

Creating a Portfolio Website with Static HTML and CSS

Documentation Prepared by: Jiya Palrecha (01FE21BCS094)

Date: 24 Oct. 2024

Premise:

This documentation will guide you in creating a learning portfolio for 'Design and Analysis of Algorithms' course. The portfolio will be presentation of project work and learning reflections from the course.

Step By Step Guide

Step 1: Create a GitHub Account

1. Visit github.com and sign up for an account if you don't already have one.
2. Once signed in, you'll see your GitHub dashboard.

Step 2: Create a New Repository


1. Go to your dashboard and click on the green **"New" button to create a new repository.**
2. Name the repository as **topic.github.io**, where topic is related to your portfolio (e.g., portfolio.github.io). This allows GitHub Pages to recognize it as a website repository.
3. Make the repository public.
4. Check the option to "Add a README file."
5. Click on "Create repository."

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Required fields are marked with an asterisk ().*

Owner *

 jiyapalrecha35


Repository name *

PortfolioDemo.github.io


✔ PortfolioDemo.github.io is available.

Great repository names are short and memorable. Need inspiration? How about [redesigned-meme](#) ?

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:

☒ **Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore


.gitignore template: **None**


Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: **None**

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

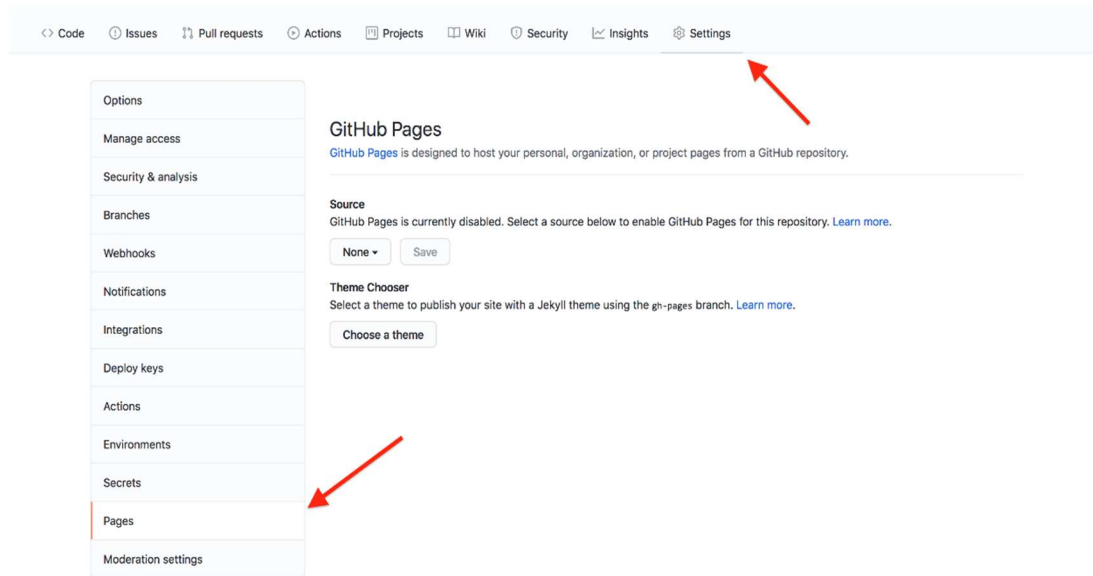
 You are creating a public repository in your personal account.

Create repository

Step 3: Enable GitHub Pages

1. In the repository created just now, **go to Settings**.
2. Scroll down to the **"Pages"** section on the left-hand side.

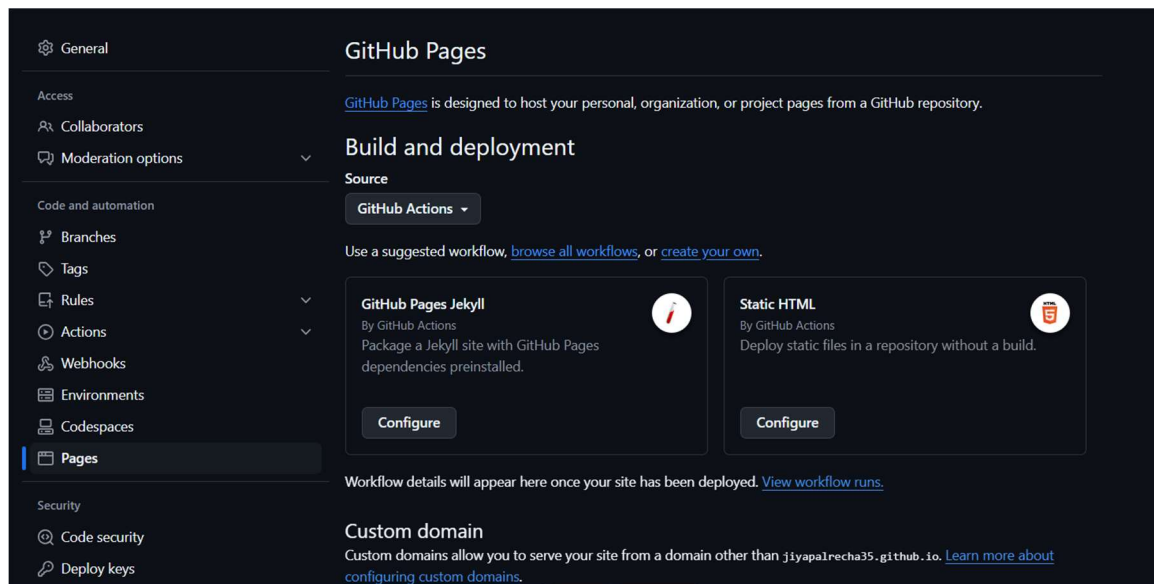
Creating Learning Portfolios



3. Under "Source", you'll see "GitHub Actions".

4. Below that, you'll find the "Static HTML" workflow on the right hand side.

5. Click on the "Configure" button next to it.



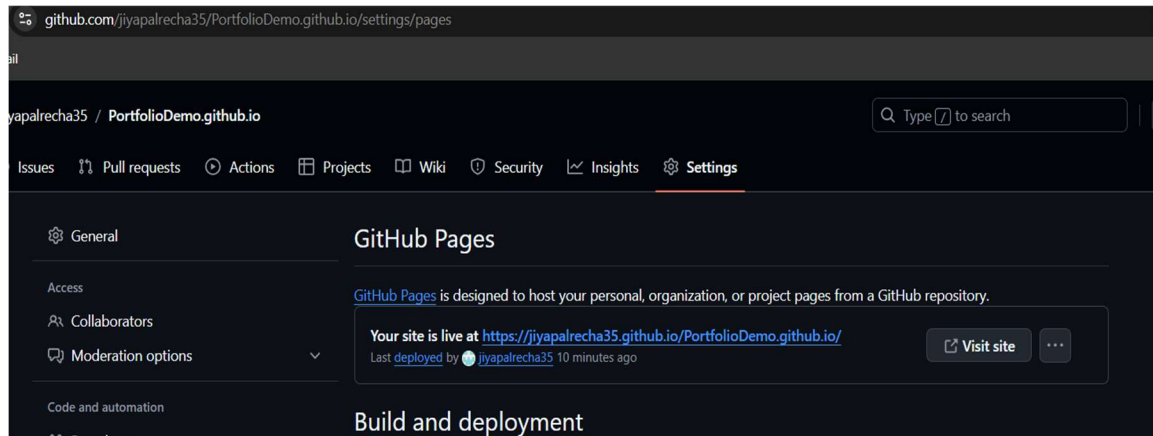
6. It will prompt you with a **static.yml** file. Commit the changes to set up the workflow.

7. The workflows are now set for automatic builds.

Creating Learning Portfolios

8. Go back to "Settings > Pages".

9. You will see a URL indicating your site's live link where your GitHub Pages site is active. (may take around 3-5 mins) (keep or refreshing until you find a link). Everytime you make a change, refresh the page and changes will be applied.



Step 4: Build Your Website with HTML and CSS

Create a new file called **index.html** in your repository. This will be your homepage. Add your HTML content inside **index.html** (file inside the drive link provided).

When you paste the HTML code into index.html and view the site, it will look something like this.

Hello, I'm [Your Name] 🙋

I am passionate about technology and problem-solving. Here's a glimpse of my work:

🔧 Projects

- **Project 1:** A short description of your project1.
- **Project 2:** A short description of another project2.

🚀 Skills

- C++, Python, JavaScript
- Data Structures and Algorithms
- Web Development (React.js, Node.js)

🌐 Find Me Online

- [GitHub](#)
- [LinkedIn](#)

Portfolio Topic

Course Name
Algorithmic Problem Solving
Course Code
Course Code Placeholder
Name
Your Name
SRN
Your SRN
Course Instructor
Instructor Name Placeholder
University
Your University Name
Portfolio Topic/Domain
Google

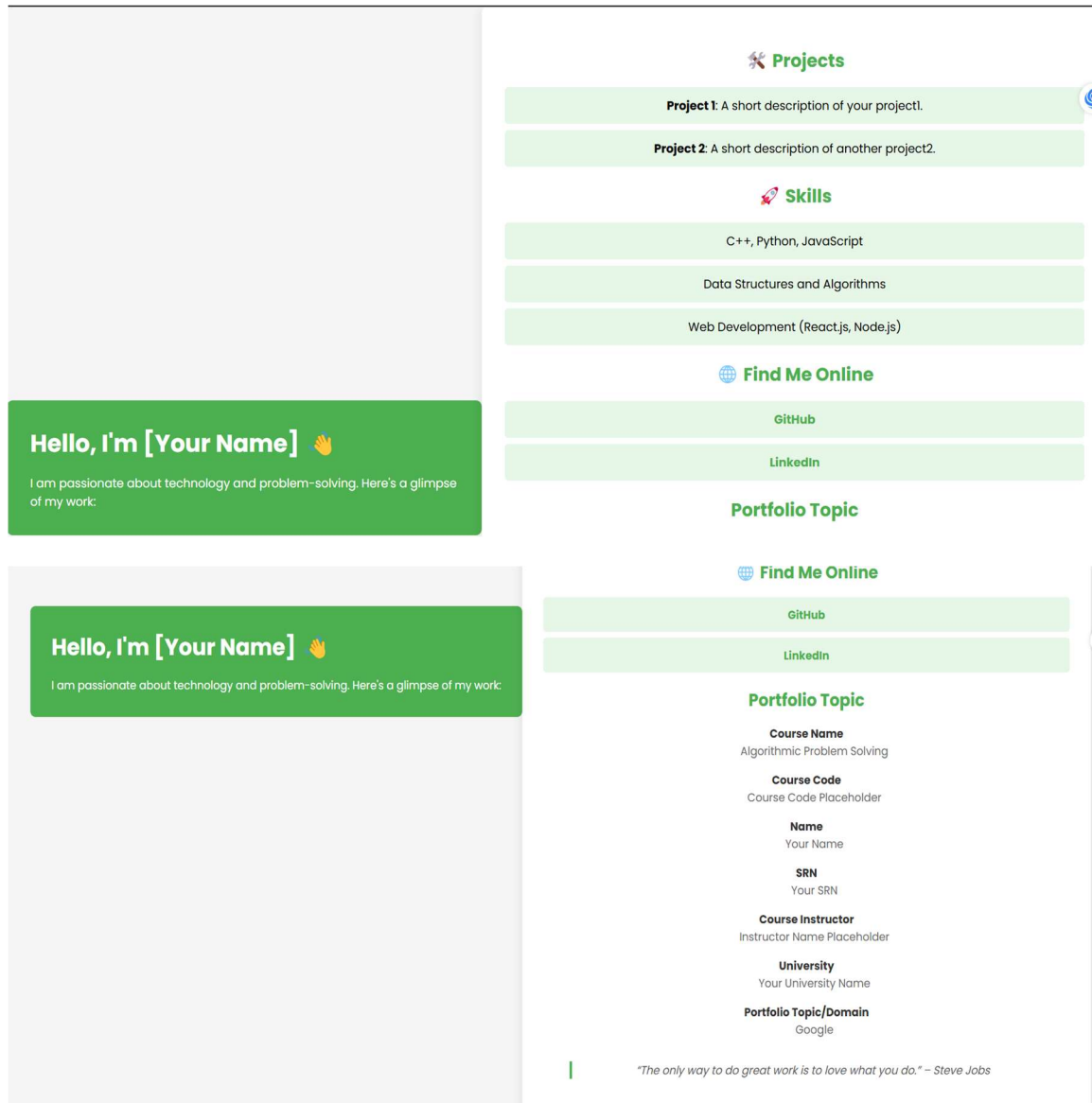
Step 5: Add Styling with CSS

Create a file named style.css to style your website.

Add some basic CSS. (Code in **style.css**)

You can customize the style.css and index.html file as per your need.

Once you paste the CSS code into style.css and view the site, it will look something like this.



Step 6: Add Images and Links

Create a folder named assets in your repository to store images. Add your image files in the assets folder, for example profile.jpg.

Reference your images in the HTML file as shown below:

```

```

To add links, you can use the following HTML code:

```
<a href="https://www.google.com" target="_blank">Visit Google</a>
```

Step 7: Commit and Push Changes

After editing the index.html and style.css files, commit and push your changes to the GitHub repository. Once you push the changes, GitHub Pages will automatically build your website, and the changes will be reflected on your live site in about 30 seconds.

Step 8: Add More Pages (Optional)

You can add more pages by creating new .html files and linking them from your main index.html.

For example, to add an 'About Me' page:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>About Me</title>
</head>
<body>
  <h1>About Me</h1>
  <p>This is the about me page. </p>
</body>
</html>
```

For linking it,

Add a link to it in index.html:

```
<a href="about.html">About Me</a>
```