 

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

Deploy your static website using github page: Host your local git repository’s static website directly using github pages.

Name: S.KRITHIKA Department: ADS



**Introduction and Overview**

GitHub Pages is a free hosting service provided by GitHub that allows developers to publish static websites directly from a Git repository. This makes it an excellent choice for hosting personal portfolios, project documentation, and simple static sites without any backend functionality. By leveraging GitHub Pages, developers can deploy their local Git repository's website seamlessly and make it accessible to a global audience.

**Objective**

The primary objective of this guide is to demonstrate how to deploy a static website using GitHub Pages. By the end of this guide, users will understand how to set up a local Git repository, push it to GitHub, and enable GitHub Pages to host the website. This will help developers and non-developers alike to share their web projects easily.

**Importance of Local Hosting**

Hosting a static website locally before deploying it to GitHub is essential for several reasons:

* **Testing & Debugging**: Ensures that the website works as expected before making it publicly accessible.
* **Version Control**: Allows developers to track changes and roll back to previous versions if needed.
* **Faster Development Cycle**: Enables quick iterations and modifications without relying on an external server.
* **Security**: Reduces risks of exposing unfinished or untested code to the public.

**Step-by-Step Deployment Guide**

**1. Set Up Your Local Git Repository**

* Create a folder for your website on your local machine.
* Navigate to the folder and initialize a Git repository:

git init

* Add your HTML, CSS, and JavaScript files to this folder.

**2. Create a GitHub Repository**

* Log in to GitHub and create a new repository.
* Do not initialize it with a README, .gitignore, or license (to avoid conflicts).
* Copy the repository URL.

**3. Link Local Repository to GitHub**

* In your terminal, navigate to your local project directory and add the remote repository:

git remote add origin <your-repository-url>

* Add and commit your files:
* git add .

git commit -m "Initial commit"

* Push the changes to GitHub:

git push -u origin main

**4. Enable GitHub Pages**

* Navigate to your repository on GitHub.
* Go to **Settings** > **Pages**.
* Under **Branch**, select main (or master) as the source.
* Click **Save**.
* GitHub will generate a URL for your website (e.g., https://yourusername.github.io/repository-name/).

**5. Verify Your Website**

* Open the provided GitHub Pages URL in a browser.
* If necessary, make updates to your local repository, commit, and push changes to reflect updates on the live site.

**Expected Outcome**

By following these steps, you will have successfully deployed your static website using GitHub Pages. The website will be accessible via a unique GitHub Pages URL and will automatically update whenever you push changes to the repository. This method ensures easy maintenance, version control, and a cost-effective way to host personal and project websites.