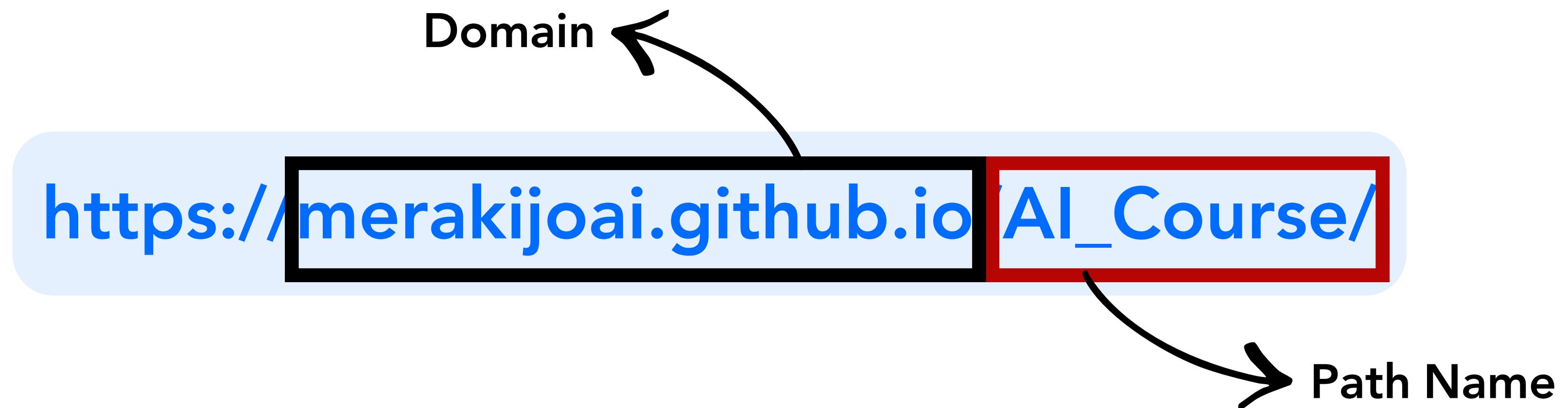


Deployment

What Does “Deployment” Mean?

- Right now, your page exists only on your laptop , no one can view it on the internet.
- Deployment gives it an address on the internet by **hosting** it on a **domain**.
- It becomes **published** so people can open it using a link



Hosting vs Domain

A website needs two things to exist online:

- Hosting
 - A place where your files live
 - Stores your HTML, CSS, images
 - Hosting can be free
 - Examples: GitHub Pages, Netlify
- Domain
 - The address people type to visit your site
 - Examples: `yourname.com`, `myproject.net`
 - Custom domain is always paid

You can have hosting without a custom domain. But you cannot have a website without hosting.

Hosting Tools

Hosting tool is a tool that allows you to publish your website online , it gives you customizable path name , but don't give you customizable domain

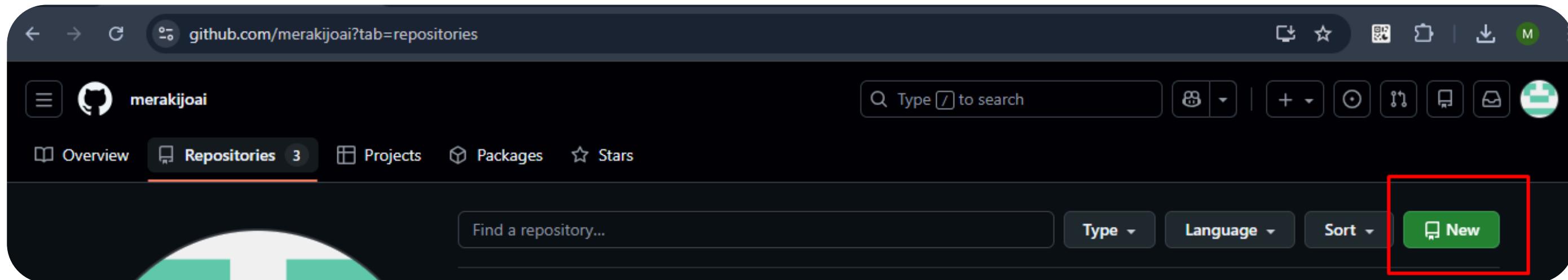
You don't need to understand how it works inside
You just need to know how to work with it

How To Deploy Your Web Page Using Github Pages

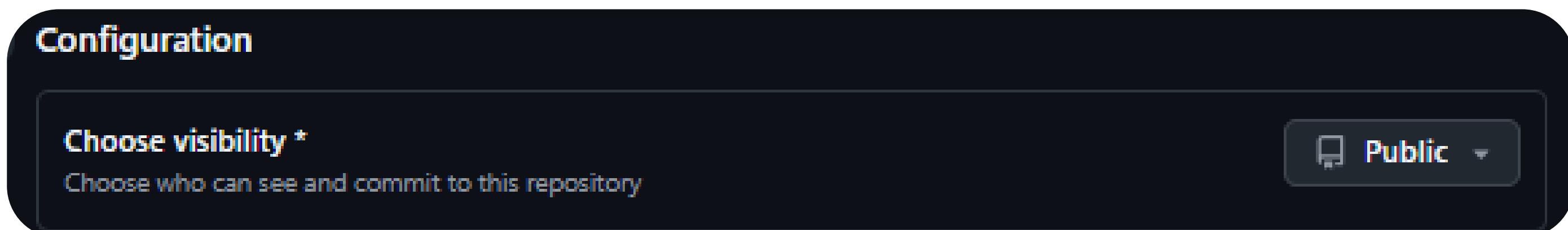
Step 1: Create a Repository

A repository is just a folder that lives on the internet and remembers versions.

1. Sign in to your github account
2. Click “New” from your repositories page



3. Name it with your site name
4. Make sure the visibility option is set to “public”



Step 1: Repository Naming



https://merakijoai.github.io/AI_Course/

The repository name is also important, because it is the path name the user see when visiting your site

For a general project, you can name the repository anything related to your project (e.g., my-project).

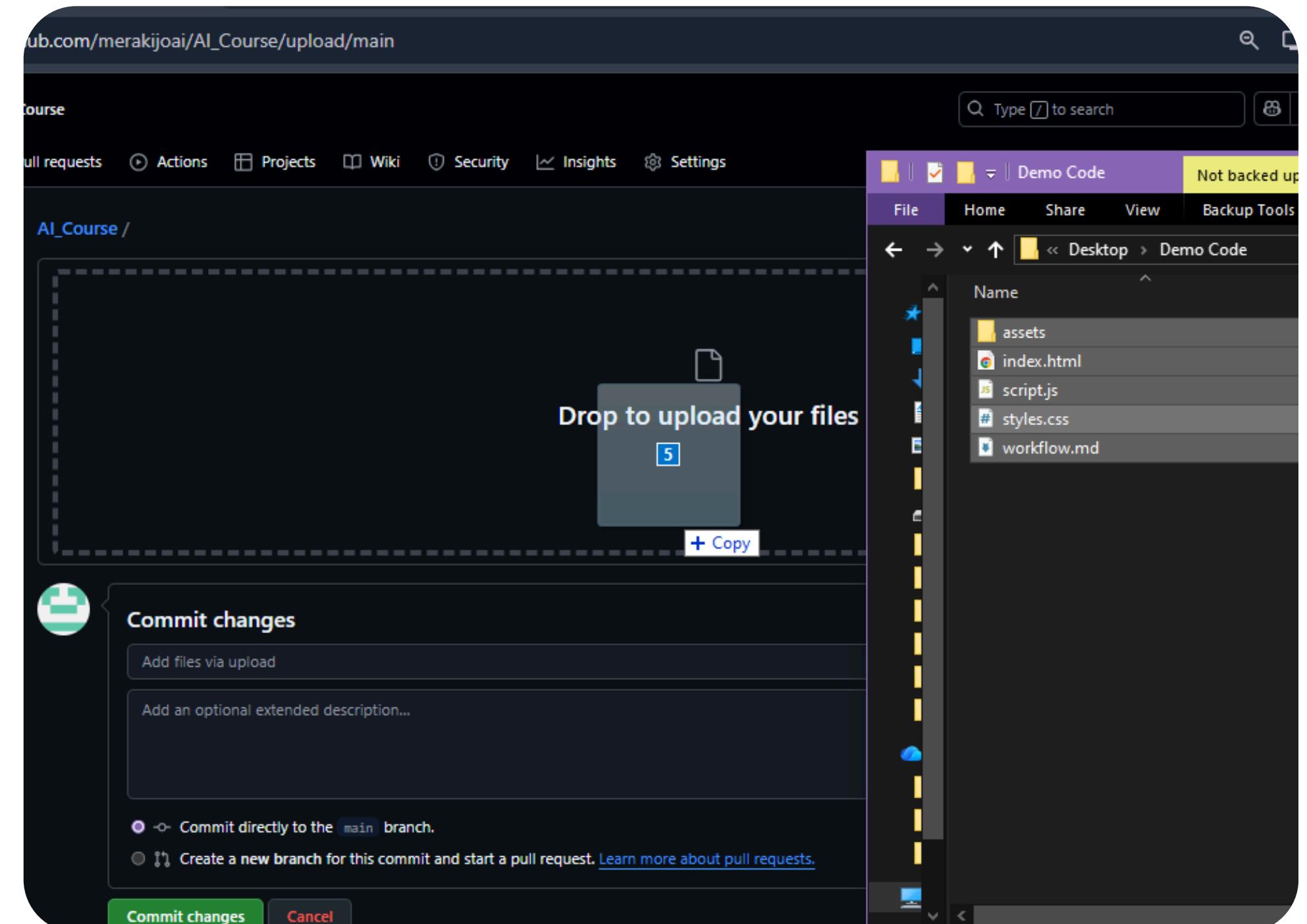
The site URL will be <user>.github.io/my-project/.



Step 2: Upload Your Files

Make sure you use the same name of the file provided below for HTML

1. Upload index.html
2. Upload CSS file
3. Files must be outside any folder
4. Upload any other files/folders related to your project
 - ex : readme file , assets folder ,etc..
5. Commit your changes



Step 3: Turn On Pages

1. Open repository settings
2. Find GitHub Pages on the left sidebar
3. Keep the default setting for Build and deployment
4. Deploy from a branch
5. Keep the default setting for branch: main branch and the root folder
6. After few minutes your published link will be ready!

Image Guide

Step 3: Turn On Pages (cont)

The screenshot shows the GitHub Pages settings interface for a repository. The top navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The Settings link is highlighted with a red box and the number 1. The left sidebar lists General, Access, Collaborators, and Moderation options under the General heading. Below these are sections for Code and automation, including Branches, Tags, Rules, Actions, Models, Webhooks, Copilot, Environments, and Codespaces. The Pages link in the sidebar is highlighted with a red box and the number 2. The main content area is titled "GitHub Pages" and contains a summary box with a red border and the number 5. It displays the message: "GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository." It also shows the live site URL: "Your site is live at https://merakijoai.github.io/AI_Workshop/", last deployed by merakijoai 3 weeks ago, with "Visit site" and "Unpublish site" buttons. The "Build and deployment" section, highlighted with a red box and the number 3, includes a "Source" dropdown set to "Deploy from a branch". The "Branch" section, highlighted with a red box and the number 4, shows the main branch is being used for deployment. A "Save" button is located in this section. At the bottom, there are links to add a Jekyll theme and learn about custom workflows.

1

2

3

4

5

Common Mistakes & How To Fix Them

1. File not named `index.html`

- rename the file on your device and upload it again to github

2. Files are inside a folder

- upload the files again in a new commit without them being in a folder

3. CSS not linked correctly to HTML

- Check the link between HTML and CSS in the HTML page header inside your code
 - The `<link>` tag has to exist
 - the CSS file name has to be exactly the name of the file in your code

Why Deployment Matters for Freelancing

1. Clients want links to view your work , they don't open zip folder or run your code locally
2. Links build trust because they are published on the internet
3. Live work looks professional in your portfolio
4. Your projects are the easiest way to show your skills

Deployment Checklist

(Preparation)

- My project is in one folder
- I have an index.html file (exact name)
- I have a styles.css file
- My CSS file is linked correctly
- My page opens correctly on my laptop
- I saved all the recent changes in my files

Deployment Checklist

Step 1: Repository

- I signed in to my GitHub account
- I created a new repository
- The repository name is my site name
- The repository visibility is public

Step 3: Turn On GitHub Pages

- I opened Settings
- I found GitHub Pages
- I selected the main branch
- I saved the settings

Step 2: Upload Files

- I uploaded my project files
- HTML and CSS files are NOT inside another folder
- I committed all the changes

Step 4: Test

- I opened the GitHub Pages link
- My website loads correctly , same content and style I want
- The page is responsive
- Now I can share the link

For any questions please ask on
Discord server technical questions channel

For urgent cases you can send WA on [+962787875915](tel:+962787875915)