?

What languages, frameworks, industries, or disciplines are you interested in?
e.g. arduino, bioinformatics, ocaml

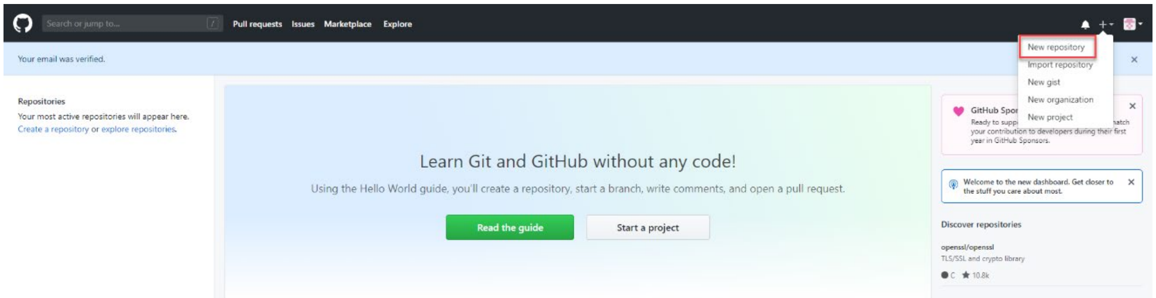
Submit skip this step

Creating Your GitHub Repository

To create a new repository, first sign into your GitHub account. On the home page, click the **Repositories** tab and then choose **New**.



If you do not have any repositories yet or have just created a new GitHub account, look for the small **plus sign** in the top-right corner of the home page screen and click on it. Select **New repository** from the drop-down list.



Your email was verified.

Repositories

Your most active repositories will appear here.
Create a repository or explore repositories.

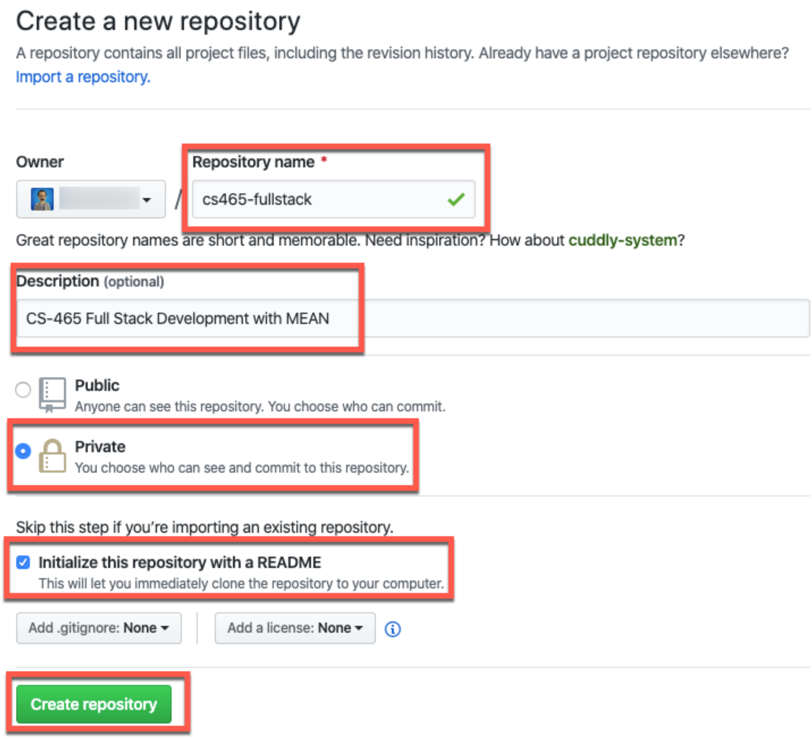
Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

GitHub Spor
Ready to supp ... atch your contribution to developers during their first year in GitHub Sponsors.

Welcome to the new dashboard. Get closer to the stuff you care about most.

The **Create a new repository** screen will be displayed. You will be prompted to enter a repository name and description, select the visibility of the repository, and choose whether to initialize the repository with a README. Fill out the prompts as shown.



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Owner /

Repository name * cs465-fullstack ✓

Great repository names are short and memorable. Need inspiration? How about cuddly-system?

Description (optional): CS-465 Full Stack Development with MEAN

○ Public
Anyone can see this repository. You choose who can commit.

● Private
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☑ Initialize this repository with a README
This will let you immediately clone the repository to your computer.

Add .gitignore: None

Add a license: None

Creating a Local Git Repository From the Remote GitHub Repository

In a PowerShell command window, make certain that the current working directory is your user directory by typing the following command and pressing **Enter**:

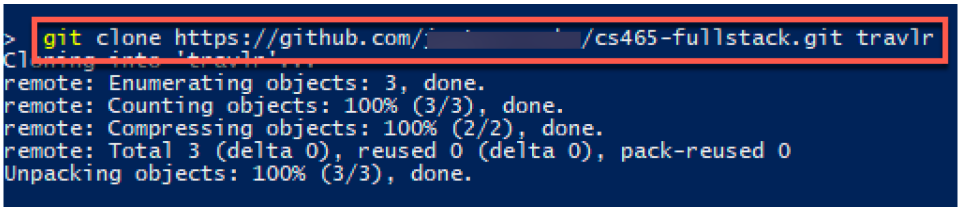
cd ~

```
> cd ~
```

Note: The tilde character (~) is shorthand for your user directory.

Clone the remote GitHub repository into a new directory in your user folder by typing the following command and pressing **Enter**. View the Cloning a Repository webpage to learn more.

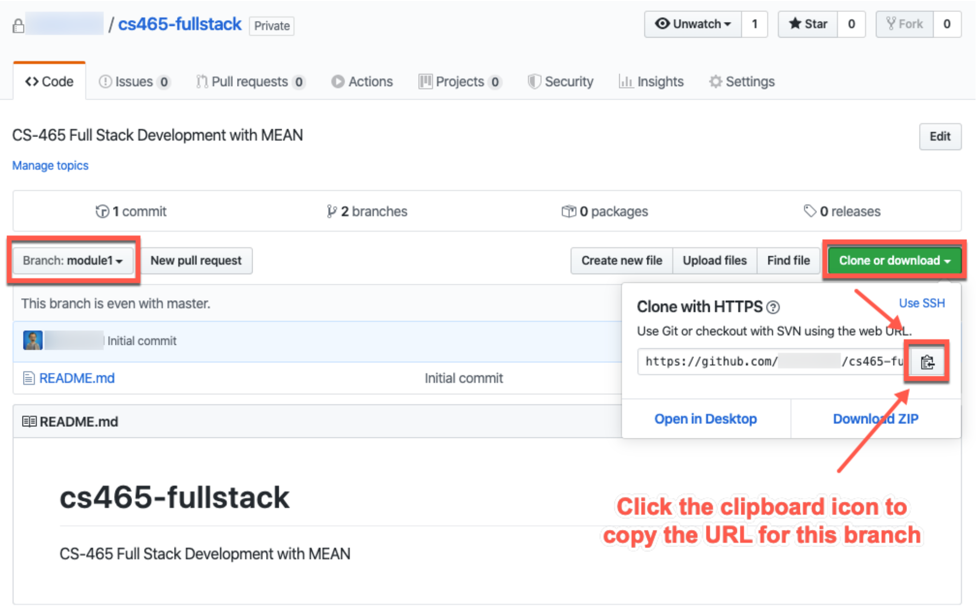
git clone https://github.com/GITHUB-LOGIN/cs465-fullstack.git travlr



```
> git clone https://github.com/[redacted]/cs465-fullstack.git travlr
Cloning into 'travlr'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

Note: The command has four parts, each separated by a space. You must replace GITHUB-LOGIN shown above with your unique GitHub login name.

Alternatively, you can copy the URL of your repository by getting it from the GitHub webpage where you just created the new branch. On the right side of the page, click the **Clone or download** button and then click the **clipboard icon** to copy the URL for this brand.



CS-465 Full Stack Development with MEAN

This branch is even with master.

[redacted] Initial commit

README.md Initial commit

Use Git or checkout with SVN using the web URL.

https://github.com/[redacted]/cs465-fu

cs465-fullstack

CS-465 Full Stack Development with MEAN

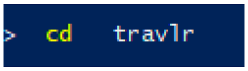
Click the clipboard icon to copy the URL for this branch

After typing “git clone” with a trailing space afterward, right-click your mouse anywhere in the PowerShell window to paste the URL from the clipboard at the end of the command. Then you only need to type another space and then “travlr”. The actions would be the following command:

git clone <space> <paste URL> <space> <travlr>

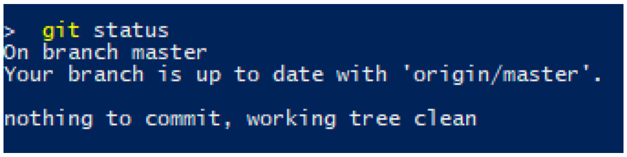
Change the working directory to **travlr.**

cd travlr



```
> cd   travlr
```

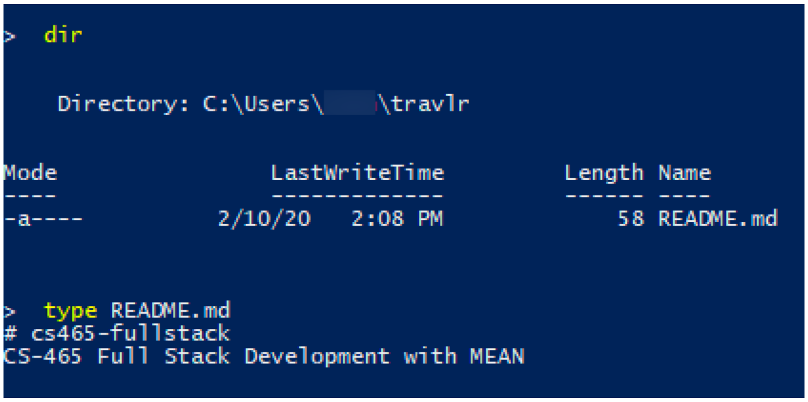
Display the status of your local Git repository by using the **git status** command.



```
> git status
On branch master
Your branch is up to date with 'origin/master'.

nothing to commit, working tree clean
```

At this point, you have a local Git repository on your computer that is linked to your remote GitHub repository. You can verify that the **travlr** directory has the files cloned from the GitHub repository with the following two commands:

dir
type README.md

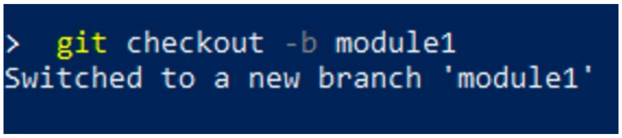


```
> dir


    Directory: C:\Users\[redacted]\travlr


Mode                LastWriteTime         Length Name
----                -------------         ------ ----
-a----        2/10/20   2:08 PM             58 README.md


> type README.md
# cs465-fullstack
CS-465 Full Stack Development with MEAN
```

Creating a New Branch in Your Git Repository

Before you begin working on a new module, create a new branch to track your changes to the code and website. When creating a new branch, it is important to understand that the branch you have currently checked out will be the basis for the new branch. For example, to make a new branch for module1 work, enter the following command:

git checkout -b module1



```
>  git checkout -b module1
Switched to a new branch 'module1'
```

Notice the **-b** in the above command indicates a new branch is being created.

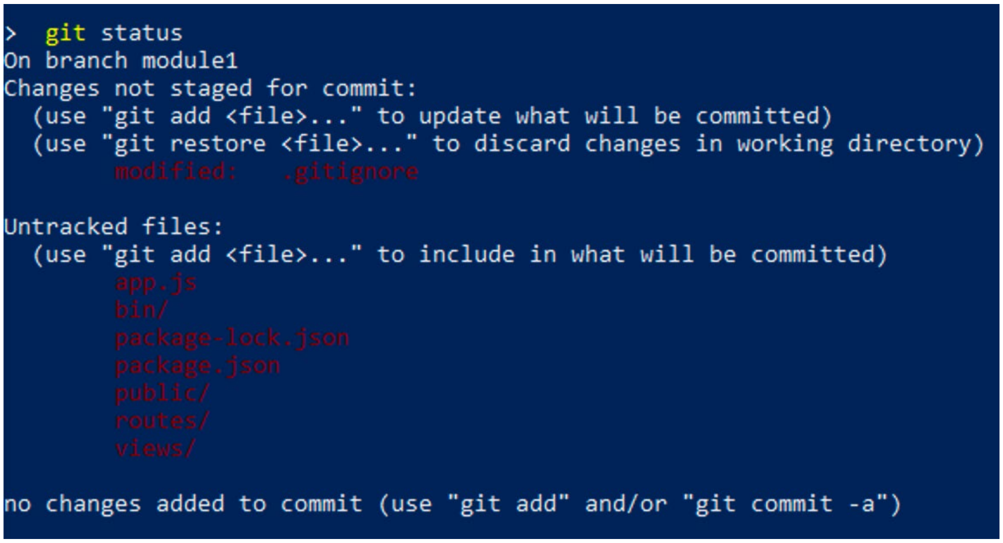
Tip: Do **not** put embedded spaces into your folder or branch names. The space character is often used to delimit different parts of a command, so when you put spaces into names, you are forever forced to use escape sequences to work with those names. If you want to separate a name into distinct parts, the recommended method is to use dashes or underscores such as “module-1” or “module_1”. Remember, this is programming code you are working with, not an English paper!

Checking the Status of Changes in Your Git Repository

Note: You cannot continue with this tutorial if you have not already completed the Module One Full Stack Guide. Stop and complete the Module One Full Stack Guide before continuing to the next step in this tutorial.

To see the status of all changes to your local Git repository, use the following command:

git status



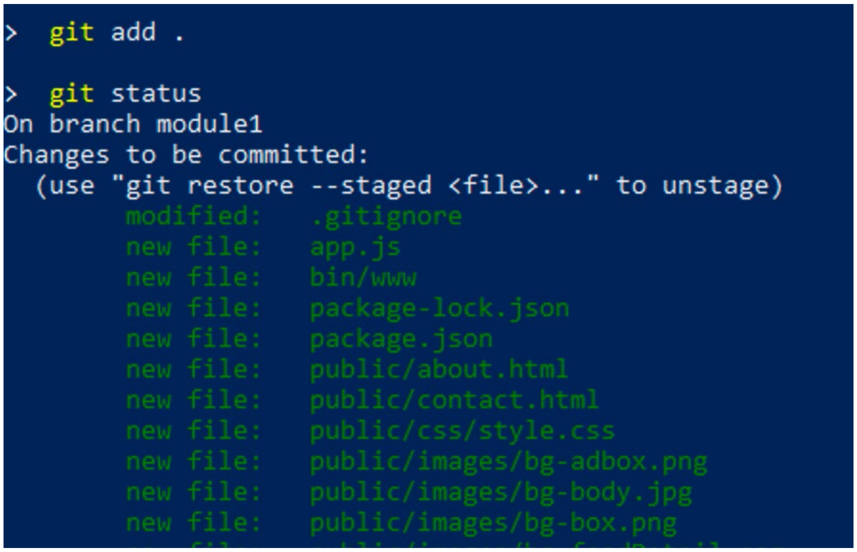
```
>  git status
On branch module1
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   .gitignore

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        app.js
        bin/
        package-lock.json
        package.json
        public/
        routes/
        views/

no changes added to commit (use "git add" and/or "git commit -a")
```

Notice the files in red font. These files are not tracked in the Git repository.

Staging and Committing Changes to Your Git Repository

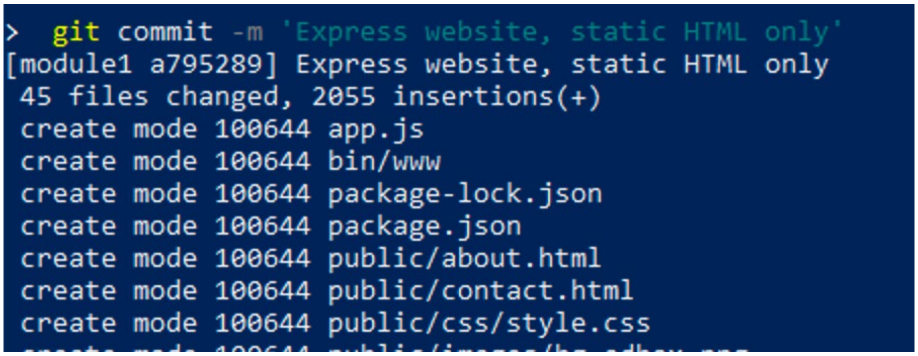
Stage all the new and modified files to be committed using the following command:

git add .



```
>  git add .

>  git status
On branch module1
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   .gitignore
        new file:   app.js
        new file:   bin/www
        new file:   package-lock.json
        new file:   package.json
        new file:   public/about.html
        new file:   public/contact.html
        new file:   public/css/style.css
        new file:   public/images/bg-adbox.png
        new file:   public/images/bg-body.jpg
        new file:   public/images/bg-box.png
```

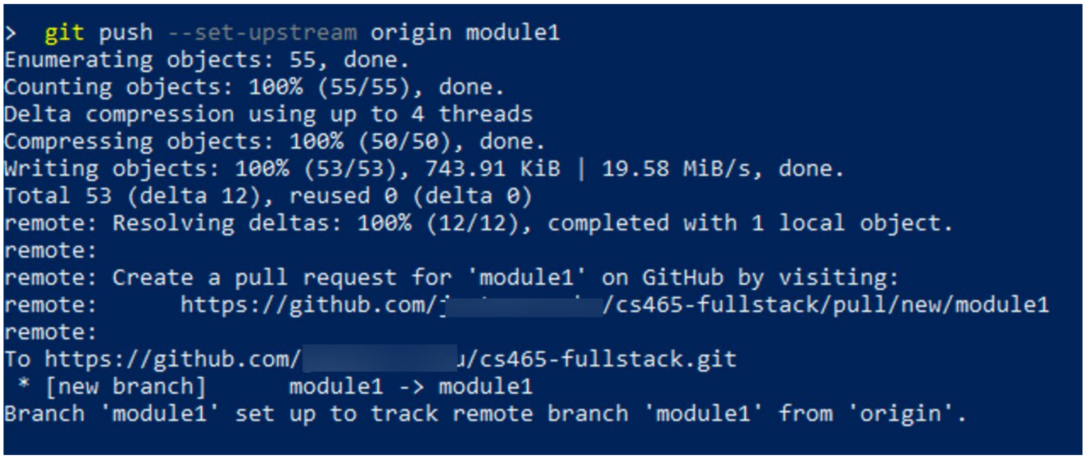
git commit -m ‘Express website, static HTML only’



```
>  git commit -m 'Express website, static HTML only'
[module1 a795289] Express website, static HTML only
 45 files changed, 2055 insertions(+)
 create mode 100644 app.js
 create mode 100644 bin/www
 create mode 100644 package-lock.json
 create mode 100644 package.json
 create mode 100644 public/about.html
 create mode 100644 public/contact.html
 create mode 100644 public/css/style.css
```

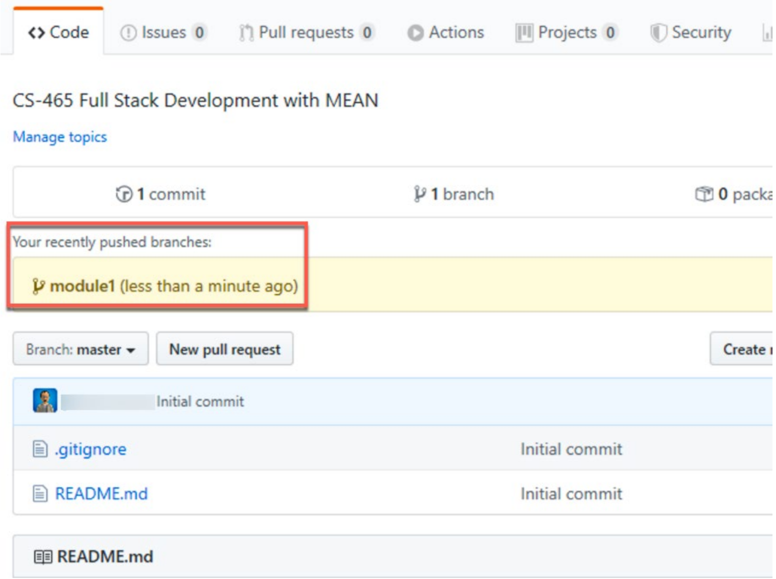
Uploading a Branch From Your Local Git Repository to a Remote GitHub Repository
To send your local committed changes to a remote repository, use the following command:

git push --set-upstream origin module1



```
>  git push --set-upstream origin module1
Enumerating objects: 55, done.
Counting objects: 100% (55/55), done.
Delta compression using up to 4 threads
Compressing objects: 100% (50/50), done.
Writing objects: 100% (53/53), 743.91 KiB | 19.58 MiB/s, done.
Total 53 (delta 12), reused 0 (delta 0)
remote: Resolving deltas: 100% (12/12), completed with 1 local object.
remote:
remote: Create a pull request for 'module1' on GitHub by visiting:
remote:      https://github.com/[redacted]/cs465-fullstack/pull/new/module1
remote:
To https://github.com/[redacted]/cs465-fullstack.git
 * [new branch]      module1 -> module1
Branch 'module1' set up to track remote branch 'module1' from 'origin'.
```

When you switch over to GitHub, you will see that the recent push is detected, and the new branch has been added.



CS-465 Full Stack Development with MEAN

Your recently pushed branches:

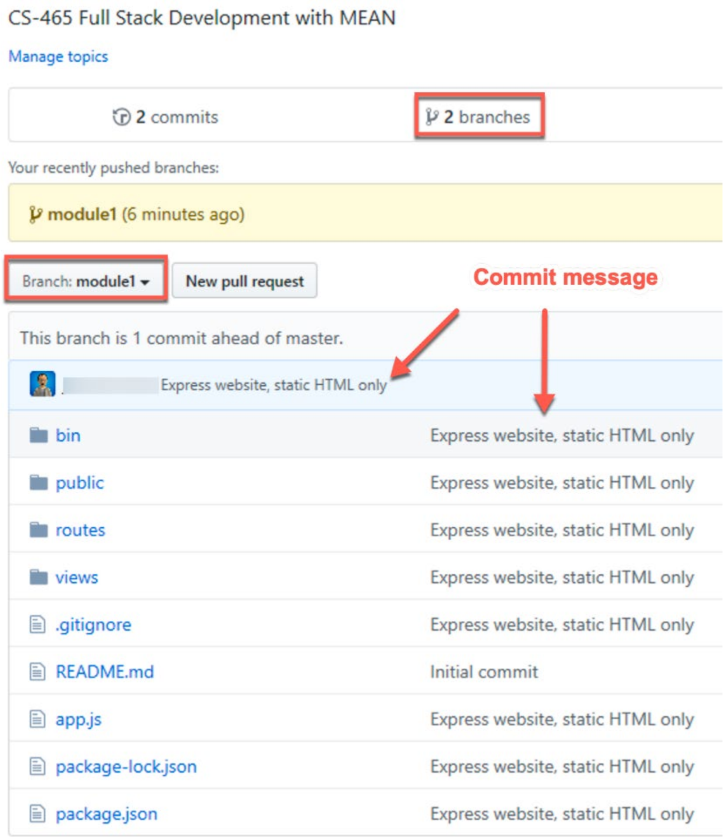
module1 (less than a minute ago)

[redacted] Initial commit

.gitignore	Initial commit
README.md	Initial commit

README.md

You can now choose the **module1** branch and see your Express website files there.


CS-465 Full Stack Development with MEAN

Your recently pushed branches:

module1 (6 minutes ago)

This branch is 1 commit ahead of master.

[redacted] Express website, static HTML only

bin	Express website, static HTML only
public	Express website, static HTML only
routes	Express website, static HTML only
views	Express website, static HTML only
.gitignore	Express website, static HTML only
README.md	Initial commit
app.js	Express website, static HTML only
package-lock.json	Express website, static HTML only
package.json	Express website, static HTML only

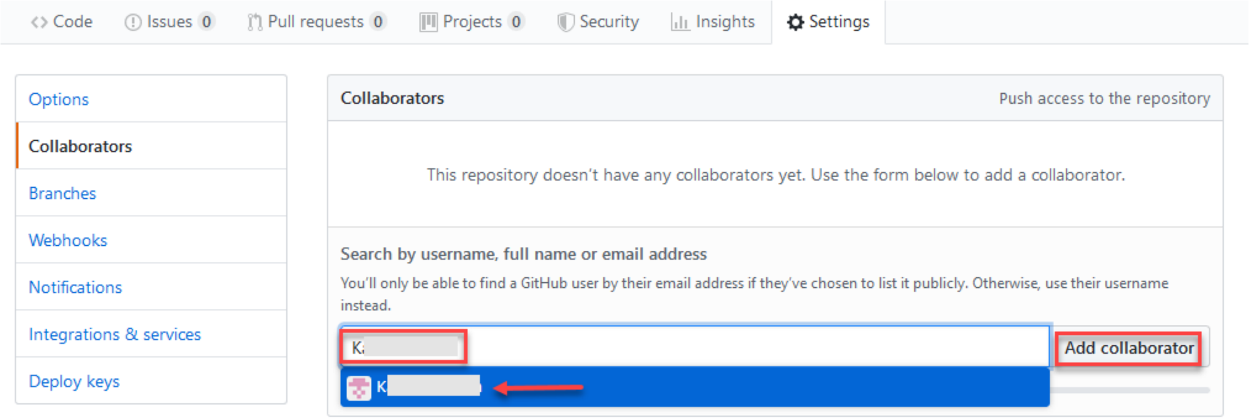
Adding a Collaborator

For your instructor to view and grade your portfolio, you must add them as a collaborator to your repository. Follow the instructions on the Inviting Collaborators to a Personal Repository webpage.

From the sidebar menu, click the **Collaborators** link to open the Collaborators page.

For security, you will be prompted to confirm your password. Once you have confirmed your password, you will be taken to the **Collaborators** page. Look for **Search by username, full name or email address** and enter your instructor's GitHub username in the box. Once you have typed their name in the box, their username should appear below the search. Click on the username, then click **Add collaborator**.

Note: Your instructor should have provided their username in an announcement or discussion post. If you are unsure of their username, please email your instructor.



Collaborators — Push access to the repository

This repository doesn't have any collaborators yet. Use the form below to add a collaborator.

Search by username, full name or email address

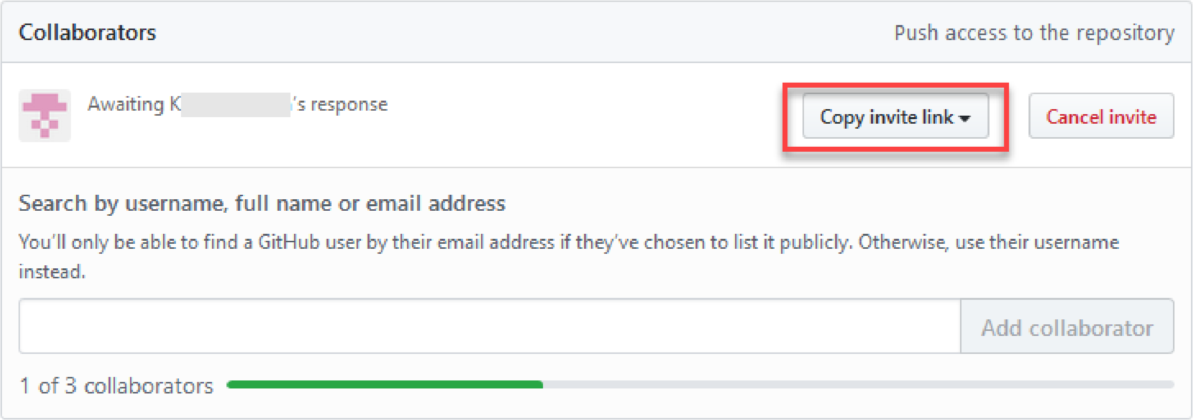
You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

K[redacted]

K[redacted]

Add collaborator

Once you have completed this step, you should see your instructor's username listed under **Collaborators**. The webpage will say, "Awaiting [username]'s response." You will also see a button labeled **Copy invite link**. Click this button.



Collaborators — Push access to the repository

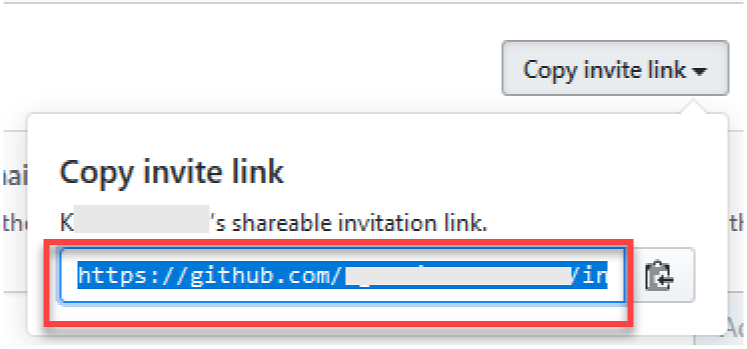
Awaiting K[redacted]'s response

Search by username, full name or email address

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

1 of 3 collaborators

A pop-up message with a URL for the invitation link will appear. Click on the text to highlight it, as shown in the image. Press **CTRL+C** to copy the link to your clipboard. Then paste your link into the indicated box and submit it to your instructor.



Copy invite link

K[redacted]'s shareable invitation link.

https://github.com/[redacted]/in