

Placement Empowerment Program

Cloud Computing and DevOps Centre

Deploy Your Static Website Using GitHub Pages
Host your local Git repository's static website directly using
GitHub Pages.

Name: Hariharan v

Department: ADS

Introduction

GitHub Pages is a free and straightforward hosting service provided by GitHub for static websites. By leveraging this feature, you can host personal, project, or organization websites directly from a GitHub repository. This guide walks you through the process of deploying your static website using GitHub Pages.

Objective

The objectives of this task are:

- Understand the basics of GitHub Pages and how it hosts static websites.
 - Learn to configure a GitHub repository for website hosting.
 - Successfully deploy a static website using GitHub Pages.
-

Step 1: Prepare Your Static Website

1. Ensure that your static website files are ready and organized. The main entry file should typically be `index.html`.
 2. Test the website locally to verify that all resources (CSS, JavaScript, images, etc.) are loading correctly.
-

Step 2: Set Up Your GitHub Repository

1. Create a New Repository

- Navigate to [GitHub](https://github.com) and create a new repository.
- Name the repository appropriately (e.g., `my-website`).
- Set the repository to public if you want it accessible to everyone.

2. Clone the Repository

```
git clone https://github.com/your-username/my-website.git
```

3. Add Your Website Files

- Copy your static website files into the cloned repository folder.

4. Commit and Push Files

```
cd my-website
```

```
git add .  
git commit -m "Initial commit"  
git push origin main
```

Step 3: Enable GitHub Pages

1. Navigate to the Repository Settings

- Go to your repository on GitHub.
- Click on the **Settings** tab.

2. Enable GitHub Pages

- Scroll down to the **Pages** section in the sidebar.
- Under the "Branch" dropdown, select main (or the branch containing your website files).
- If your website files are in a folder (e.g., /docs), specify the folder as the source.
- Click **Save**.

3. Note the Published URL

- GitHub will provide a URL (e.g., <https://your-username.github.io/my-website>) where your website will be hosted.
-

Step 4: Verify the Deployment

1. Wait a few minutes for GitHub Pages to process your files.
 2. Open the published URL in a browser to view your website.
 3. If the website does not display correctly, check:
 - File paths for resources (ensure they are relative).
 - The correct branch and folder settings in GitHub Pages.
-

Step 5: Update and Maintain Your Website

1. Make Changes Locally

- Edit your website files as needed.

2. Commit and Push Updates

`git add .`

`git commit -m "Updated website content"`

`git push origin main`

3. Automatic Deployment

- GitHub Pages automatically updates the hosted website when changes are pushed to the configured branch.

Additional Options

- **Custom Domain:** If you own a domain, you can configure GitHub Pages to use it.
 - Add your custom domain in the Pages settings.
 - Configure DNS settings as per GitHub's instructions.
- **Themes:** Use Jekyll themes provided by GitHub Pages for simple customization.

Conclusion

Deploying a static website using GitHub Pages is a quick and efficient way to make your site publicly accessible. By following this guide, you can host your projects effortlessly and update them as needed. GitHub Pages simplifies website deployment, making it ideal for personal and project pages.