**Placement Empowerment Program**

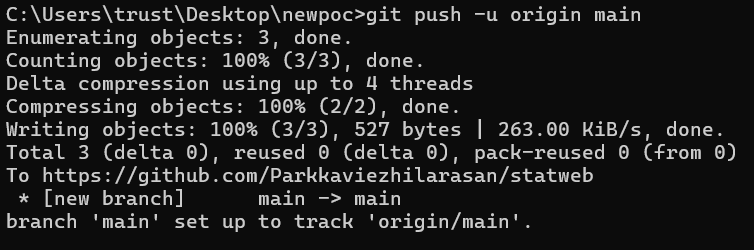
***Cloud Computing and DevOps Centre***

**DAY 7 TASK**:

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

Name: Parkkavi E Department: ADS





**INTRODUCTION :**

GitHub Pages is a static site hosting service designed to publish your projects directly from a GitHub repository. It allows developers to showcase their work, create personal websites, or host documentation in an efficient, free, and straightforward way.

**OVERVIEW :**

This project demonstrates how to deploy a static website using GitHub Pages. Starting with the basics of setting up a GitHub repository, we’ll explore each step required to host a functional static website. This includes initializing a Git repository, pushing files to GitHub, and configuring GitHub Pages for deployment.

**Key Features of GitHub Pages:**

Free hosting for public repositories. Support for static files (HTML, CSS, JavaScript). Easy integration with version control through Git. **OBJECTIVES**:

1. Learn the fundamentals of GitHub Pages and its deployment process.

2. Understand the importance of static website hosting and its use cases.

3. Gain hands-on experience in using Git and GitHub for project versioning and hosting.

4. Successfully publish a static website and make it publicly accessible.

**Importance of Hosting with GitHub Pages:**

1. Cost-effective: Free for public repositories, making it accessible for students and developers

2. Version Control: Seamlessly integrates with GitHub, enabling easy updates and collaboration.

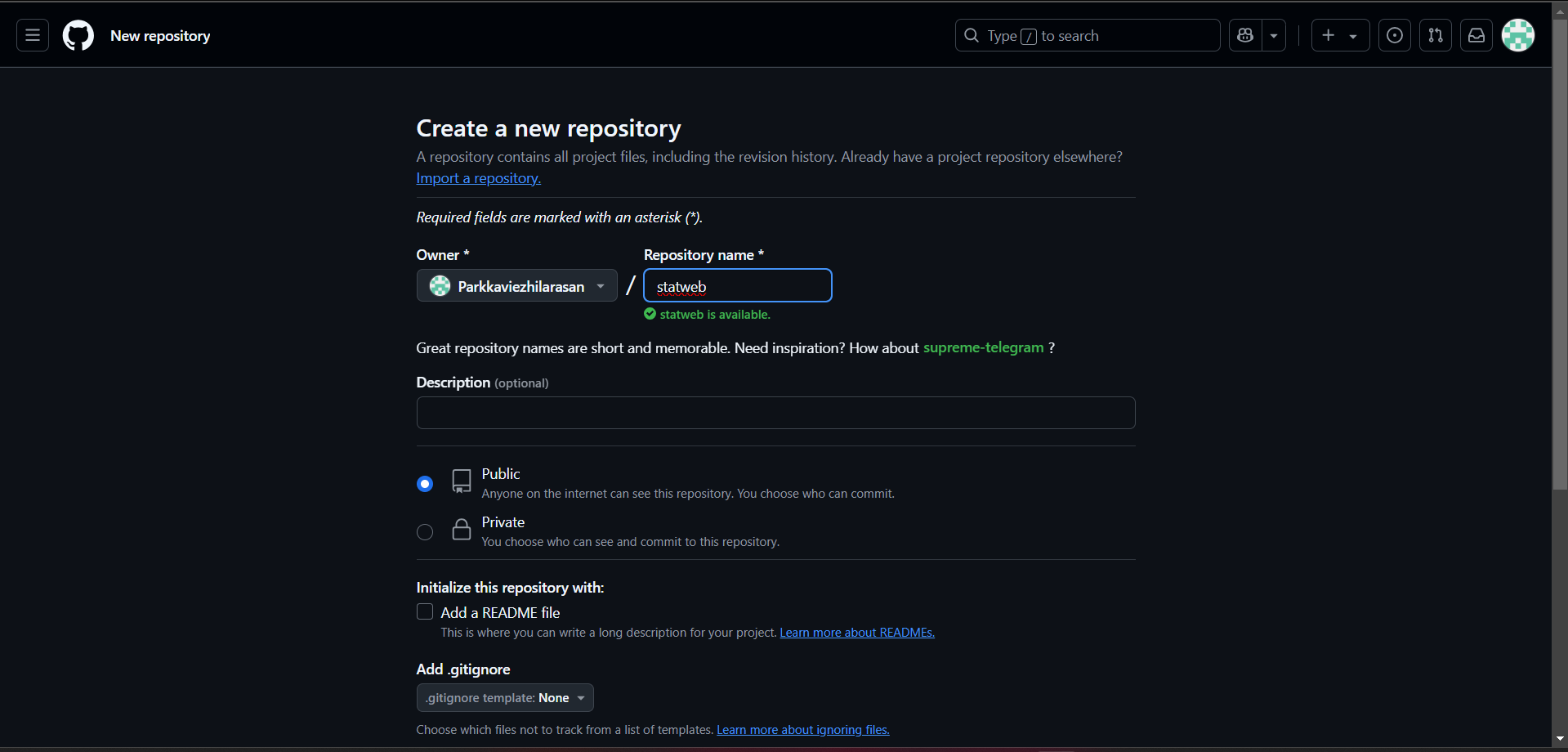
3. Visibility: A great way to showcase personal portfolios, projects, or documentation.

4. Ease of Use: Minimal setup required compared to other hosting platforms.

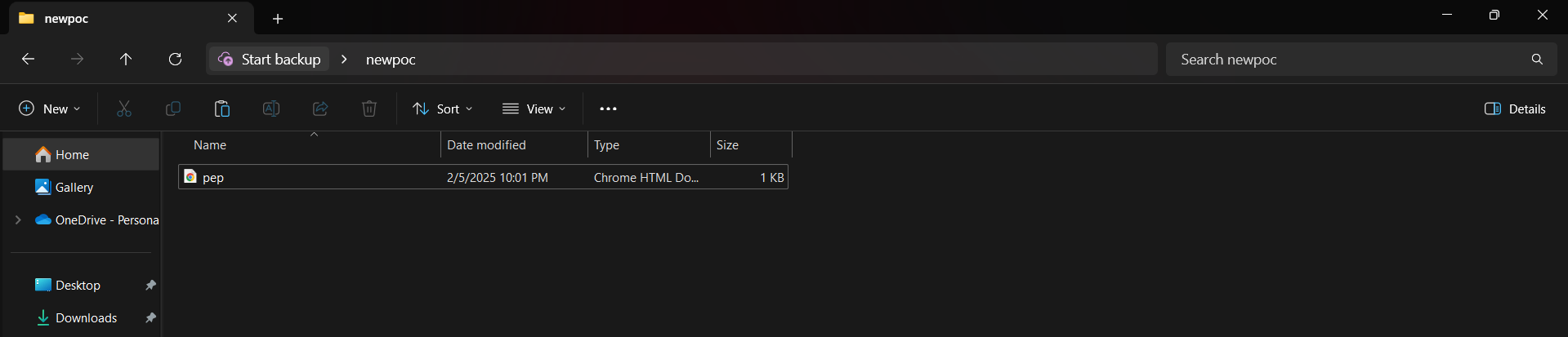
5. Custom Domains: Option to configure custom domains, enhancing the professional appeal of your website.

**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



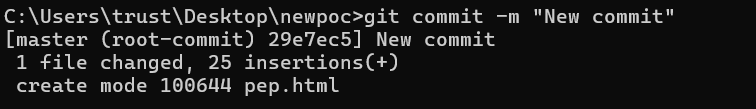
Step 3: Open Command Prompt and navigate to the folder where your index.html file is saved. Use the cd command to navigate.

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

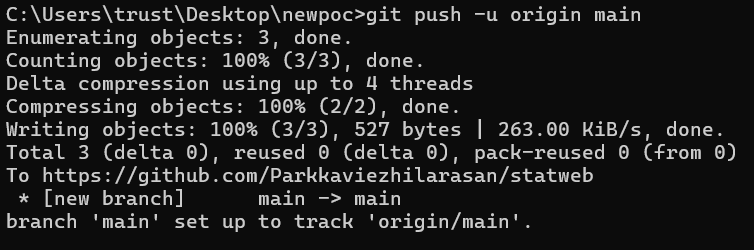
Step 5: Add your website files to the repository and commit the change with a message.



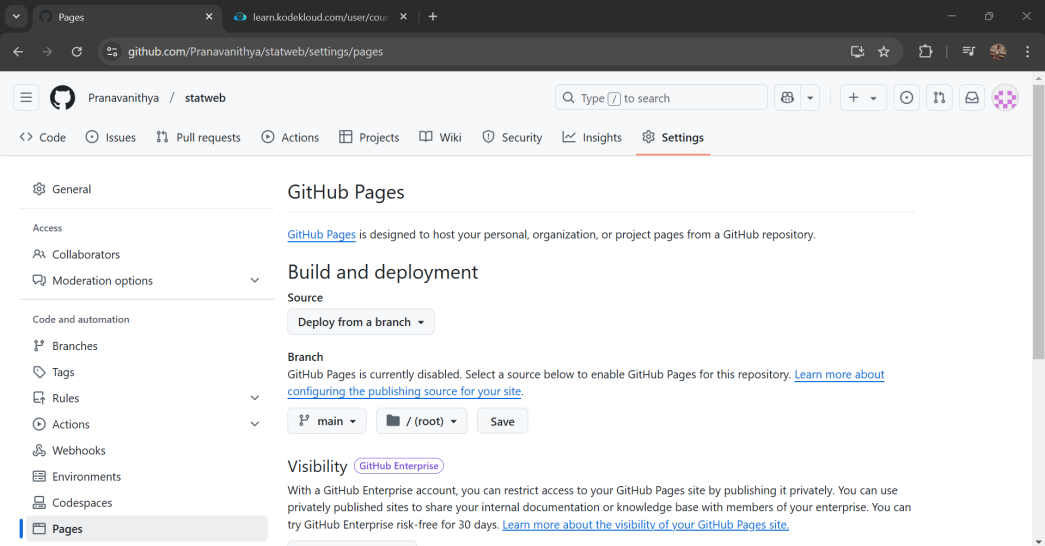
Step 6: Go to your GitHub repository . Copy the repository URL and link your local repository to the GitHub repository.

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

Step 7: Push your files to GitHub:

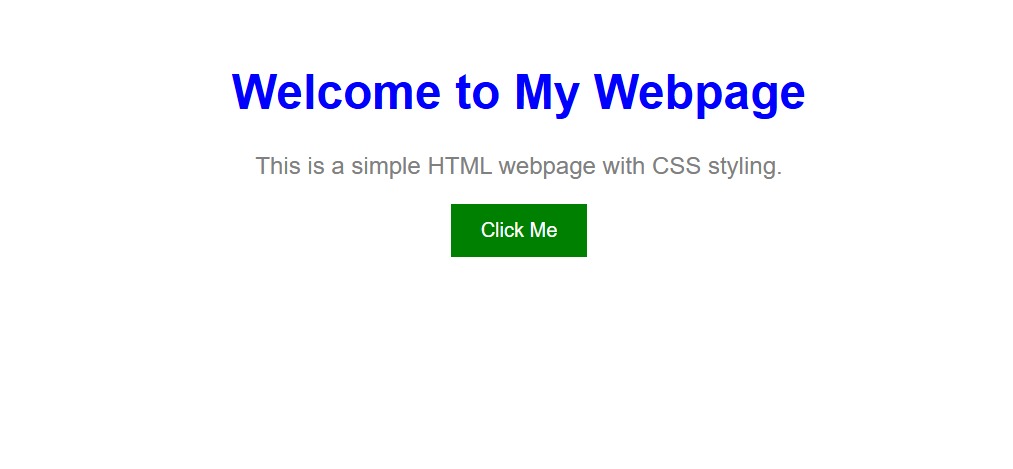


Step 8: To Enable GitHub Pages, Go to your repository on GitHub. >Click on the Settings tab. >Scroll down to the Pages. > Under Source, select: Branch: main and Folder: / (root). > Click Save.



Step 9:Visit your website at:

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



**OUTCOME:**

By completing this PoC of deploying a static website using GitHub Pages, you will

1)Upload your static website files (HTML, CSS, JavaScript) to GitHub.

2)Enable GitHub Pages in the repository settings to host your static website.

3)Access your static website live on the web via a GitHub Pages URL