



**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

To host a new static website using github pages in a git repository.

Name: Subiksha R Department:AML



**INTRODUCTION:**

To host a static website using GitHub Pages as part of your Proof of Concept (PoC), you would start by creating a new GitHub repository, ensuring it's public to use GitHub Pages for free. Once the repository is set up, you can prepare your static website files, which typically consist of HTML, CSS, JavaScript, and any other assets like images or fonts. These files should be uploaded to the repository, either directly in the root directory or within a subfolder, depending on your preference for organizing the files. After pushing your website files to the GitHub repository, you can enable GitHub Pages by going to the repository settings and selecting the branch or folder where your static files are located. GitHub Pages will automatically generate a URL (e.g., username.github.io/repository-name), allowing you to access your website live on the web. This setup is an ideal solution for showcasing simple static websites, documentation, portfolios, or any other project that doesn't require server-side processing. GitHub Pages offers a quick, free, and effective way to deploy static websites directly from your Git repository**.**

**OVERVIEW: Top of Form**

Hosting a static website using GitHub Pages is a practical and efficient solution for deploying websites directly from a Git repository. GitHub Pages is a free hosting service offered by GitHub that allows users to publish static content such as HTML, CSS, JavaScript, and other static assets like images or fonts. For your Proof of Concept (PoC), the process begins by creating a public GitHub repository where you can upload the files for your static website. Once the files are in place, you enable GitHub Pages through the repository settings, choosing the branch or folder that contains the website files. GitHub will then automatically generate a URL, typically in the format username.github.io/repository-name, where your website can be accessed live on the web. This solution is especially useful for small-scale projects, portfolios, documentation, or project demos, as it requires minimal setup and no backend server. It’s a straightforward way to showcase content, with no need for expensive hosting or complex configurations, making it ideal for static websites.

**Bottom of Form**

**OBJECTIVE:**

1. Easy Deployment: To quickly deploy a static website without the need for a server or backend infrastructure.

2. Free Hosting: To utilize GitHub Pages as a free hosting service for public repositories, making it cost-effective for small projects.

3. Version Control Integration: To leverage GitHub's version control system to manage and update website files easily.

4. Automatic Hosting: To automatically generate a live URL (e.g., username.github.io/repository-name) that can be accessed by anyone on the web.

5. Minimal Configuration: To set up and configure the website with minimal effort using GitHub's built-in settings for Pages.

6. Showcase Projects: To demonstrate and showcase portfolios, documentation, or any other static content in a professional and accessible way.

7. Cross-Platform Accessibility: To ensure the website is accessible from any device with an internet connection.

8. Collaborative Features: To enable collaboration and contributions from others by sharing the repository and allowing others to submit changes or updates.

**Importance of Setting Up a Local Repository**

1. **Cost-Effective**: GitHub Pages offers free hosting for public repositories, making it an ideal solution for small projects, portfolios, and documentation without incurring any hosting costs.

2. **Easy Setup**: The setup process is quick and simple, requiring no additional server configurations or complex deployment steps, making it beginner-friendly.

3. **Version Control**: By using GitHub, your website is automatically integrated with Git’s version control system, allowing you to track changes, roll back updates, and collaborate with others efficiently.

4. **Automatic Deployment**: GitHub Pages automatically publishes the website whenever you push updates to the repository, making it easy to keep the site live and up-to-date with minimal manual intervention.

5. **Reliability and Uptime**: GitHub, a platform known for its reliability, ensures that your website will have strong uptime and performance without the need for additional hosting management.

6. **Easy Collaboration**: Since your website is stored in a GitHub repository, it becomes easy for multiple collaborators to contribute, submit changes, and provide feedback in an organized manner.

7. **SEO and Visibility**: A live, public website hosted on GitHub Pages enhances visibility, especially if you're showcasing a project or portfolio, and helps with search engine optimization (SEO

8.**Custom Domains**: GitHub Pages allows you to configure custom domain names, which helps personalize your website and make it more professional without needing to pay for external hosting.

Top of Form

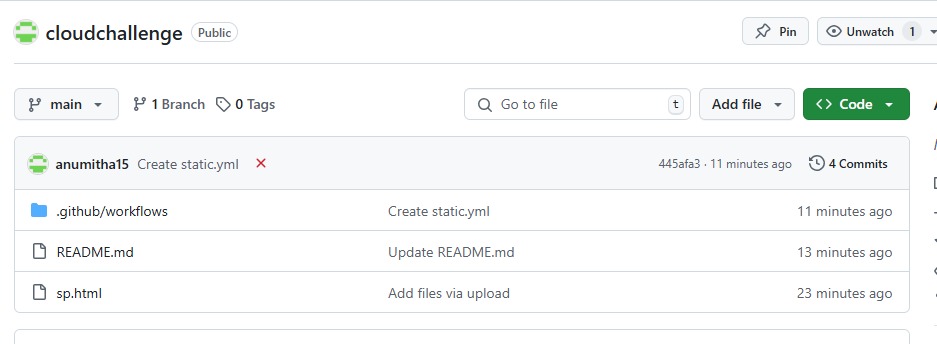
Bottom of Form

**STEP BY STEP OVERVIEW:**

**STEP 1:**

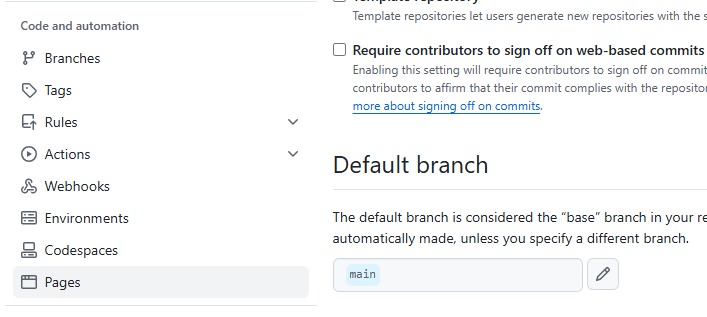
**Create a git repository in your github:**

Add a html file to it to host that using github



**Step 2**:

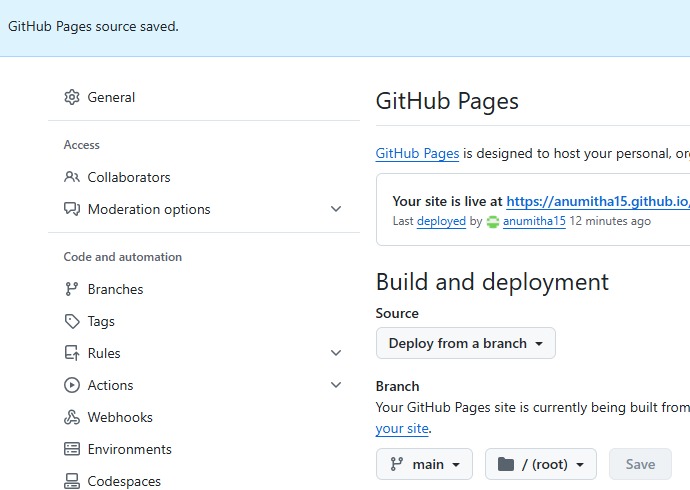
Select pages and configure to that pages to see the url for your website:



**Step 3:**

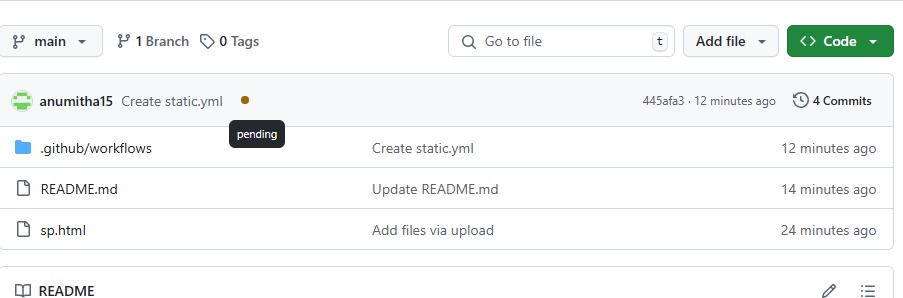
**Under pages go to settings to update the changes:**

o host a website on GitHub Pages, go to the repository's settings and scroll to the "Pages" section. In the "Source" dropdown, select the branch (usually main or master) where your website files are located. You can choose to host the site from the root directory or a subfolder, like /docs/. After saving the settings, GitHub will provide a URL (e.g., username.github.io/repository-name) to access your live site. The website will automatically update whenever you push changes to the selected branch.



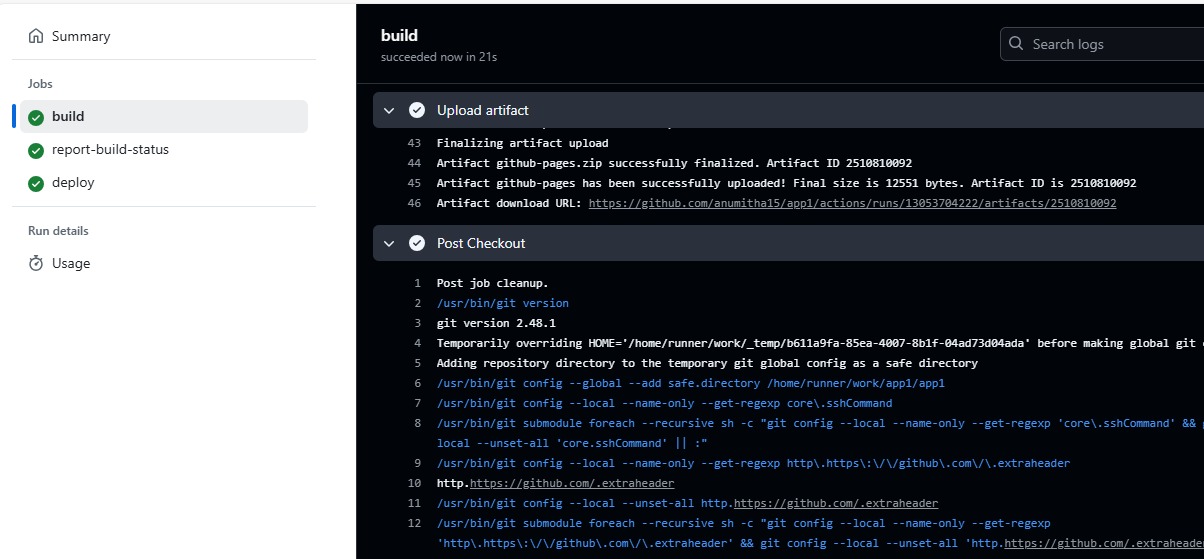
**Step 4:**

**Click the yellow configured button to review the process:**



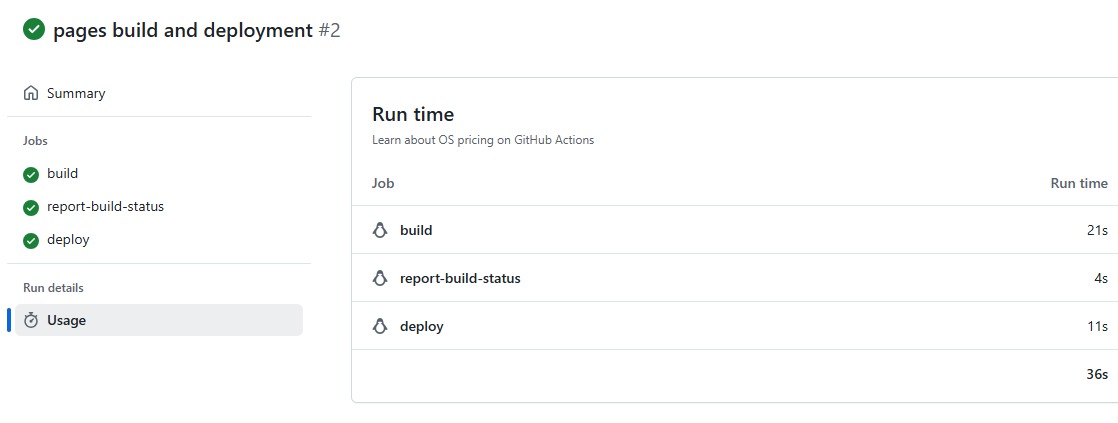
**Step 5:**

**Check the backend process behind the github pages:**



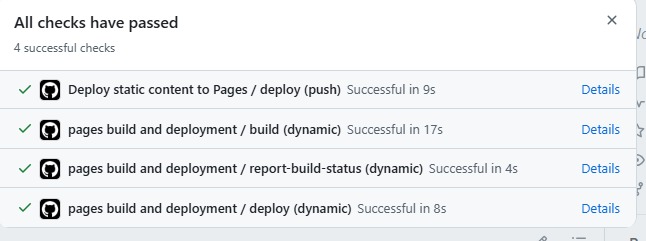
**Step 6:**

**Check the build,report-status,deploy process to successfully host website**



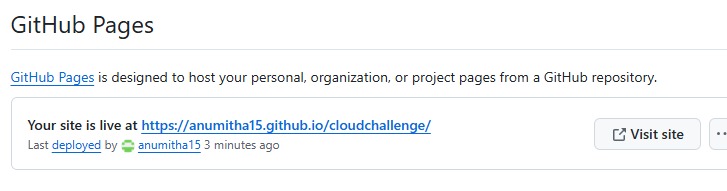
**Step 7:**

**All the four checks to process have passed:**



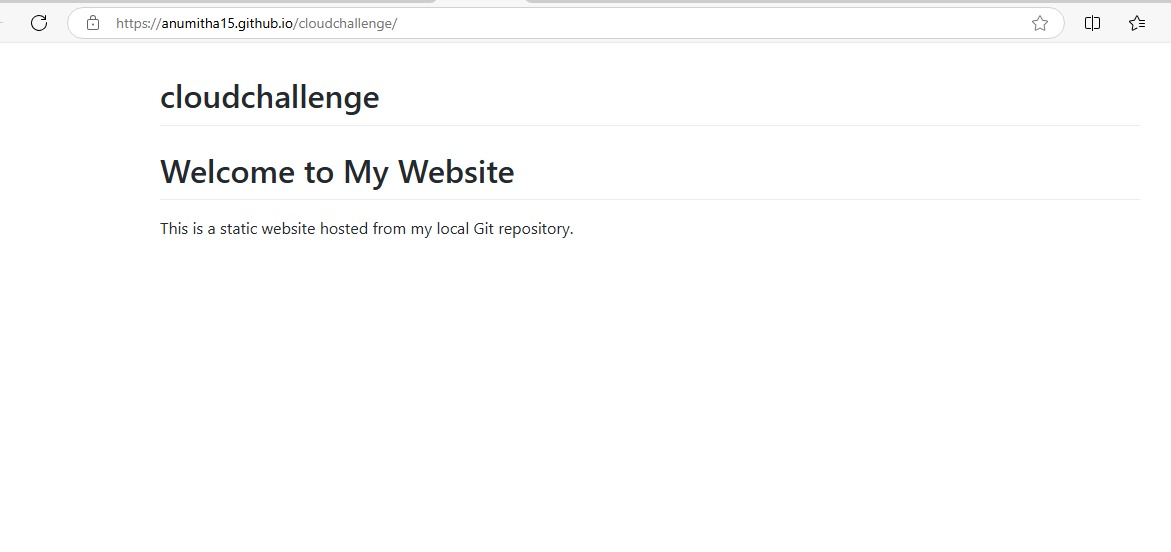
**Step 8:**

**Github url for the static website:**



**Step 9:**

**The website got successfully hosted using github provided url:**



**Expected outcome:**

1. Live Website: A fully functional, publicly accessible website hosted on GitHub Pages with a unique URL (e.g., username.github.io/repository-name).
2. Automatic Deployment: Any changes pushed to the repository's selected branch (e.g., main) will automatically update the live site, streamlining the deployment process.
3. Easy Access and Sharing: The website is easily accessible by anyone with the URL, making it simple to share your project, portfolio, or documentation.
4. Version Control Integration: The website’s content is fully integrated with Git, allowing for version tracking and rollback if needed.
5. Scalability for Static Sites: Your website will be reliably hosted without the need for backend infrastructure, ideal for static content, personal projects, portfolios, and documentation.