

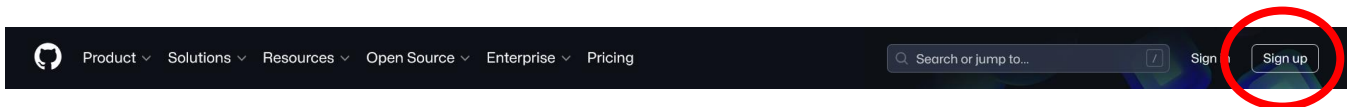
GitHub Setup Sheet

This worksheet will help you to set up your very first GitHub account, clone your first repository, and deploy your template website!

Please raise your hand if you're struggling 😊

Making an Account

1. Go to github.com (or search for GitHub on Google)
2. Click on “Sign Up” on the top.



Follow the instructions to make your account. Remember to make your username memorable! e.g. the first half of your school email

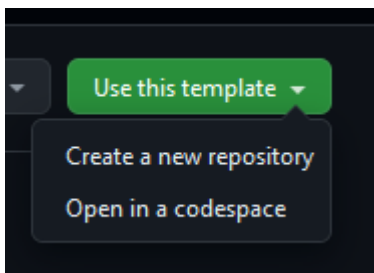
Once you've confirmed your account via email, follow any further instructions that appear on the GitHub website.

Creating a Template

1. Navigate to github.com/trinity-academy/website-template



2. Click on “Use this template”, then “Create a new repository”




3. Fill in the details as shown, but the repository name and description are up to you!
(And of course the owner will be your account as well).

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 ().*


Repository template

 trinity-coding-club/website-template ▾

Start your repository with a template repository's contents.

☐ **Include all branches**
Copy all branches from trinity-coding-club/website-template and not just the default branch.

Owner *

 wokaine ▾


Repository name *


trinity-web-wokaine


✔ trinity-web-wokaine is available.

Great repository names are short and memorable. Need inspiration? How about **fantastic-couscous** ?

Description (optional)

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

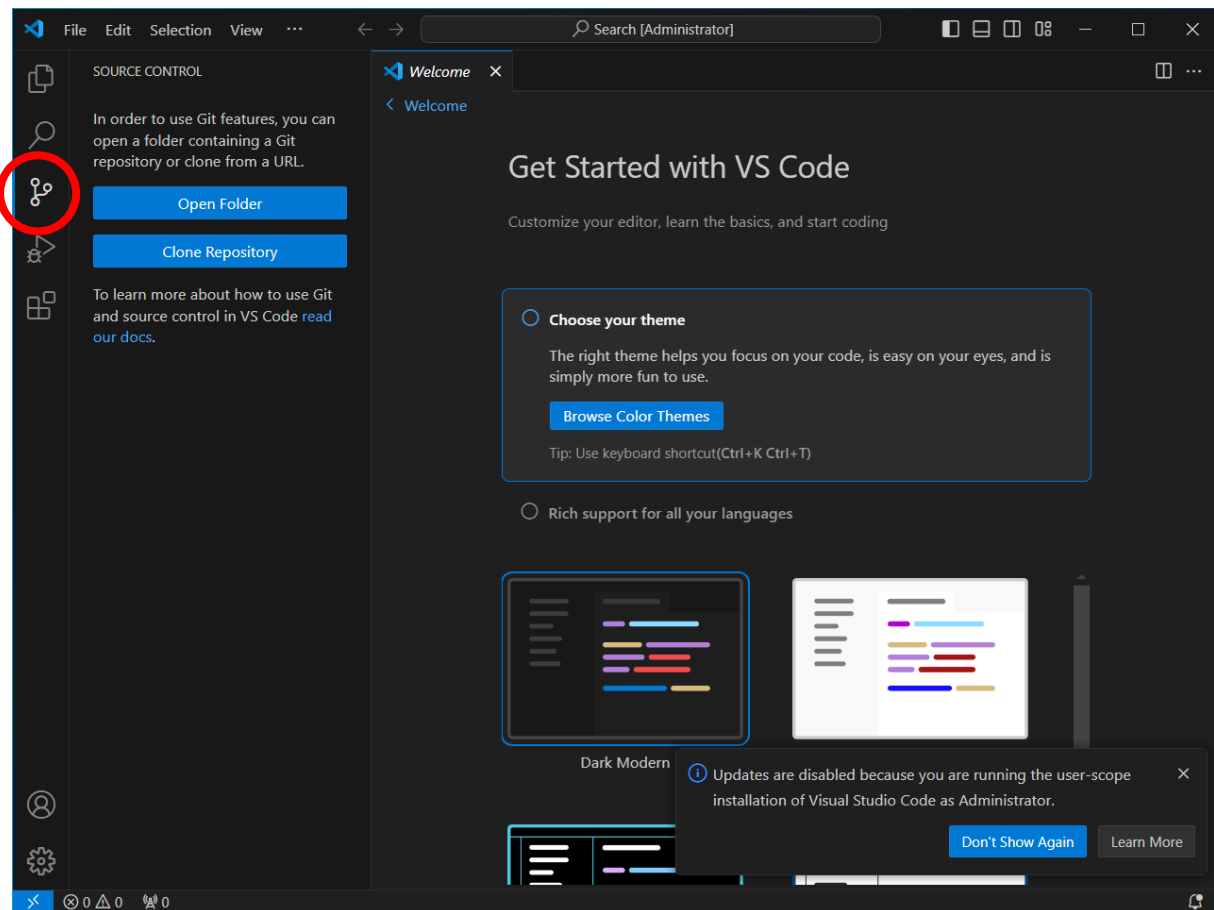
☐  **Private**
You choose who can see and commit to this repository.

 You are creating a public repository in your personal account.

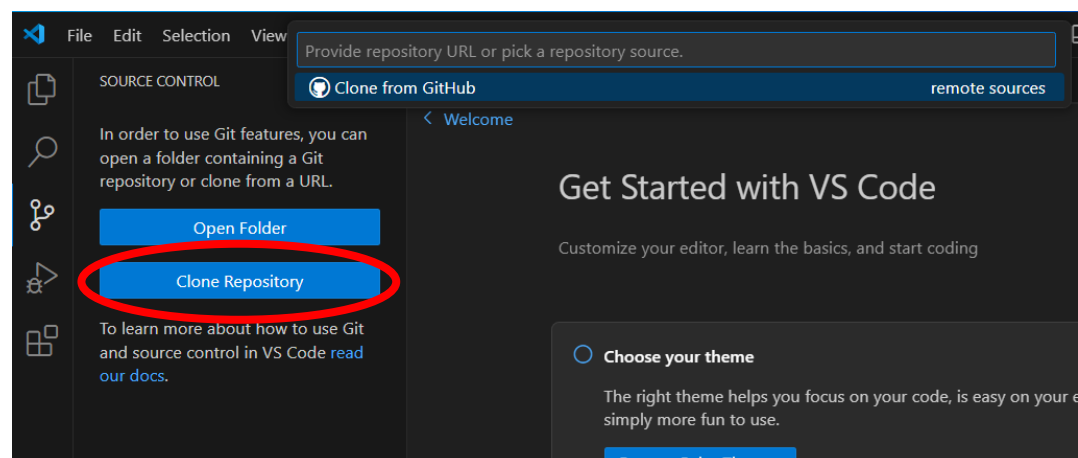
Create repository

Cloning the Repository

1. Open up **VSCode** on your computer.
2. Click on the icon pointed out below, and you should see the “Source Control” tab on the left like this:



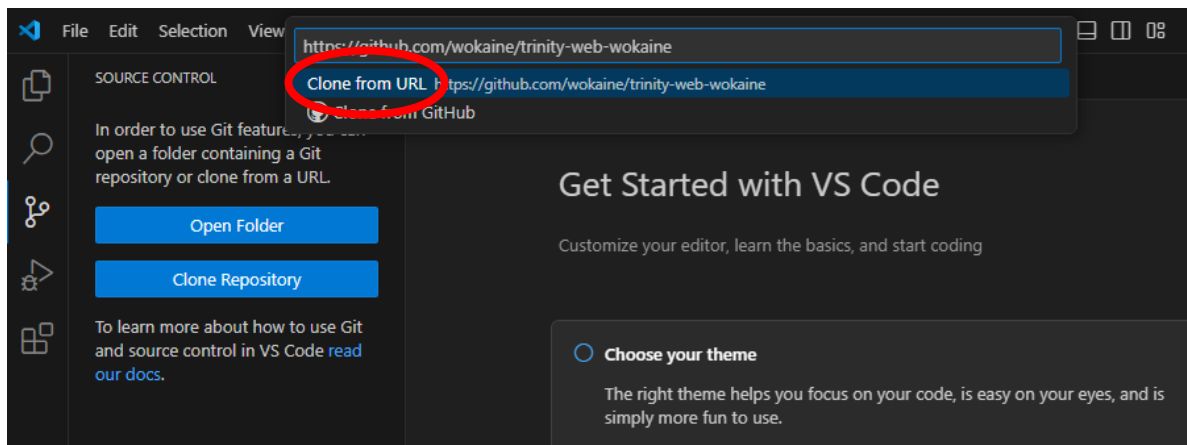
3. Click on “Clone Repository”.



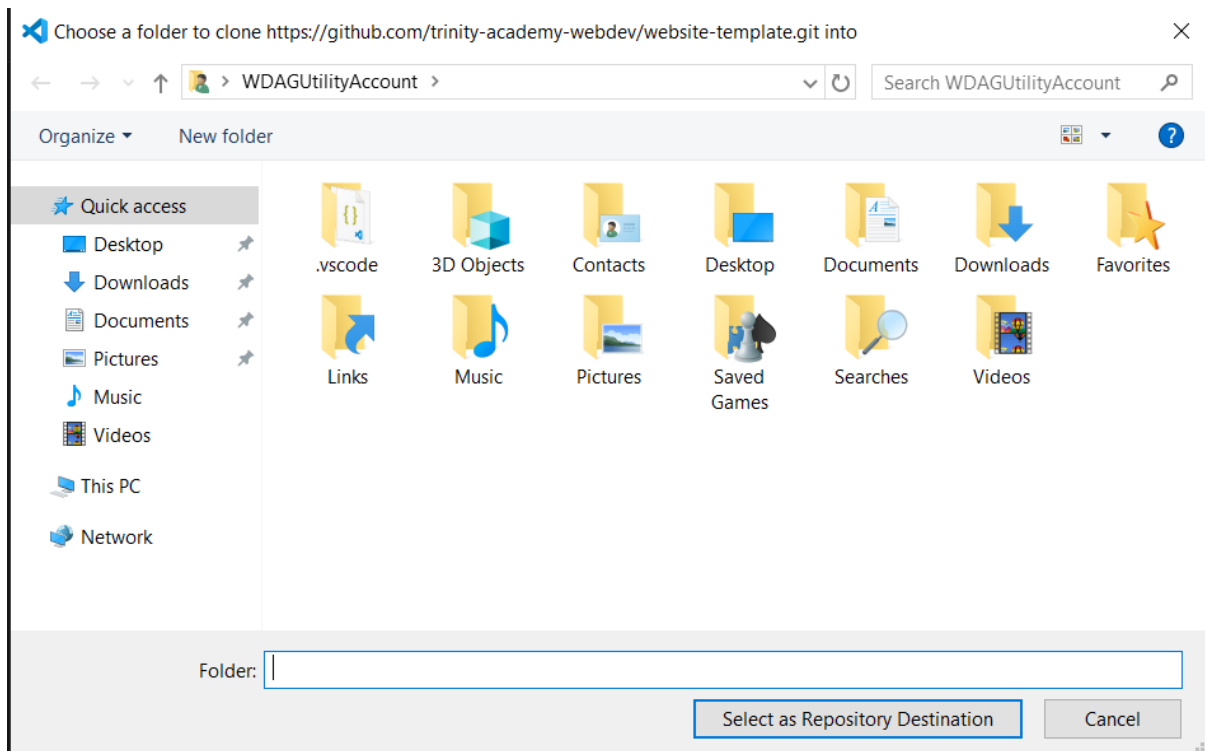
4. Where it says “Provide repository URL or pick a repository source”, put this link in:
https://github.com/<your_github_username_here>/<your_repo_name_here>.git

Be sure to replace [<your_github_username_here>](#) with your GitHub username, and [<your_repo_name_here>](#) with the name you gave to your repository!

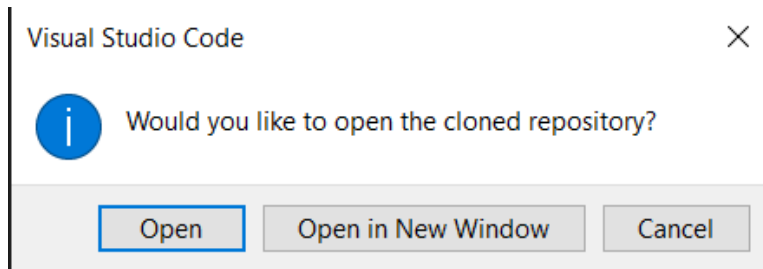
5. Click on “Clone from URL”



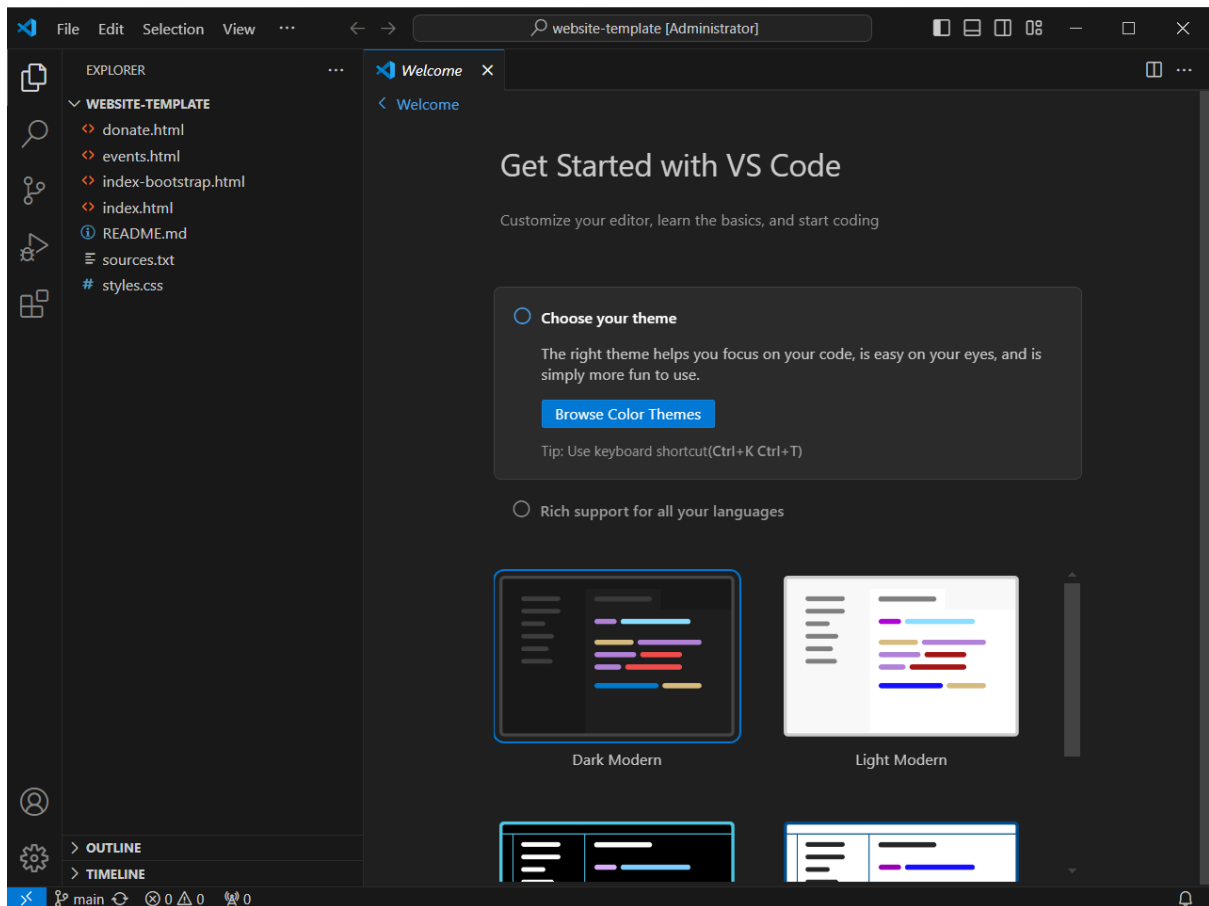
6. Select a suitable folder to put your cloned repository in. Make sure it is on your personal school drive though!



7. Press “Open”



And now you have the whole project in your VSCode window!

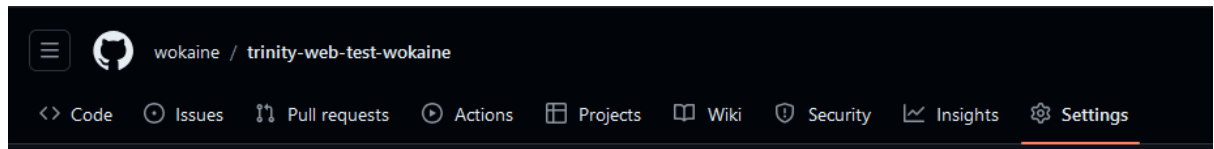


Deploying the Website to GitHub Pages

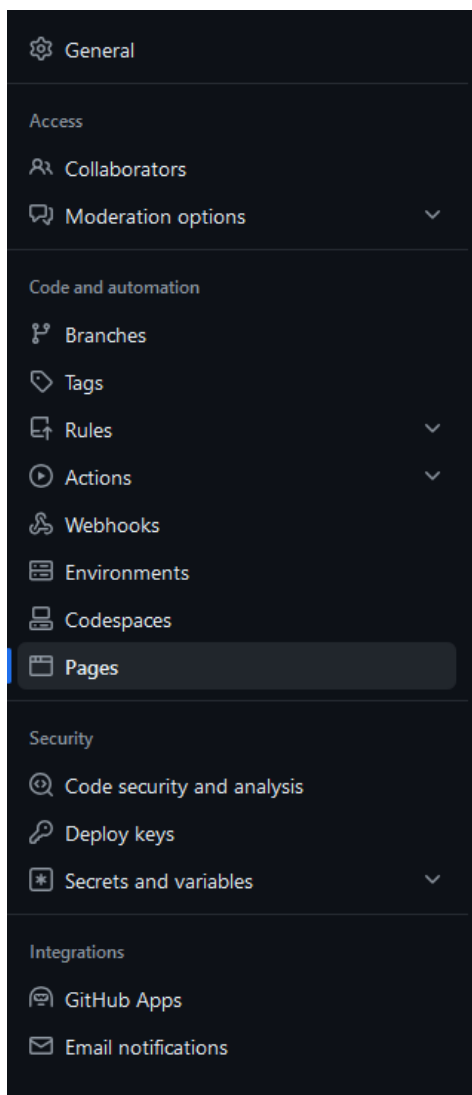
1. Navigate to your repository site

(https://github.com/<your_github_username_here>/<your_repo_name_here>.git)

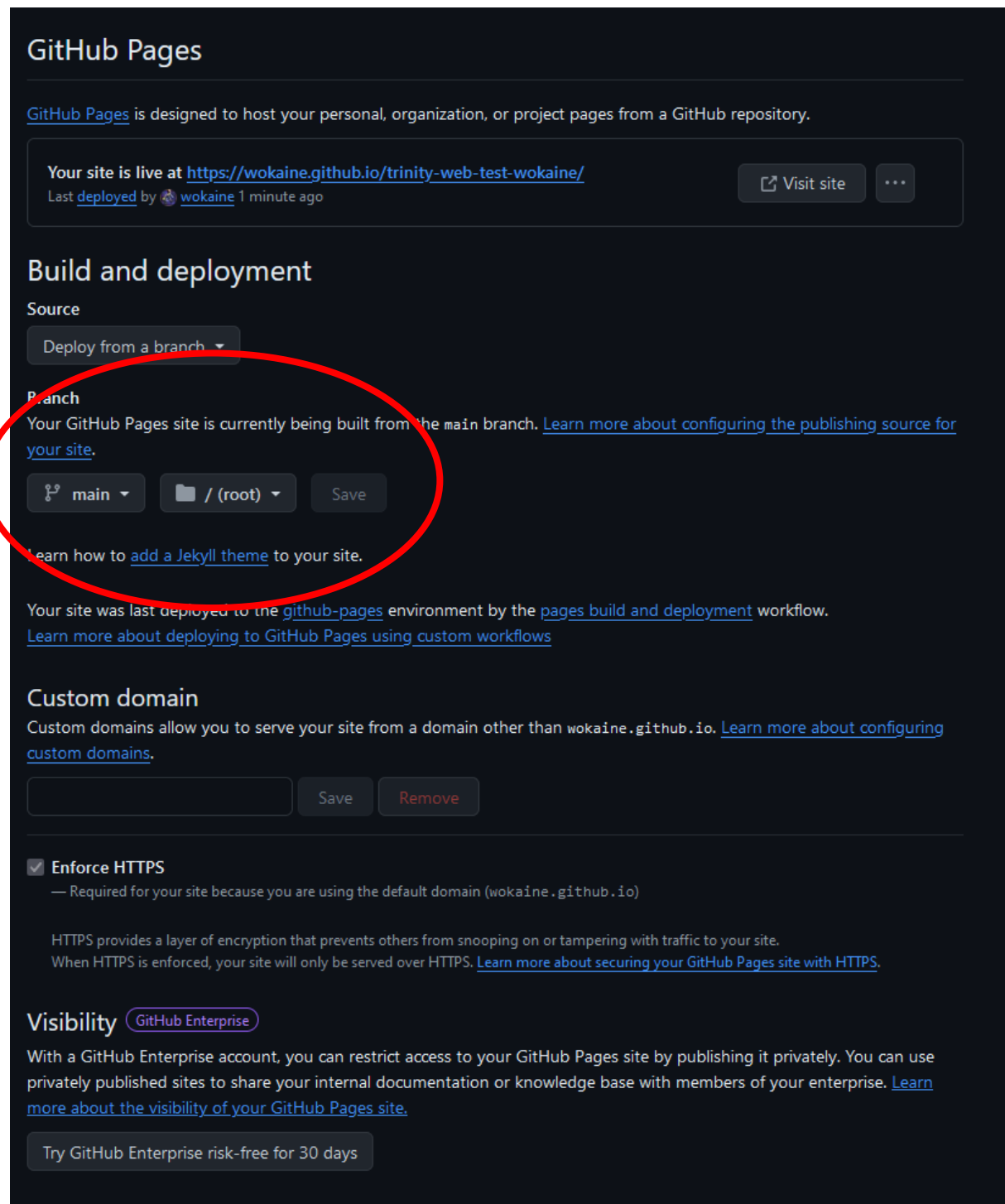
2. Click on “Settings”



3. On the left, click on “Pages”



- Under the “Branch” heading, click on the box saying “None” and change the branch to “main”, as shown below, then click “Save”.
It should take around 2 minutes to deploy.



GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Your site is live at <https://wokaine.github.io/trinity-web-test-wokaine/>
Last [deployed](#) by wokaine 1 minute ago

[Visit site](#) [...](#)

Build and deployment

Source

Deploy from a branch ▼

Branch

Your GitHub Pages site is currently being built from the main branch. [Learn more about configuring the publishing source for your site.](#)

main ▼ / (root) ▼ [Save](#)

[Learn how to add a Jekyll theme](#) to your site.

Your site was last deployed to the [github-pages](#) environment by the [pages build and deployment](#) workflow.
[Learn more about deploying to GitHub Pages using custom workflows](#)

Custom domain

Custom domains allow you to serve your site from a domain other than `wokaine.github.io`. [Learn more about configuring custom domains.](#)

[Save](#) [Remove](#)

Enforce HTTPS

☒ **Enforce HTTPS**
— Required for your site because you are using the default domain (`wokaine.github.io`)

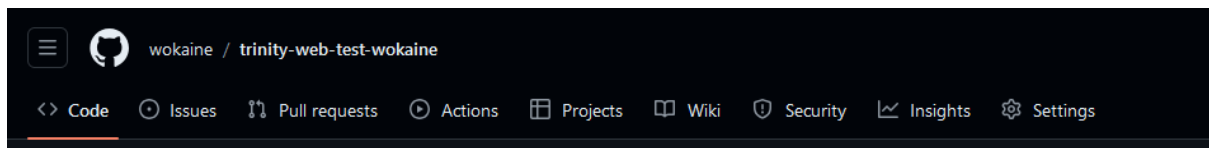
HTTPS provides a layer of encryption that prevents others from snooping on or tampering with traffic to your site.
When HTTPS is enforced, your site will only be served over HTTPS. [Learn more about securing your GitHub Pages site with HTTPS.](#)

Visibility GitHub Enterprise

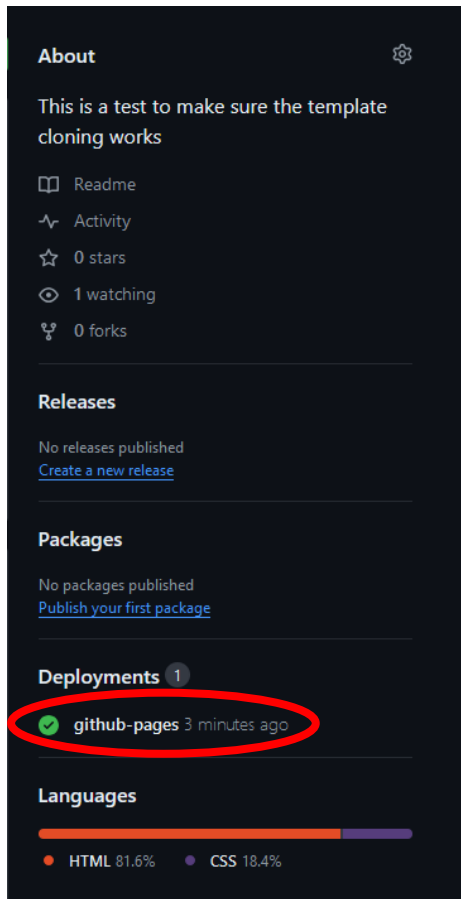
With a GitHub Enterprise account, you can restrict access to your GitHub Pages site by publishing it privately. You can use privately published sites to share your internal documentation or knowledge base with members of your enterprise. [Learn more about the visibility of your GitHub Pages site.](#)

[Try GitHub Enterprise risk-free for 30 days](#)

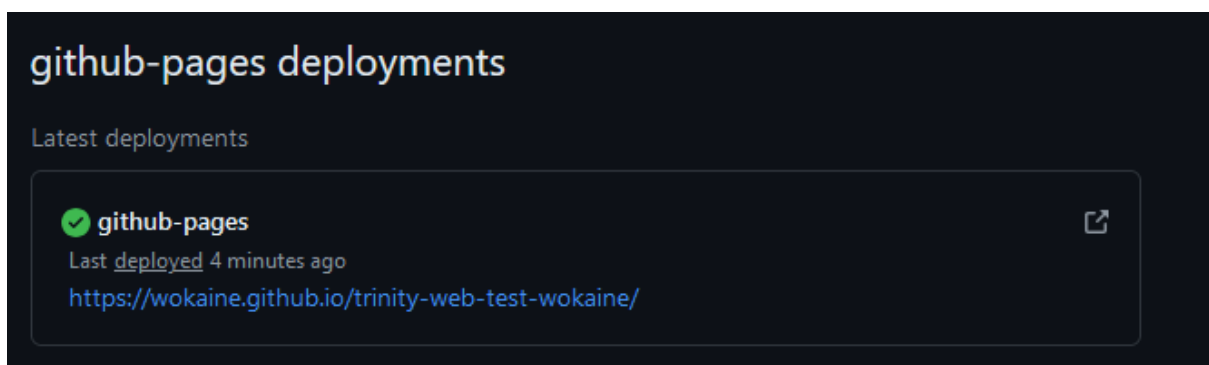
5. Click on “Code”.



6. On the right, click on the “github-pages” deployment.



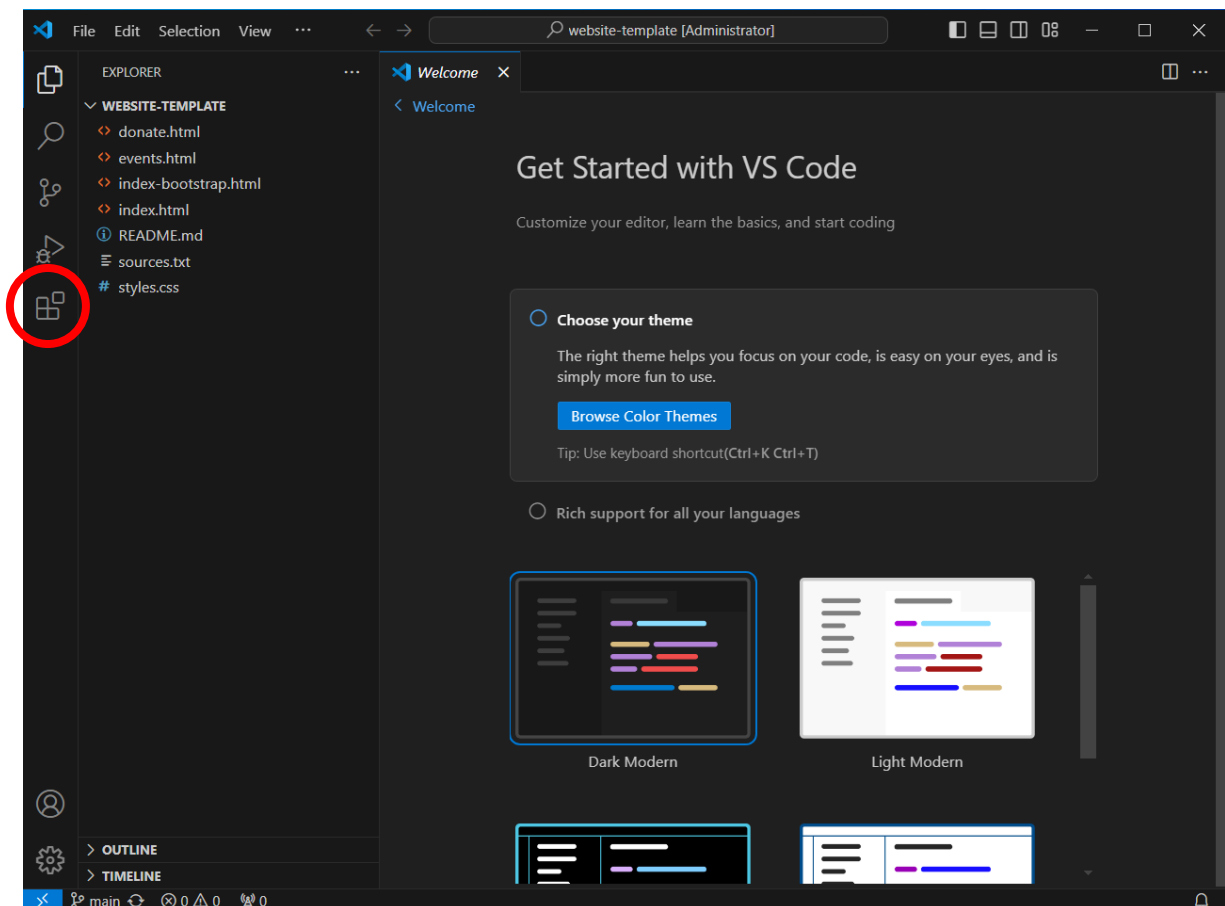
7. Click on the https link to visit your website!



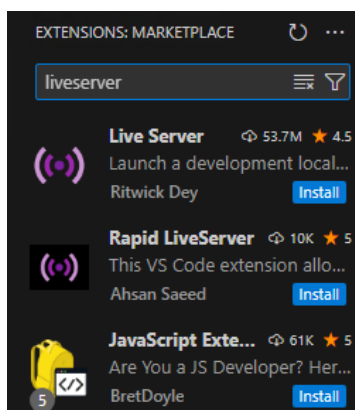
Installing the “Live Server” Extension

To make development a little bit easier, we will be downloading an **extension** called “LiveServer”. This will allow us to see our changes to our website automatically.

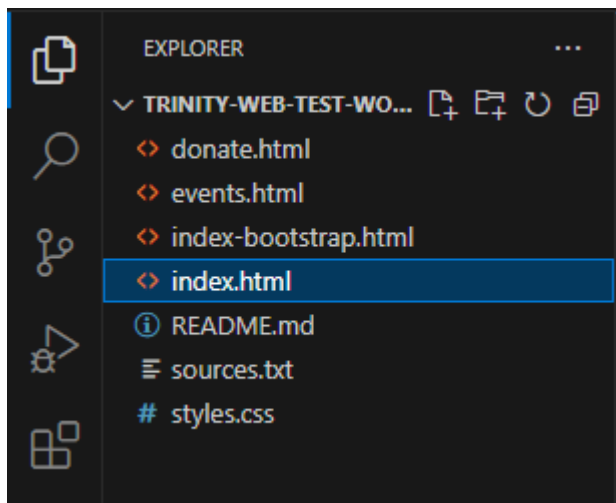
1. Go to VSCode and click on the button on the left with 4 blocks.



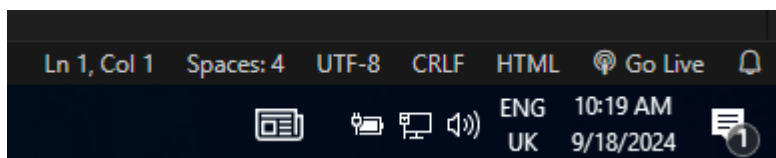
2. Search for “Live Server” and install the top one.



3. Then, open your homepage file (index.html).



Look to the bottom right of your screen, LiveServer has added the “Go Live” button to our VSCode!



Press “Go Live” and your website should be live! Any changes you make to your HTML or CSS should appear instantly!

What’s the difference between Live Server and deploying the website with GitHub Pages?

- By deploying the website with GitHub pages, you’re making the website available for anyone to access on the internet.
- Live Server allows you to preview your website on a browser on your own computer. It automatically updates the preview as you code.
- You need to commit and push your changes for the deployed website to be updated.