

Hosting a Simple Personal Website on GitHub

User Guide

Introduction

Many people wish to create a personal website to showcase their achievements, experiences, and skills to potential employers or individuals interested in connecting with them. A personal website serves as a more detailed and dynamic version of a resume. It not only presents a professional image but also offers insights into recent interests, recent work, and engaging personal stories that often do not find a place on a traditional resume.

However, hosting a personal website using a cloud instance can be time-consuming and require dealing with complex networking tasks like port forwarding. This is where GitHub Pages comes in, offering a straightforward solution for hosting lightweight personal websites. For personal website use, GitHub Pages simplifies the setup of complex components, leaving you with the task of uploading a simple HTML file and, optionally, CSS to enhance its appearance. Think of it as having your own uncomplicated server for your personal website needs.

With this guide, we will walk you through the steps to harness the power of GitHub Pages, making it easy for you to create and maintain your personal website with minimal hassle.

Prerequisites

Before you embark on your journey to host a personal website on GitHub Pages, here are a few things you should keep in mind:

1. **GitHub Account (Optional but Recommended):** While not strictly necessary, it's highly recommended that you have an existing GitHub account. If you already have one, you can skip the step of creating a new GitHub account. If not, we'll guide you through the process.
2. **Basic HTML and CSS Knowledge:** To create and customize your website, some basic knowledge of HTML and CSS is **Required**. You should be comfortable either writing HTML/CSS files from scratch or, at the very least, editing content within HTML templates. Don't worry if you're not an expert; we'll provide guidance along the way.
3. **A Patient Mindset:** Building and hosting a website might seem daunting at first, but remember, many aspects that may appear challenging are quite manageable. Patience is key as you learn and grow during this process. We're here to help you every step of the way.

With these prerequisites in mind, you're ready to dive into the world of hosting your personal website on GitHub Pages!

Instructions

Setting Up a GitHub Account:

1. Sign up for GitHub:

If you don't already have a GitHub account, start by going to www.github.com. Click on the upper-right corner of the website to begin the registration (figure1.1). You will be prompted to enter your: Email address, create a password, choose a username, complete any additional required information. As part of the registration process, GitHub may require account verification. This typically involves solving a simple visual puzzle or a video puzzle. Examples are provided (figure1.2, 1.3). After completing the initial registration steps, GitHub will send you a verification email to the address you provided. Check your email inbox and locate the email from GitHub. Inside the email, click the "Verify Email Address" button to confirm your account. Return to the GitHub website after verifying your email. Follow any on-screen instructions to complete the registration process.

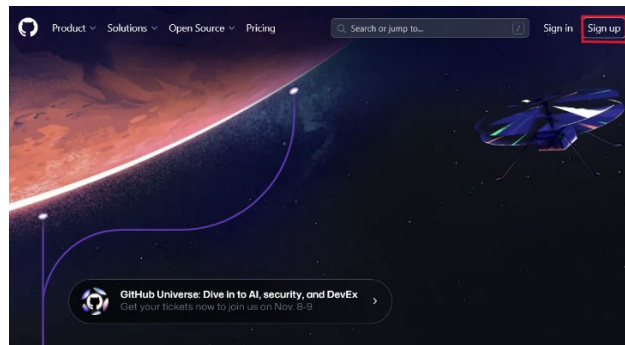


Figure 1.1

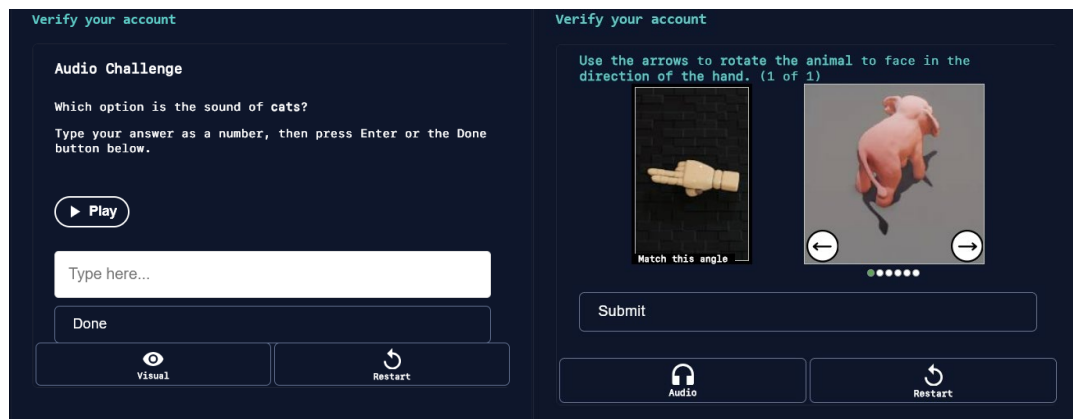


Figure 1.2

Figure 1.3

2. Creating a Repository:

- Every subpage on GitHub has a top menu with a 'plus' button (figure2.1). You can click this button to access a dropdown menu containing the 'Create New Repository' option. Click this option to be redirected to the repository creation page. In our case, a repository can simply be treated as storage or a placeholder for your HTML and CSS files.
- On the "Create a New Repository" page, you have the flexibility to modify specific details: Repository Name (a name for your repository that suits your project), Description (a brief description to offer context about your repository) (figure2.2).
- For now, refrain from making changes to any other settings unless you are already well-versed in GitHub's advanced features. The default settings should suffice for our current needs. Clicking 'create repository' will take you to the created repository page.
- **WARNING: Ensure that the "Repository" setting is configured as "Public" rather than "Private."** Generally, repositories are initially set to "Public," so there's often no need to modify this setting.

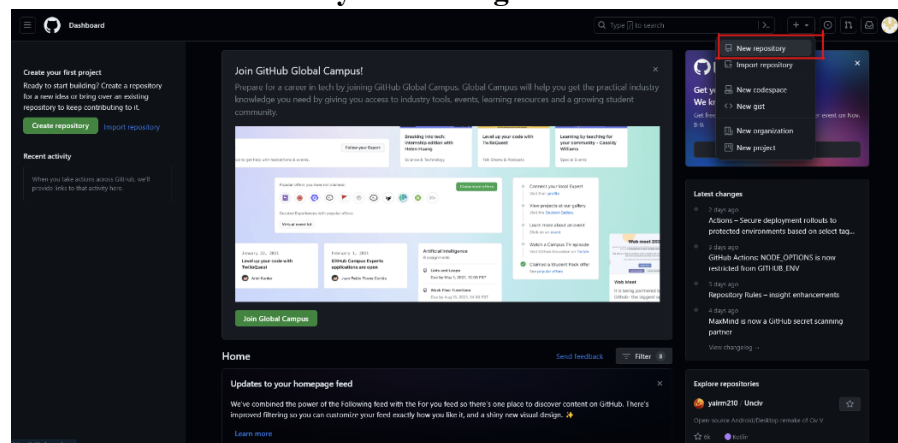


Figure 2.2

A screenshot of the 'Create a new repository' form on GitHub. The form has several sections. The 'Repository name' field is highlighted with a red box, and the 'Description' field is also highlighted with a red box. The 'Public' radio button is selected. The form includes fields for 'Owner', 'Repository name', 'Description', 'Visibility' (Public/Private), 'Initialize this repository with' (Add a README file), 'Add .gitignore', 'Choose a license', and a 'Create repository' button at the bottom right. A red arrow points to the 'Create repository' button.

Figure 2.3

3. Upload your HTML/CSS to the Repository:

- If you don't already have an HTML file, you can use the HTML template provided in the appendix as a trial. **Make sure to rename it to 'index.html'**
- On the page that displays the newly created repository, click 'Upload an Existing File,' and you will be redirected to a file upload page (figure 3.1).
- Next, click 'Choose Your Files' in the file selection window to find the HTML file you prepared earlier for uploading. Then, click 'Commit Changes.' (figure 3.2)

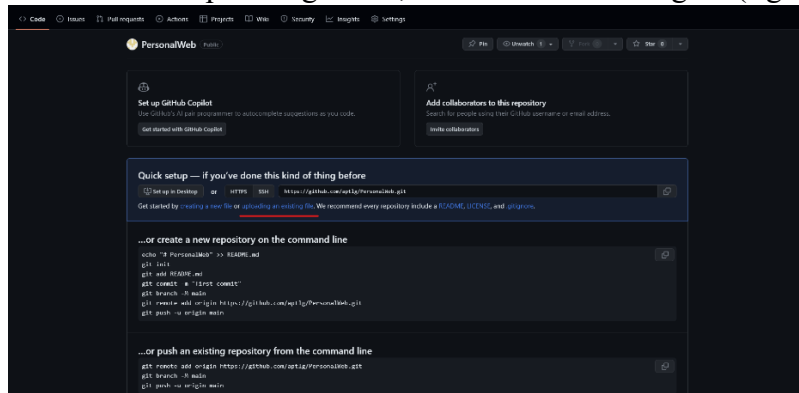


Figure 3.1

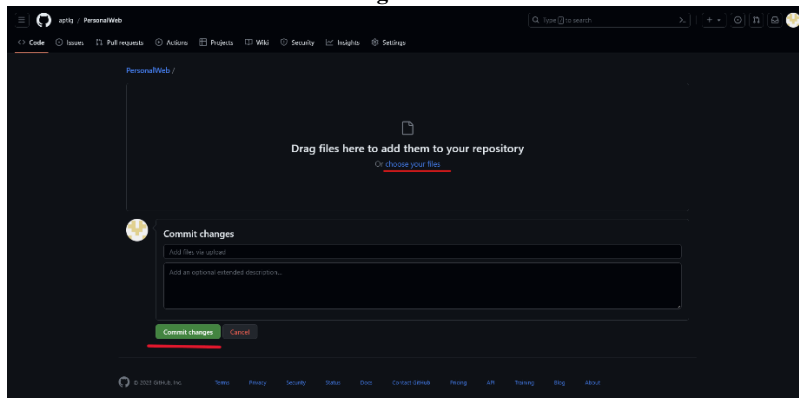


Figure 3.2

4. Configuring GitHub Pages

- Once you have successfully committed your changes to the repository, it's time to configure the GitHub Pages settings.
- Navigate to the repository's settings. You can do this by clicking on the "Settings" tab located within your repository. Within the settings menu, look for the "Pages" subtab and select it (figure 4.1).
- In the "Pages" settings, locate the "Branch" section. Initially, it might be set to "None." Change this setting to "main" to specify which version of your project you want to publish. (Note: Since you've just uploaded your project, "main" is typically the only branch available, and you don't need to worry too much about this unless you are working on a larger web project with multiple branches.) (figure 4.1)

- Once you've set the branch to "main," look for a "Save" button. Click this button to save your configuration.
- **Warning: Successful configuration should result in a page similar to the one shown below. Please refrain from altering any other options unless you are well-versed in GitHub's advanced features. (figure 4.2)**

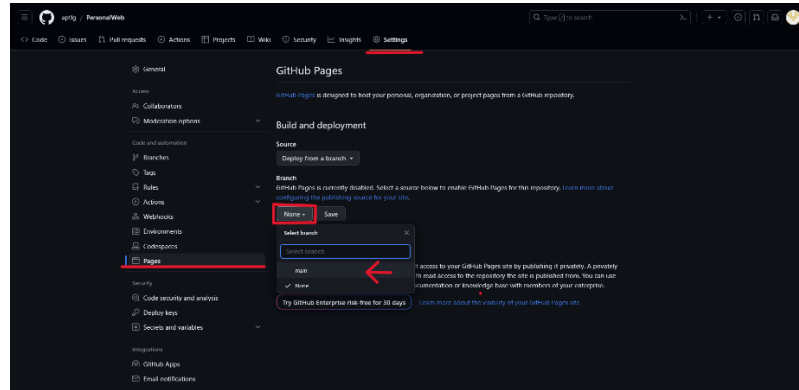


Figure 4.1

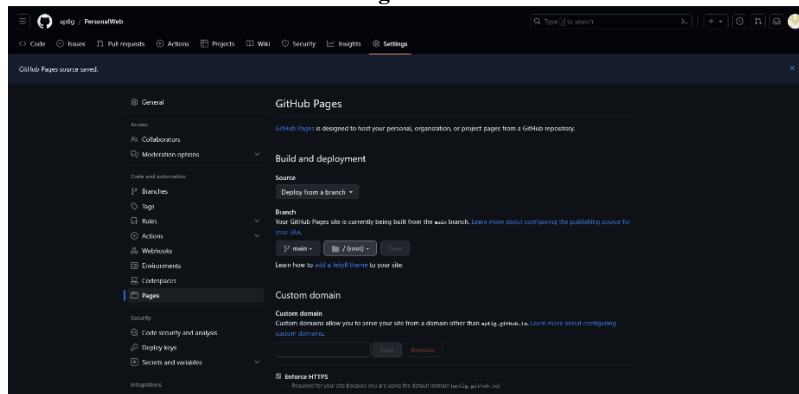


Figure 4.2

5. Accessing Your Website

- Congratulations! You have successfully published your first website on GitHub Pages. To access it, simply refresh the page, and you will see your web address (see Figure 5.1).
- A sample page is looks like (figure 5.2)

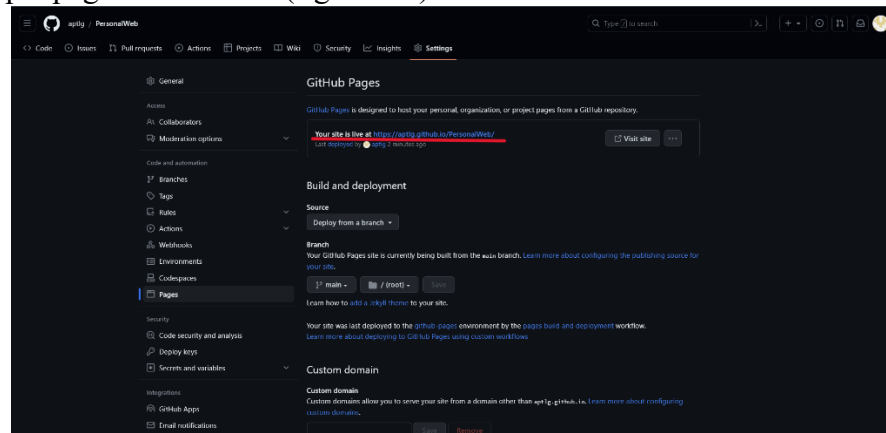


Figure 5.1

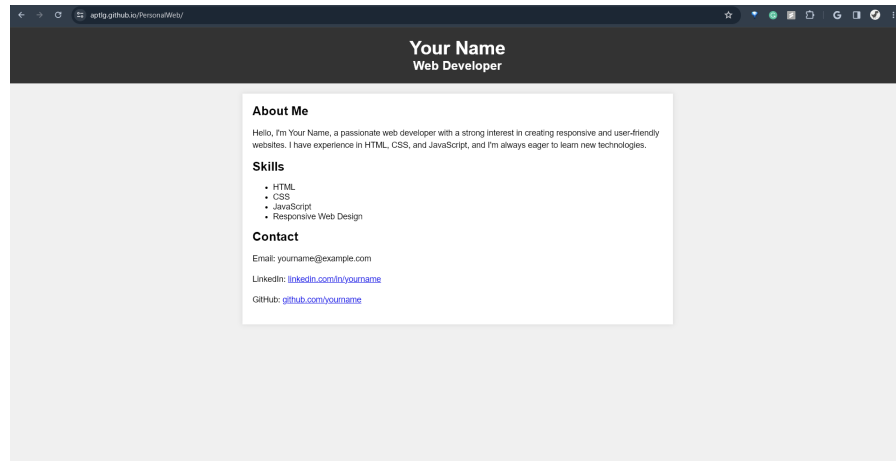


Figure 5.2

Appendix

index.html (sample)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Your Name - Personal Profile</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
      background-color: #f0f0f0;
    }

    header {
      background-color: #333;
      color: #fff;
      text-align: center;
      padding: 20px;
    }

    h1 {
      font-size: 36px;
      margin: 0;
    }

    h2 {
      font-size: 24px;
```

```

        margin: 0;
    }

    .container {
        max-width: 800px;
        margin: 20px auto;
        padding: 20px;
        background-color: #fff;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }

    p {
        font-size: 16px;
        line-height: 1.5;
    }
</style>
</head>
<body>
    <header>
        <h1>Your Name</h1>
        <h2>Web Developer</h2>
    </header>

    <div class="container">
        <h2>About Me</h2>
        <p>
            Hello, I'm Your Name, a passionate web developer with a strong
            interest in creating responsive and user-friendly websites. I have experience in
            HTML, CSS, and JavaScript, and I'm always eager to learn new technologies.
        </p>

        <h2>Skills</h2>
        <ul>
            <li>HTML</li>
            <li>CSS</li>
            <li>JavaScript</li>
            <li>Responsive Web Design</li>
        </ul>

        <h2>Contact</h2>
        <p>Email: yourname@example.com</p>
        <p>LinkedIn: <a href="https://www.linkedin.com/in/yourname"
target="_blank">linkedin.com/in/yourname</a></p>
        <p>GitHub: <a href="https://github.com/yourname"
target="_blank">github.com/yourname</a></p>

```

```
    </div>  
</body>  
</html>
```