Build Your Portfolio: Website with Quarto + GitHub Pages (Part 2)

What You'll Learn

This guide will walk you through creating your very own website using Quarto and GitHub Pages.
By the end, you'll have a working link like:

`https://yourusername.github.io/my-portfolio/`

That link will open a real website — built by you — that lives on the internet and shows off your skills, projects, and learning journey.

You'll learn how to:

- Set up a personal website using **Quarto** (a free site builder)
- Edit that site using either **RStudio** or **VS Code**
- Use **GitHub** to save and publish your site online
- Understand how folders (directories) work and how to keep your work organized

You do **not** need to know how to code websites or use the command line before starting this. We'll walk through everything step-by-step.

What is Quarto, and Why Are We Using It?

Think of Quarto as a smart notebook that turns plain text into a beautiful website.
It's like writing in Google Docs, but Quarto turns your writing into something much fancier — a public site you can show off.

Why Quarto is Great for Beginners

- You don't need to write any code to create pages
- It's designed for students, educators, and researchers
- It works well with data (tables, plots, code) if you decide to include that later
- It's completely free and works on both **Windows** and **Mac**

When you create a Quarto project, you get a folder (like a digital backpack) with all your web pages inside. Quarto helps you organize them and turn them into something beautiful.

Choose a Tool to Work In

Before we start building, we need to pick a workspace — the app you'll use to write and edit your website files.

You can use either of these tools:

Tool	Best For	You'll like this if…
RStudio	Quarto + R users	You've used R before, or want menus & buttons instead of code
VS Code	General coding	You like working in plain text editors and want more control

If you're not sure, start with **RStudio**. It's beginner-friendly and we'll show lots of screenshots for it.

A Quick Note on Folders & Files (a.k.a. "Directories")

Before we jump in, let's explain something really important:

When we say things like "create a project folder" or "save it in Documents," we're talking about **directories** — another word for *folders* on your computer.

Here's what that might look like:

```
Documents/
  DS101/
     my-portfolio/        ← this is your Quarto website folder
         index.qmd        ← homepage file
         about.qmd        ← about me page
         _quarto.yml      ← controls the website layout
```

When you "open a project" in RStudio or VS Code, you're just telling the app,
"Hey, this is the folder I'm working in — please treat it like my workspace."

That's why it's important to **keep everything inside one folder** — it keeps your project organized and avoids errors later when publishing.

Option A — Using RStudio

What is RStudio?

RStudio is a tool that helps you write and preview Quarto websites.
You can think of it like Microsoft Word, but designed for coding and website building.
It includes menus, tabs, and buttons to help you stay organized.

Step 1: Create a New Quarto Website Project

1. Open **RStudio**
2. Click on **File**   **New Project**
3. Choose **New Directory**
4. Select **Quarto Website**

Now fill in the details:

- **Directory name** = `my-portfolio` (you can name it whatever you want)
- **Create project as a subdirectory of** = choose where to save it, like `Documents/DS101`

  This will create a **new folder** called `my-portfolio` on your computer.
  Inside that folder, you'll find all the files you need to build your website.

Click **Create Project**.

RStudio will now switch into that folder and open it as your project workspace.
You'll see some files appear in the Files pane (bottom right).

Step 2: What's Inside Your Website Folder?

After you create the project, your folder will contain:

File	What it does
`index.qmd`	Your homepage. This is the first page visitors will see.
`about.qmd`	A second page about you — you can write a short bio here.
`_quarto.yml`	A config file that controls your site's name and navigation bar.

Quarto uses these files to build your website. You can always rename or add more pages later.

Step 3: Preview Your Website

Let's see what your site looks like right now.

In RStudio, click the **Render** button at the top.
After a few seconds, a browser window should pop up — and you'll see your site!

It might look simple, but that's a real website being created from `.qmd` files (Quarto Markdown files).

> If rendering doesn't work, try this:
>
> ```r
> install.packages("quarto")
> ```

Step 4: Edit Your Homepage and About Page

Let's personalize your site.

Change Your Homepage (`index.qmd`)

- Open `index.qmd` (just click the file name in RStudio)
- You'll see something like this:

```
title: "Quarto Website"
format: html
---

Welcome to your Quarto website.
```

Change the title and the message:

```
title: "Brian's Data Portfolio"
format: html
---

Hi there! I'm a first-year student learning data, storytelling, and coding.
```

Click **Render** again to see your update live.

Write Your About Page (`about.qmd`)

- This is a great place to talk about:
  - What school you go to
  - What major or interests you have
  - What you're excited to learn this term

Write in plain language — this is your space to shine!

Up Next: Once you're happy with how your site looks, we'll connect it to GitHub and make it live on the internet!

~~~
##  Option B - Using VS Code

> If you prefer a clean, code-focused workspace or plan to write in multiple languages (like

VS Code is a free, flexible code editor that works on all major operating systems. You’ll use

---

### Step 1: Install Quarto CLI (Command Line Interface)

Before we can build the website, we need to install **Quarto** - the tool that converts your

1. Open your browser and visit: [https://quarto.org/download](https://quarto.org/download)
2. Click the download that matches your computer:
   - **Windows** → `.exe` file
   - **Mac** → `.pkg` file
   - **Linux** → `.deb` or `.rpm` package depending on your system

3. After downloading, open the file and go through the installer (you can accept all the defa

Once it's installed, we’ll double-check that it works.

---

###  Check That Quarto Installed Correctly

1. Open a **terminal**.
   - Windows users: open **Git Bash** or search for **Terminal** in the Start Menu.
   - Mac users: press <kbd>  + Space</kbd>, type "Terminal," and hit <kbd>Enter</kbd>.

2. In the terminal, type:

```bash
quarto check
~~~

If everything worked, you should see a version number (e.g., `Quarto Version 1.4.550`) and some green checkmarks.

> If you get a “command not found” error, Quarto wasn’t installed properly. Ask for help in Canvas!

Step 2: Create a New Website Project

Let's make your new website!

In your terminal:

```
quarto create-project my-portfolio --type website
```

This creates a folder called `my-portfolio` and places three important files inside. Next, navigate into that folder:

```
cd my-portfolio
```

To **open the folder in VS Code**, just type:

```
code .
```

This tells VS Code to launch and open the current folder — the one with your website files inside.

> **Quick note on folders ("directories")**:
> When we say "your project folder", we mean a folder on your computer where all your website files live together. Always keep your project files in one place!

Step 3: Understand the Files

Once VS Code opens, look at the file list in the sidebar. You should see:

File	What it does
`index.qmd`	Your homepage — this is where visitors land first
`about.qmd`	A page where you can introduce yourself
`_quarto.yml`	This controls your site's layout, navigation, and theme

These are plain text files — just click them to open and edit. Quarto uses `.qmd` files (Quarto Markdown) to create pages.

Step 4: Preview Your Site

Let's see what your site looks like before publishing.

In the terminal (at the bottom of VS Code), type:

```
quarto preview
```

This starts a local web server. A browser tab will open with your live preview. As you make changes to the files and save them, the preview will update automatically.

> You can leave `quarto preview` running in the background while you work. Every time you save, your changes appear right away in the browser!

Connect to GitHub & Publish Your Site

You've created your site — now let's publish it to the internet using **GitHub Pages**.

Step 1: Make a GitHub Repository

1. Go to https://github.com
2. Log in to your GitHub account
3. Click the + icon (top-right corner) → choose **New repository**
4. Name your repository something like `my-portfolio`
5. Keep the default settings:
    - **Public**
    - Do NOT check "Add README"
6. Click **Create repository**

GitHub will take you to a new page with setup instructions.

Step 2: Link Your Project Folder to GitHub

In the terminal **(inside your portfolio folder)**, run the following:

```
git init
git add .
git commit -m "Initial site"
git branch -M main
git remote add origin https://github.com/YOURUSERNAME/my-portfolio.git
git push -u origin main
```

> Replace `YOURUSERNAME` with your real GitHub username.

These commands do a few things: - `git init` → tells Git to track this folder - `git add .` → stages all files for upload - `git commit` → makes a save point - `git push` → uploads it to GitHub

---

Step 3: Turn on GitHub Pages

Now that your site is on GitHub, let's make it live!

1. In your GitHub repo, click **Settings**
2. In the left sidebar, click **Pages**
3. Under **Source**, choose:
   - **Branch** = `main`
   - **Folder** = `/ (root)`
4. Click **Save**

After ~30 seconds, GitHub will show a green success message:

```
Your site is live at https://yourusername.github.io/my-portfolio/
```

Congratulations — your site is now on the internet!

Make It Yours

Now that your site is live, let's personalize it.

Update the Homepage (`index.qmd`)

Open `index.qmd` and change the title and message. Example:

```
title: "Nia's Data Journey"
format: html
---

Welcome! I'm a first-year student learning how to use data and tell stories.
```

Add More Pages

Want a page for your projects?

1. Create a new file called `projects.qmd`
2. Add a title and description:

```
title: "Projects"
format: html
---

Here's a list of class projects I’ve worked on so far.
```

3. Link it in `_quarto.yml` under the navigation bar:

```yaml
navbar:
  left:
    - text: "Home"
      file: index.qmd
    - text: "About"
      file: about.qmd
    - text: "Projects"
      file: projects.qmd
```

You can add as many pages as you want! Just remember to update `_quarto.yml`.

Updating Your Site

Every time you make changes:

1. Save your files in VS Code
2. In the terminal, run:

```
git add .
git commit -m "Update homepage"
git push
```

Within ~1 minute, your live website will update at the same link!

Final Check

Can you visit your website at `yourusername.github.io/my-portfolio`?
Did you change the title and homepage message?
Did you try editing a second page (like `about.qmd`)?
Did you push the changes to GitHub?

If yes — you're officially a published website owner!

What's Next?

- Add your GitHub Pages link to your LinkedIn or resume
- Keep adding content as you grow your skills
- Submit this for **Extra Credit** → Extra Credit: Portfolio Website

---

Need Help?

- Post in the Canvas discussion **#portfolio-help**
- Bring your questions to lab time
- Ask a TA or classmate (we've all had to debug our first push)

You've just taken your first step toward building your online presence as a student in data science. Well done!