Lab #1:
INTRODUCTION TO GIT AND GITHUB

Topic	Introduction to Git and GitHub for Web Development
Domain of Learning	1. Cognitive (C3: Applying) – Students will apply version control concepts to track changes in web development. 2. Psychomotor (P3: Guided Response) – Students will follow step-by-step procedures to configure and use Git and GitHub. 3. Affective (A2: Valuing) – Students will recognize the importance of version control in web development.
Learning objective	1. Understand the role of Git and GitHub in version control for web development. (C2: Understanding) 2. Configure Git and GitHub by setting up a repository and managing commits. (P3: Guided Response) 3. Demonstrate Git operations including committing, pushing, pulling, and managing changes. (P4: Mechanism) 4. Apply GitHub Pages to host a static website. (C3: Applying)
Lab activity objective	To set up and use Git and GitHub for version control in web development by creating, committing, and managing files in a repository, and deploying a simple web page using GitHub Pages.

Introduction to Git and GitHub

Version control is an essential skill for web developers, allowing you to track changes, collaborate efficiently, and manage project versions systematically. **Git** is a distributed version control system that enables you to track code changes, revert to previous versions, and work on different features simultaneously. It is widely used in software development due to its efficiency and flexibility.

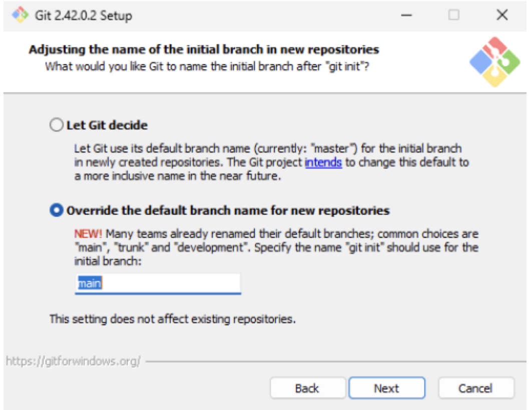
GitHub is a cloud-based platform that provides hosting for Git repositories, enabling collaboration among multiple developers. It allows you to store, manage, and share code while providing features such as issue tracking, pull requests, and GitHub Pages for hosting static websites.

In this lab, you will learn how to use Git and GitHub by setting up a repository, committing changes, pushing to GitHub, and deploying a simple web page using GitHub Pages.

Instruction: Follow the steps given.

1. **Registration of GitHub account:**
   1.1 Register GitHub account at https://github.com/

2. **Git installation:**
   2.1 Download Git at https://github.com/git-for-windows/git/releases/download/v2.51.0.windows.2/Git-2.51.0.2-64-bit.exe
   2.2 For MacOS users, follow the instructions at https://git-scm.com/download/mac
   2.3 **Leave the installation settings as default EXCEPT for the following**:



Git 2.42.0.2 Setup

Adjusting the name of the initial branch in new repositories

What would you like Git to name the initial branch after "git init"?

○ **Let Git decide**

Let Git use its default branch name (currently: "master") for the initial branch in newly created repositories. The Git project intends to change this default to a more inclusive name in the near future.

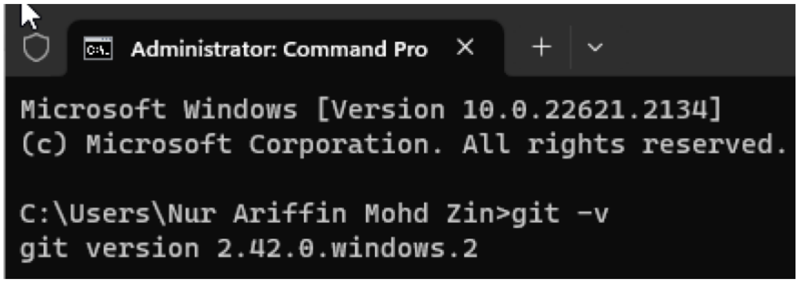
● **Override the default branch name for new repositories**

NEW! Many teams already renamed their default branches; common choices are "main", "trunk" and "development". Specify the name "git init" should use for the initial branch:

main

This setting does not affect existing repositories.

2.4 To verify that Git has been successfully installed, open **Command Prompt** and type `git -v`. The output should be as follows:



```
Microsoft Windows [Version 10.0.22621.2134]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Nur Ariffin Mohd Zin>git -v
git version 2.42.0.windows.2
```

2.5 Next, type the following command in Command Prompt. Replace `"Your GitHub Username"` with your actual name.

```
git config --global user.name "Your GitHub Username"
```

EXAMPLE: `git config --global user.name "ariffinmzin"`

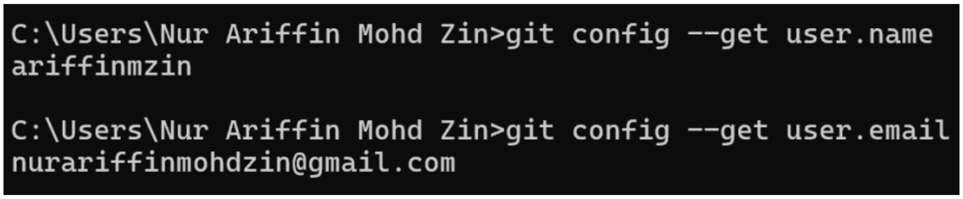
2.6 Then, type the following command to set the email address attached to your commits. Replace `"youremail@example.com"` with your actual email address. **THE EMAIL ADDRESS MUST BE THE SAME AS THE EMAIL ADDRESS YOU REGISTERED WITH GITHUB IN STEP 1.1**

```
git config --global user.email "youremail@example.com"
```

EXAMPLE: `git config --global user.email "nurariffinmohdzin@gmail.com"`

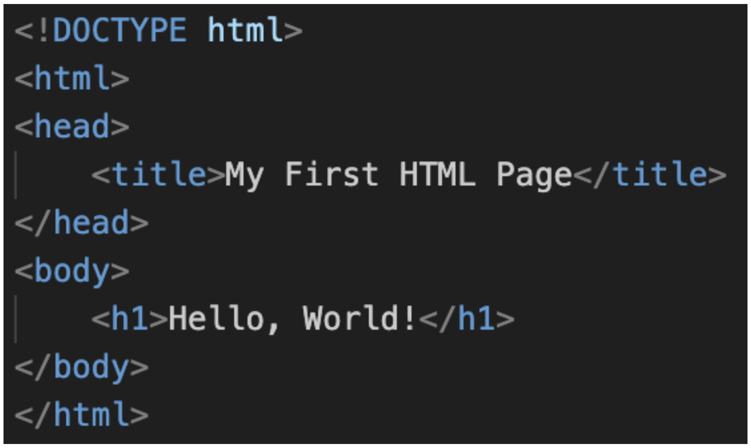
2.7 You can verify the `user.name` and `user.email` that has been set up by running the following command.

```
git config --get user.name

git config --get user.email
```



```
C:\Users\Nur Ariffin Mohd Zin>git config --get user.name
ariffinmzin

C:\Users\Nur Ariffin Mohd Zin>git config --get user.email
nurariffinmohdzin@gmail.com
```

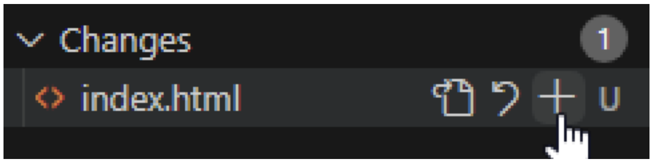
3. **Creating a New Folder and Repository**:
   3.1 Create a new folder where you want to store your project. Name it as **webdev_lab_1**.
   3.2 Open this folder in VS Code: **File** -> **Open Folder**.
   3.3 Once the folder is opened, open the Source Control pane .
   3.4 Click on the **Initialize Repository** button. This sets up a new Git repository in your folder.
4. **Creating an HTML file**:
   4.1 In VS Code, go to **File** -> **New File** or click the button.
   4.2 Save this file with the name **index.html** in your project folder.
   4.3 Write a basic HTML code as follows:



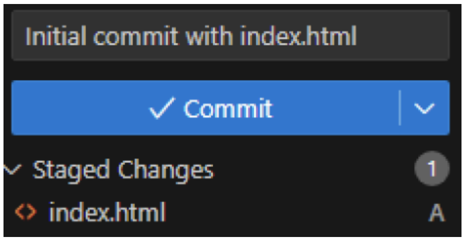
```
<!DOCTYPE html>
<html>
<head>
    <title>My First HTML Page</title>
</head>
<body>
    <h1>Hello, World!</h1>
</body>
</html>
```

   4.4 Make sure to save the file (Ctrl + S).

5. **Committing changes to Git**:
   5.1 After saving your HTML file, go back to the Source Control pane .
   5.2 You should see your **index.html** listed under **CHANGES**.
   5.3 Click the + sign next to the file name to stage the changes.



   5.4 Enter a commit message in the text box at the top (e.g., "Initial commit with index.html").



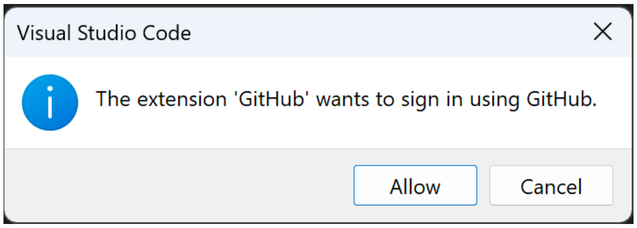
Initial commit with index.html

5.5 Lastly, press the checkmark icon ✓ Commit to commit your changes.

6. **Pushing to GitHub**:

6.1 Click the Publish Branch button.

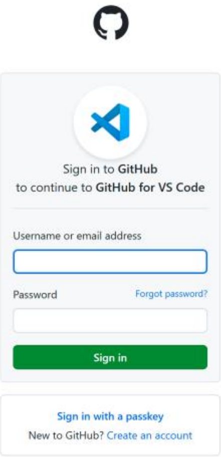
6.2 Select Allow if the following alert pops out.



Visual Studio Code

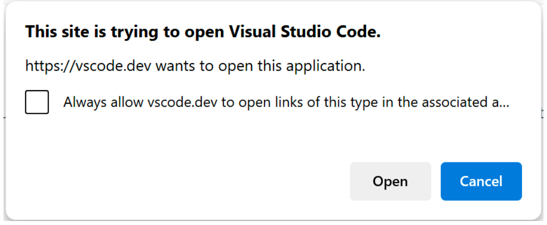
The extension 'GitHub' wants to sign in using GitHub.

6.3 Then, it will navigate to the GitHub login page. Enter your credentials.



Sign in to GitHub
to continue to GitHub for VS Code

6.4 Choose Open if the following alert appears.

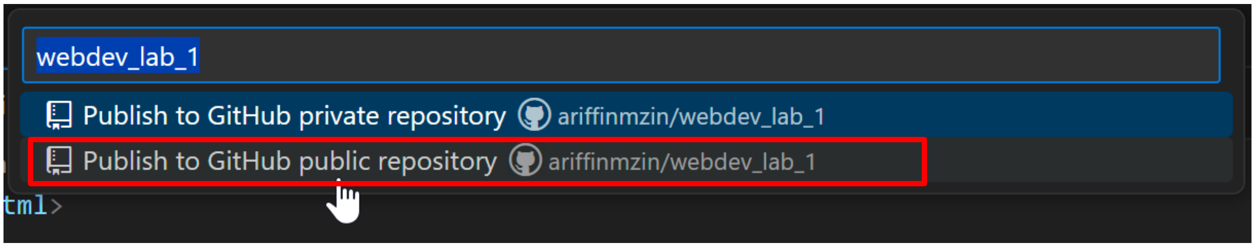


This site is trying to open Visual Studio Code.

https://vscode.dev wants to open this application.

□ Always allow vscode.dev to open links of this type in the associated a...

6.5 Make sure you choose Publish to GitHub **public** repository.

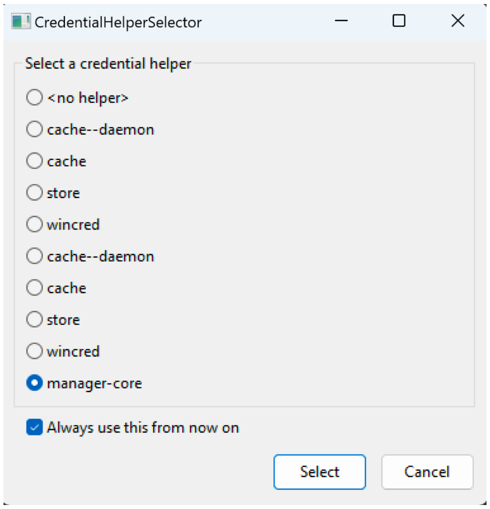


webdev_lab_1

Publish to GitHub private repository ariffinmzin/webdev_lab_1

Publish to GitHub public repository ariffinmzin/webdev_lab_1

6.6 If the following window pops out, select the following settings then press Select.



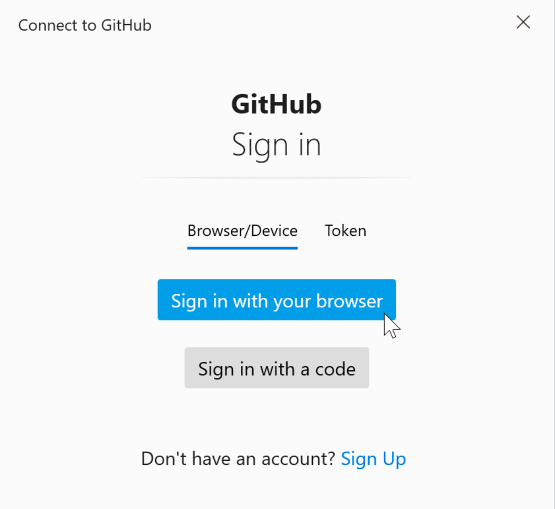
CredentialHelperSelector

Select a credential helper

○ <no helper>
○ cache--daemon
○ cache
○ store
○ wincred
○ cache--daemon
○ cache
○ store
○ wincred
● manager-core

☑ Always use this from now on

6.7 Choose Sign in with your browser.

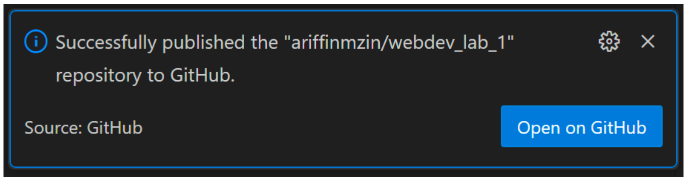


Connect to GitHub

GitHub
Sign in

Don't have an account? Sign Up

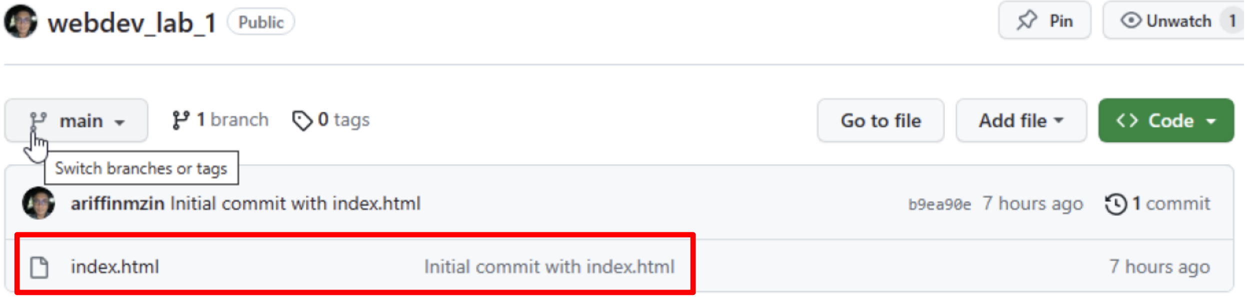
6.8 You will get a message indicating that your push was successful. Click Open on GitHub to view the file.



Successfully published the "ariffinmzin/webdev_lab_1" repository to GitHub.

Source: GitHub

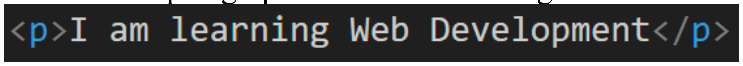
6.9 You should now see your **index.html** file in the repository.



ariffinmzin Initial commit with index.html b9ea90e 7 hours ago 1 commit

index.html Initial commit with index.html 7 hours ago

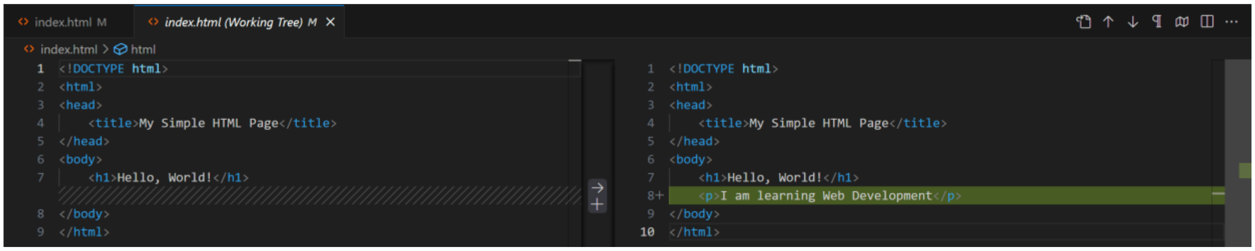
7. **Making Changes to the HTML File**:
   7.1 In VS Code, open your **index.html** file.
   7.2 Add a new paragraph below the **<h1>** tag and **save** the changes:



```
<p>I am learning Web Development</p>
```

8. **Viewing the Differences**:

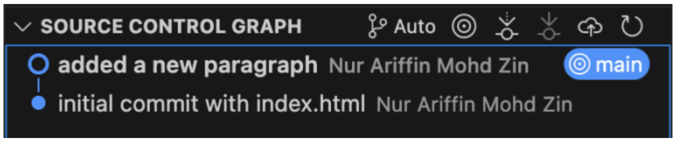
   8.1 Go to the Source Control pane in VS Code.
   8.2 Under **CHANGES**, you'll see **index.html**. Click on it.
   8.3 A new tab will open, showing the differences between the previous and current (after saved) of the same file. **Added lines will be highlighted in** green and removed (if any) in red.



```
<!DOCTYPE html>
<html>
<head>
    <title>My Simple HTML Page</title>
</head>
<body>
    <h1>Hello, World!</h1>
</body>
</html>
```

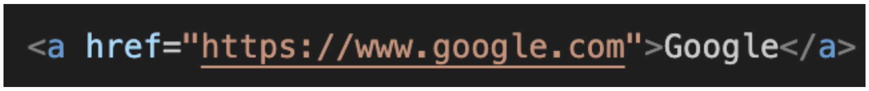
```
<!DOCTYPE html>
<html>
<head>
    <title>My Simple HTML Page</title>
</head>
<body>
    <h1>Hello, World!</h1>
+   <p>I am learning Web Development</p>
</body>
</html>
```

9. **Committing the New Changes**:
   9.1 Stage the changes by clicking the + sign next to the **index.html** under **CHANGES**.
   9.2 Enter a commit message ("Added a new paragraph").
   9.3 Press the checkmark icon ✓ Commit to commit your changes.
   9.4 You can also see the visualization of every commit in the Source Control Graph.



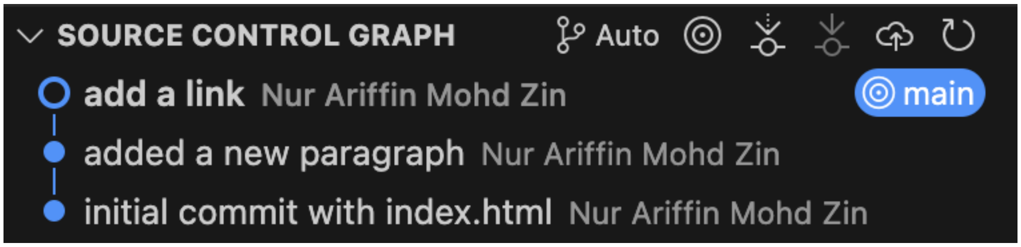
added a new paragraph Nur Ariffin Mohd Zin
initial commit with index.html Nur Ariffin Mohd Zin

10. **Undo Changes**:
   10.1 Open your **index.html** file.
   10.2 Add a new paragraph below the **<p>** tag and **save** the changes:



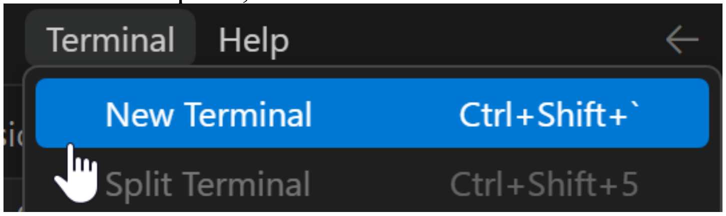
```
<a href="https://www.google.com">Google</a>
```

10.3 Go to the Source Control pane.
10.4 Stage the changes by clicking the + sign next to the **index.html**.
10.5 Enter a commit message ("Added a link").
10.6 Press the checkmark icon ✓ Commit to commit your changes.
10.7 You can see the visualization of the new commit in the Source Control Graph.

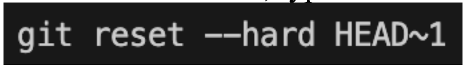


SOURCE CONTROL GRAPH Auto
○ add a link Nur Ariffin Mohd Zin main
● added a new paragraph Nur Ariffin Mohd Zin
● initial commit with index.html Nur Ariffin Mohd Zin

10.8 On the top bar, click Terminal – New Terminal

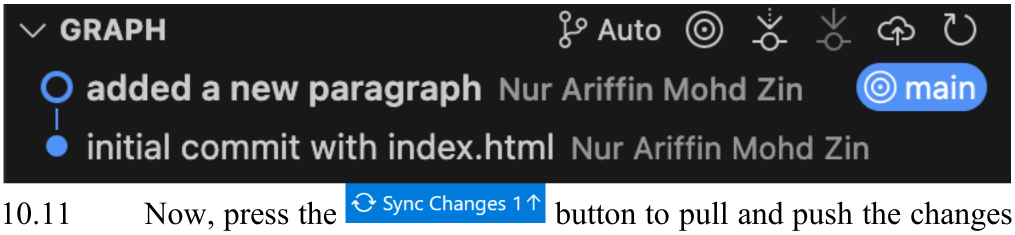


10.9 On the terminal, type the command below:



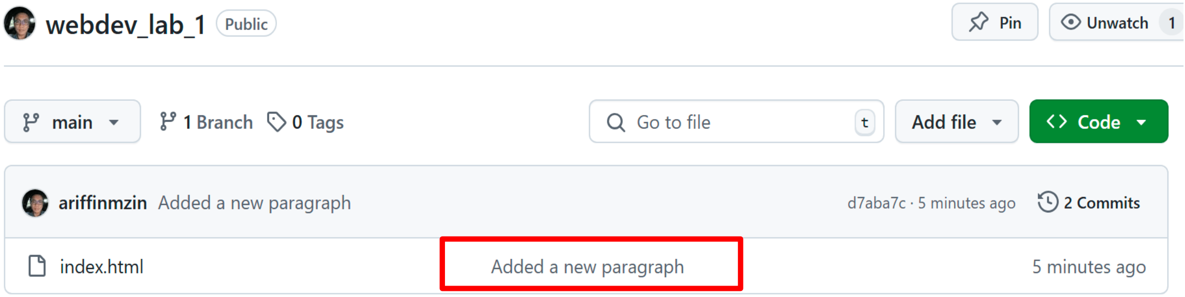
```
git reset --hard HEAD~1
```

10.10 You will see that the previous commit has been deleted.



GRAPH Auto
○ added a new paragraph Nur Ariffin Mohd Zin main
● initial commit with index.html Nur Ariffin Mohd Zin

10.11 Now, press the Sync Changes 1↑ button to pull and push the changes to the GitHub repository.
10.12 Verify the changes on GitHub.



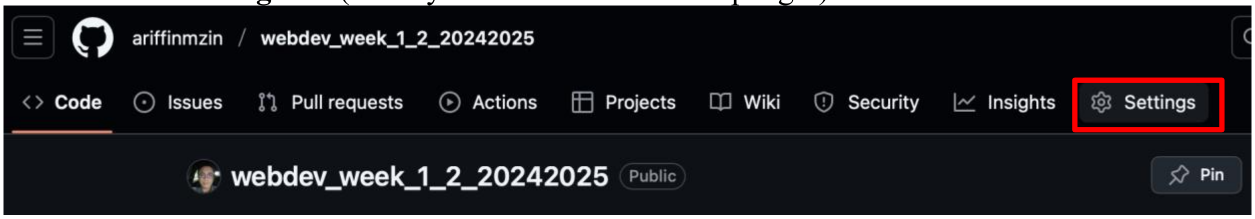
webdev_lab_1 Public Pin Unwatch 1

main 1 Branch 0 Tags Go to file t Add file Code

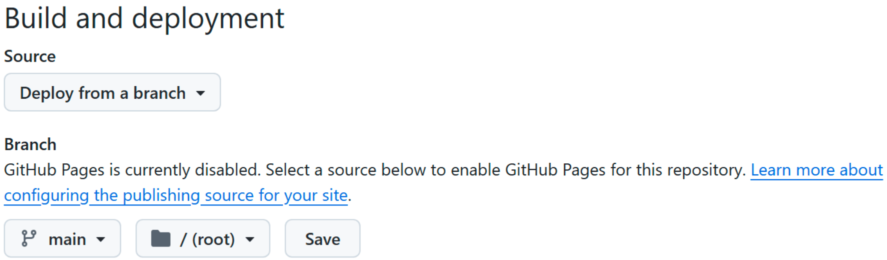
ariffinmzin Added a new paragraph d7aba7c · 5 minutes ago 2 Commits

index.html Added a new paragraph 5 minutes ago

11. **Enable GitHub Pages**:
    11.1 Navigate to your GitHub repository in your web browser.
    11.2 Click on the **Settings** tab (usually located towards the top-right).



    11.3 Find the **Pages** section (on the left navigation panel).
    11.4 Under **Source**, make sure to select **Deploy from a branch**.
    11.5 Under **Branch**, select the **main**. You might also have an option to choose the root or a **/docs** folder, but the **root** is fine for this lab.
    11.6 Click **Save**.



Build and deployment

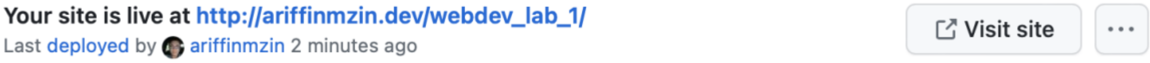
Source

Deploy from a branch

Branch

GitHub Pages is currently disabled. Select a source below to enable GitHub Pages for this repository. Learn more about configuring the publishing source for your site.

12. **Viewing Your Live Site**:
    12.1 After enabling GitHub Pages, refresh the **Pages** section until you see a link that says "Your site is live at [link]". Click on that link.



Your site is live at http://ariffinmzin.dev/webdev_lab_1/

Last deployed by ariffinmzin 2 minutes ago

    12.2 You'll now see your live **index.html** page! It might take a minute or two for your changes to appear the first time, so if your latest edits aren't showing up, give it a moment and refresh.
    12.3 You may try to visit the page from any device since the page is already up and running.

13. **AUTHOR submission:**
    13.1 Upload a **Microsoft Word file (.docx) or PDF (.pdf) to Author (INDIVIDUAL ACTIVITIES – LAB 1)** containing the following:
        a) The screenshot of your website,
        b) The screenshot of your GitHub repository, and
        c) The link generated in **Step 12.1**
    13.2 **The due date is BEFORE 12AM 07/10/2025**. Name your file as **YOURNAME_YOURMATRICNUMBER**.