

Homework 5

Build and publish a website with GitHub Pages. Many labs, projects, and papers share their code, data, and results this way, and it is also a good home for your class project results.

You will make a small site that has:

- a home page written in **Markdown**
- one page written by hand in **HTML**
- one page that is **generated by a Python script** from a data file

This is due Nov 24, 2026. Submit the URL of your website and the URL of its GitHub repository in Canvas.

See the [Building Websites](#) lecture notes and the GitHub Pages [quickstart](#) for background.

Your website is public. Don't put unpublished lab data, passwords, or personal information you don't want online on it.

Part 1: Create the repository and turn on GitHub Pages

1. On github.com create a new **public** repository in your own account named `YOURGITHUBID.github.io` (use your exact GitHub username). A repository with this special name is served at `https://YOURGITHUBID.github.io/`.

If you already have a site with that name, make a repository called `gen220-website` instead; it will be served at `https://YOURGITHUBID.github.io/gen220-website/`.

2. Clone it to the HPCC cluster or your laptop:

```
git clone git@github.com:YOURGITHUBID/YOURGITHUBID.github.io.git
cd YOURGITHUBID.github.io
```

3. On github.com go to the repository **Settings** -> **Pages**. Under **Build and deployment**, set **Source** to *Deploy from a branch*, then choose branch `main` and folder `/` (root) and click **Save**.
4. Create a file called `_config.yml` which tells GitHub Pages how to build the site. This picks a theme and gives the site a title:

```
title: YOUR NAME
description: GEN220 website
theme: minima
```

Other themes you can try are listed at <https://pages.github.com/themes/> (e.g. `jeekyll-theme-cayman`).

Part 2: A home page in Markdown

Create `index.md`. This becomes `index.html`, the front page of your site. It should include **all** of the following Markdown features:

- ☐ a top level heading (`#`) and at least two section headings (`##`)
- ☐ a paragraph about you and your research interests (or a research topic you find interesting)
- ☐ a bulleted or numbered list
- ☐ **bold** and *italic* text
- ☐ a link to an outside site (e.g. your lab, a paper, a database you use)
- ☐ an image stored in your repository (e.g. put it in an `images/` folder) - use a picture you took or made, or one you have permission to use
- ☐ a table with at least 3 rows
- ☐ a block of code in a fenced code block with a language tag (e.g. a short bash or python example from class)
- ☐ links to your other two pages (`about.html` from Part 3 and `data.html` from Part 4)

Start the file with a front matter block so the theme knows the page title:

```
---
title: Home
```

```
---  
  
# Hello, I'm YOUR NAME
```

I am a graduate student in ...

Add, commit, and push:

```
git add _config.yml index.md images/  
git commit -m "Add home page"  
git push
```

Wait a minute or two, then open your site URL. You can watch the build under the **Actions** tab of your repository - if it fails, click on the failed run to see the error message.

Part 3: A page written in HTML

Create a file called `about.html` by writing the HTML yourself (don't copy the output of a website builder). A plain HTML file with no front matter is served exactly as you wrote it, without the theme. It must be a complete, valid page with:

- ☐ `<!DOCTYPE html>`, plus `<html>`, `<head>` (containing a `<title>`), and `<body>` sections
- ☐ a heading (`<h1>`) and at least one paragraph (`<p>`)
- ☐ a list (`<ul>` or `<ol>` with `<li>` items)
- ☐ a link (`<a href="...">`) back to your home page and one to an outside site
- ☐ an image (``) - the `alt` text describes the image for people using screen readers
- ☐ a little CSS styling in a `<style>` block in the `<head>` (e.g. change the font, a color, or center the page)

Here is a skeleton to start from:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="utf-8">  
  <title>About me</title>  
  <style>  
    body { font-family: sans-serif; max-width: 40em; margin: auto; }  
  </style>  
</head>  
<body>  
  <h1>About me</h1>  
  <p>Write something here.</p>  
  <p><a href="index.html">Back to the home page</a></p>  
</body>  
</html>
```

To check your HTML, open the file in a web browser on your laptop before you push it, and paste it into the W3C validator at https://validator.w3.org/#validate_by_input. Fix any errors it reports.

Part 4: A page generated from data with Python

Websites are great for sharing results, and results should come from code, not be typed by hand. Write a script called `make_table.py` which reads a data file and **writes** a Markdown page called `data.md`, which GitHub Pages turns into `data.html`.

This starter script uses the `threatened-species.csv.gz` file you used in Homework 1 (from https://github.com/biodataprogram/GEN220_data/raw/main/tabular/threatened-species.csv.gz) and counts the species in each kingdom:

```
#!/usr/bin/env python3
# Summarize threatened species by kingdom and write a Markdown table
import csv
import gzip

infile = "threatened-species.csv.gz"
outfile = "data.md"

counts = {}
with gzip.open(infile, "rt") as fh:
    reader = csv.DictReader(fh)
    for row in reader:
        kingdom = row["kingdom_name"]
        counts[kingdom] = counts.get(kingdom, 0) + 1

with open(outfile, "w") as out:
    out.write("---\ntitle: Threatened species\n---\n\n")
    out.write("# Threatened species by kingdom\n\n")
    out.write("| Kingdom | Species |\n")
    out.write("| :----- | -----: |\n")
    for kingdom in sorted(counts, key=counts.get, reverse=True):
        out.write(f"| {kingdom} | {counts[kingdom]} |\n")
```

Extend it so that `data.md` contains:

- ☐ a sentence explaining where the data came from, with a link to the source
- ☐ **at least two** summary tables. Ideas: the number of species in each IUCN category (e.g. CR, EN, VU) for one kingdom, or the 10 plant families with the most threatened species
- ☐ a sentence or two describing what the tables show

You can use a different data set instead - for example data from your own research that you are allowed to share, or one from the [GEN220_data](#) repository. Tell me in your README what you used.

Commit the script **and** the `data.md` it produced (you don't need to commit the data file if it is large, but explain how to download it in the README).

```
curl -LO https://github.com/biodataprogram/GEN220_data/raw/main/tabular/threatened-species.csv.gz
python3 make_table.py
git add make_table.py data.md
git commit -m "Add data summary page"
git push
```

Part 5: README

Add a `README.md` to the repository that explains:

- ☐ the URL of the website
- ☐ what each file in the repository is for
- ☐ how to re-create `data.md` (which data to download and what command to run)

Note that GitHub Pages will also turn `README.md` into a page; that is fine.

Bonus (pick any)

- Make a plot of your data with Python (`matplotlib`) or R, save it as a PNG, and show it on `data.md`.
- Add a GitHub Actions workflow that runs `make_table.py` and rebuilds `data.md` whenever you push.
- Make the navigation consistent: add a menu of links to all your pages at the top of every page.
- Write a page for your class project that describes the question, the data, and your progress.

Tips

- File names are **case sensitive** on the web: `Images/Me.JPG` and `images/me.jpg` are different files.
- Use *relative* links (`data.html`, `images/me.jpg`), not links to files on your own computer (`/Users/me/Desktop/me.jpg` will not work for anyone else).
- In Markdown a link to another page on your site can point at the `.md` file or the `.html` file; GitHub Pages will fix up `.md` links for you.
- If a page doesn't update, check the **Actions** tab to see whether the build finished, then reload the page (browsers sometimes show an old copy; try Shift+Reload).
- A blank line is needed before a list or a table in Markdown or it may not render.

Grading

Part	Points
1. Repository and GitHub Pages site is live	15
2. Markdown home page with all required elements	25
3. Hand-written HTML page with all required elements	25
4. Script and generated data page	25
5. README	10
Total	100