**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

**DAY 7 TASK**:

Deploy your static website using Github Pages : Host your local Git repository’s static website directly using Github pages

Name: SHAHANA.M.S Department: ADS



**Introduction**

GitHub Pages is a free hosting service provided by GitHub that allows developers to deploy static websites directly from a Git repository. It is a simple and efficient way to host portfolios, documentation, personal blogs, and other static sites without needing a separate hosting provider. By using GitHub Pages, developers can publish their websites quickly with version control and collaboration benefits.

**Objective**

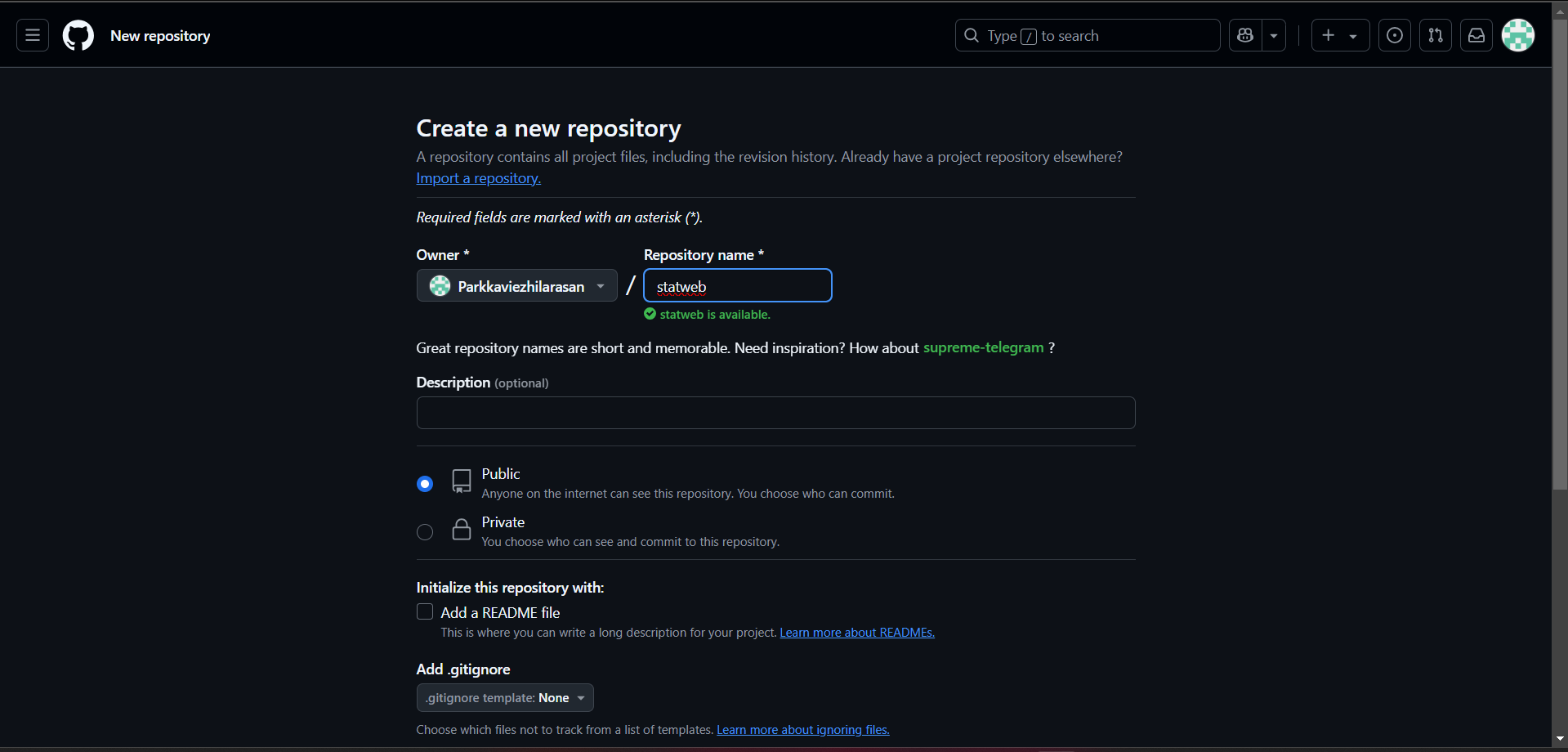
* To deploy a static website directly from a GitHub repository.
* To understand the process of enabling GitHub Pages for a project.
* To learn how to manage website updates using Git and GitHub workflows.
* To enhance project visibility and accessibility through a hosted webpage.

**Importance**

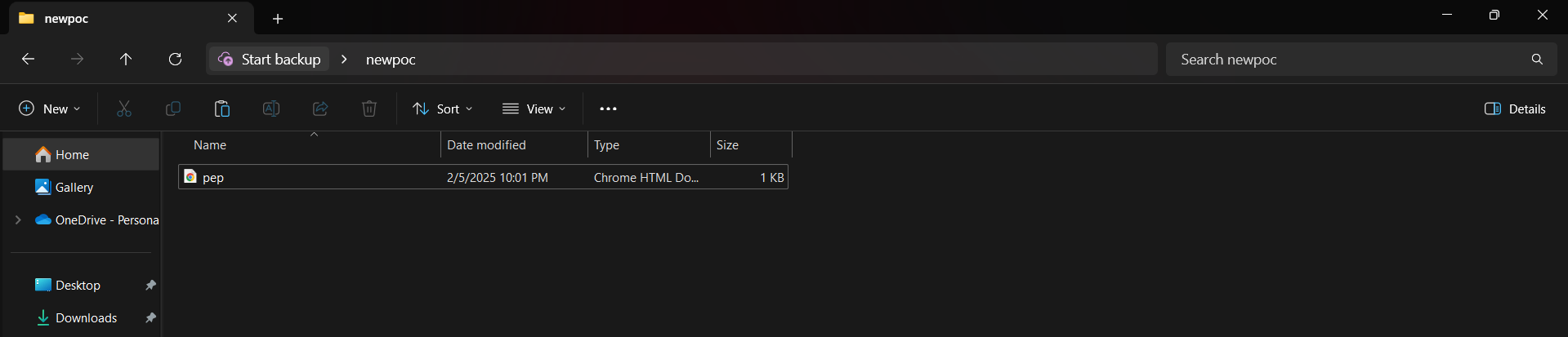
* **Free and Reliable Hosting**: GitHub Pages provides a cost-effective and stable hosting solution for static websites.
* **Easy Deployment**: No additional hosting setup is required; deployment is done with simple Git commands.
* **Version Control**: Since the website is stored in a Git repository, changes and updates are well-managed and trackable.
* **Ideal for Documentation & Portfolios**: Many developers and organizations use GitHub Pages to host project documentation and personal portfolios.
* Gaining hands-on experience with Git-based website hosting.

**Step-by-Step Overview**:

Step 1: Create a New Repository: Once you're logged in, click the green "New" button on the top- right of your GitHub homepage to create a new repository.



Step 2: Create a folder where you’ll keep all your website files. Inside that folder, create the main file for your website, called pep.html



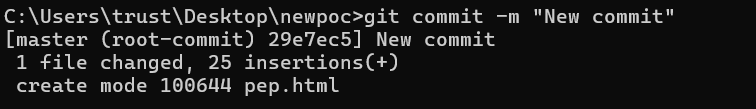
Step 3: Open Command Prompt and navigate to the folder where your index.html file is saved. Use the cd command to navigate.

C:\Users\trust\Pictures\Screenshots\Screenshot 2025-02-05 222227.png

Step 4: Initialize a Git repository by running:

C:\Users\trust\Pictures\Screenshots\Screenshot 2025-02-05 222234.png

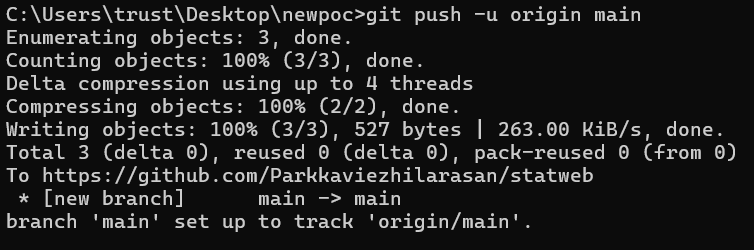
Step 5: Add your website files to the repository and commit the change with a message.



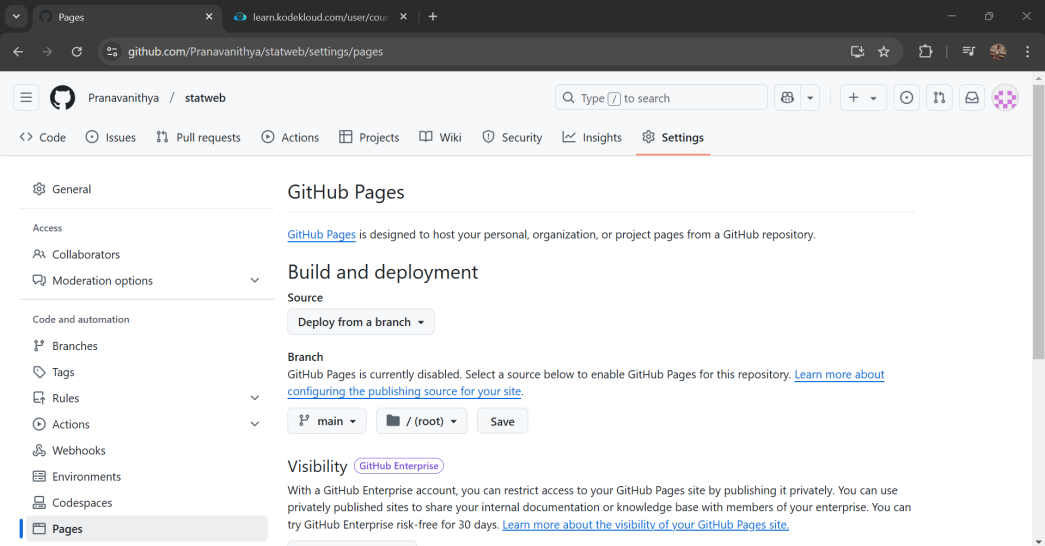
Step 6: Go to your GitHub repository . Copy the repository URL and link your local repository to the GitHub repository.

C:\Users\trust\Pictures\Screenshots\Screenshot 2025-02-05 222620.png

Step 7: Push your files to GitHub:

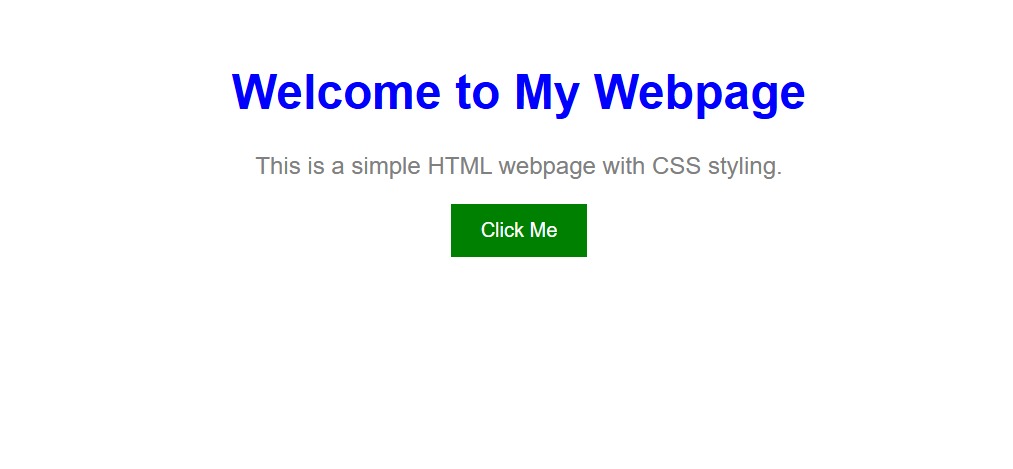


Step 8: To Enable GitHub Pages, Go to your repository on GitHub. >Click on the Settings tab. >Scroll down to the Pages. > Under Source, select: Branch: main and Folder: / (root). > Click Save.



Step 9:Visit your website at:

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



**OUTCOME:**

* Successfully deploying a static website using GitHub Pages.
* Gaining hands-on experience with Git-based website hosting.
* Understanding how to maintain and update a hosted website via Git.
* Learning best practices for organizing a static site within a repository.