**📄 TASK 6: Host a Static Website with GitHub Pages**

**🎯 Objective:**

Host a simple HTML website using **GitHub Pages**, making it live on the internet for free.

**🛠 Tools Required:**

* GitHub account
* Web browser
* Basic HTML file (like index.html)

**✅ Step-by-Step Documentation**

**🔹 Step 1: Create the HTML File (index.html)**

📝 This file is your website's main page.

**➤ How to do it:**

1. Open **Notepad** or **VS Code**.
2. Paste this code:

html

CopyEdit

<!DOCTYPE html>

<html>

<head>

<title>My Website</title>

</head>

<body>

<h1>Hello, world!</h1>

<p>This is my first website hosted on GitHub Pages.</p>

</body>

</html>

1. Save the file as: **index.html**

**💡 Why?**

GitHub Pages looks for a file named index.html to display as the home page.

**🔹 Step 2: Create a GitHub Repository**

**➤ How to do it:**

1. Login to <https://github.com>
2. Click the **"+"** sign at the top → Choose **"New repository"**
3. Enter a repository name like: My-Sample-Wesite
4. **Do NOT** select "Initialize with README" (optional)
5. Click **Create repository**

**💡 Why?**

The repository stores your website files online, like a cloud folder.

**🔹 Step 3: Upload Your Website Files**

**➤ How to do it:**

1. Go to your newly created repo
2. Click **"Add file" → "Upload files"**
3. Drag and drop your index.html file (and later style.css)
4. Scroll down and click **"Commit changes"**

**💡 Why?**

You need to upload your code to GitHub before GitHub Pages can host it.

**🔹 Step 4: Enable GitHub Pages**

**➤ How to do it:**

1. Go to the **repo settings** (click "Settings" on the top bar)
2. In the left sidebar, click **“Pages”**
3. Under **“Source”**, select:
   * Branch: main
   * Folder: / (root)
4. Click **"Save"**

**💡 Why?**

This tells GitHub to turn your repo into a live website.

**🔹 Step 5: Get Your Website Link**

**➤ How to do it:**

1. After saving, GitHub shows a green message:

✅ Your site is live at:  
https://your-username.github.io/your-repo-name/

1. Example for you:  
   ➜ https://raichal028.github.io/My-Sample-Wesite/

**💡 Why?**

This is the public URL where anyone can see your website.

**🔹 (Optional) Step 6: Add CSS to Make It Beautiful**

**➤ 1. Create style.css:**

css

CopyEdit

body {

background-color: #f2f2f2;

text-align: center;

font-family: Arial, sans-serif;

padding-top: 100px;

}

h1 {

color: #003366;

}

p {

color: #333;

}

**➤ 2. Update your index.html to link this CSS:**

html

CopyEdit

<head>

<title>My Website</title>

<link rel="stylesheet" href="style.css">

</head>

**➤ 3. Upload style.css to your GitHub repo.**

**✅ Final Output:**

* 🔗 Website link: https://raichal028.github.io/My-Sample-Wesite/
* 📦 GitHub repo: <https://github.com/raichal028/My-Sample-Wesite>

**🧠 Summary:**

| **Step** | **Task** | **Purpose** |
| --- | --- | --- |
| 1 | Create index.html | Home page of your website |
| 2 | Make a GitHub repo | Store your files online |
| 3 | Upload files | Send files to GitHub |
| 4 | Enable GitHub Pages | Make the site live |
| 5 | Get the URL | Share your site |
| 6 (Optional) | Add CSS | Make your site look nice |

**📖 1️⃣ What is GitHub Pages?**

**GitHub Pages** is a **free static site hosting service** provided by GitHub.  
It allows you to host:

* Static websites (HTML, CSS, JS)
* Personal portfolios
* Project documentation
* Blogs (Jekyll-powered)

You host the website directly from a GitHub repository, and GitHub serves the pages publicly at:

php-template

CopyEdit

https://<username>.github.io/<repository-name>/

**⚙️ 2️⃣ Can you host dynamic apps here?**

👉 **No**, GitHub Pages only supports **static content**:

* HTML
* CSS
* JavaScript  
  (No server-side processing like PHP, Python, Node.js)

⚠️ If you need dynamic features (like databases, user login, etc.), you’d have to:

* Use **client-side JavaScript** to call external APIs
* Or host the dynamic backend elsewhere (like AWS Lambda, Firebase)

**📏 3️⃣ What are the limits of GitHub Pages?**

✅ Key limitations:

* **Only static files**
* Max **repository size: 1 GB**
* Max **published site size: 1 GB**
* Bandwidth limit: **100 GB per month**
* Max **file size: 100 MB**
* No backend or databases
* Uses **Jekyll** by default (can be disabled)

**🔄 4️⃣ How do you update the website?**

You update your GitHub Pages site by:

* **Pushing new commits** to the designated branch (commonly main, master, or gh-pages)
* GitHub automatically rebuilds and deploys your site

**For example:**

bash

CopyEdit

git add .

git commit -m "Updated homepage"

git push origin main

The site will update automatically within a few seconds.

**❌ 5️⃣ What happens when you delete the repo?**

If you delete the GitHub repository:

* The **associated GitHub Pages site is instantly removed**
* The URL will return a **404 error**
* There’s no automatic backup — you’ll need to have a local copy to restore it

**📂 6️⃣ What is the default file that loads?**

The default landing file is:

diff

CopyEdit

index.html

When a user visits your site URL, GitHub Pages automatically looks for index.html in the root directory (or /docs if configured) and loads it.

**🌐 7️⃣ Can you use a custom domain?**

✅ Yes — you can assign your own domain to a GitHub Pages site.

Steps:

1. In your repo:  
   Create a CNAME file in your root directory with your domain name.
2. In your domain registrar’s DNS settings:  
   Set a **CNAME record** pointing to:

lua

CopyEdit

<username>.github.io

1. GitHub will serve your pages at your custom domain.