**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

**Set up Git Branching**

Name: PARKKAVI E Department: ADS



GitHub Pages is a free service that allows you to host static websites directly from your GitHub repositories. It's an excellent choice for personal projects, portfolios, documentation, and more. By leveraging GitHub Pages, you can deploy your static website without the need for additional hosting services.

**Overview:**

1. **Create a Repository:** Start by creating a new repository on GitHub. For user or organization sites, name the repository as username.github.io or organization.github.io. For project sites, any repository name will suffice.

[docs.github.com](https://docs.github.com/articles/creating-project-pages-manually?utm_source=chatgpt.com)

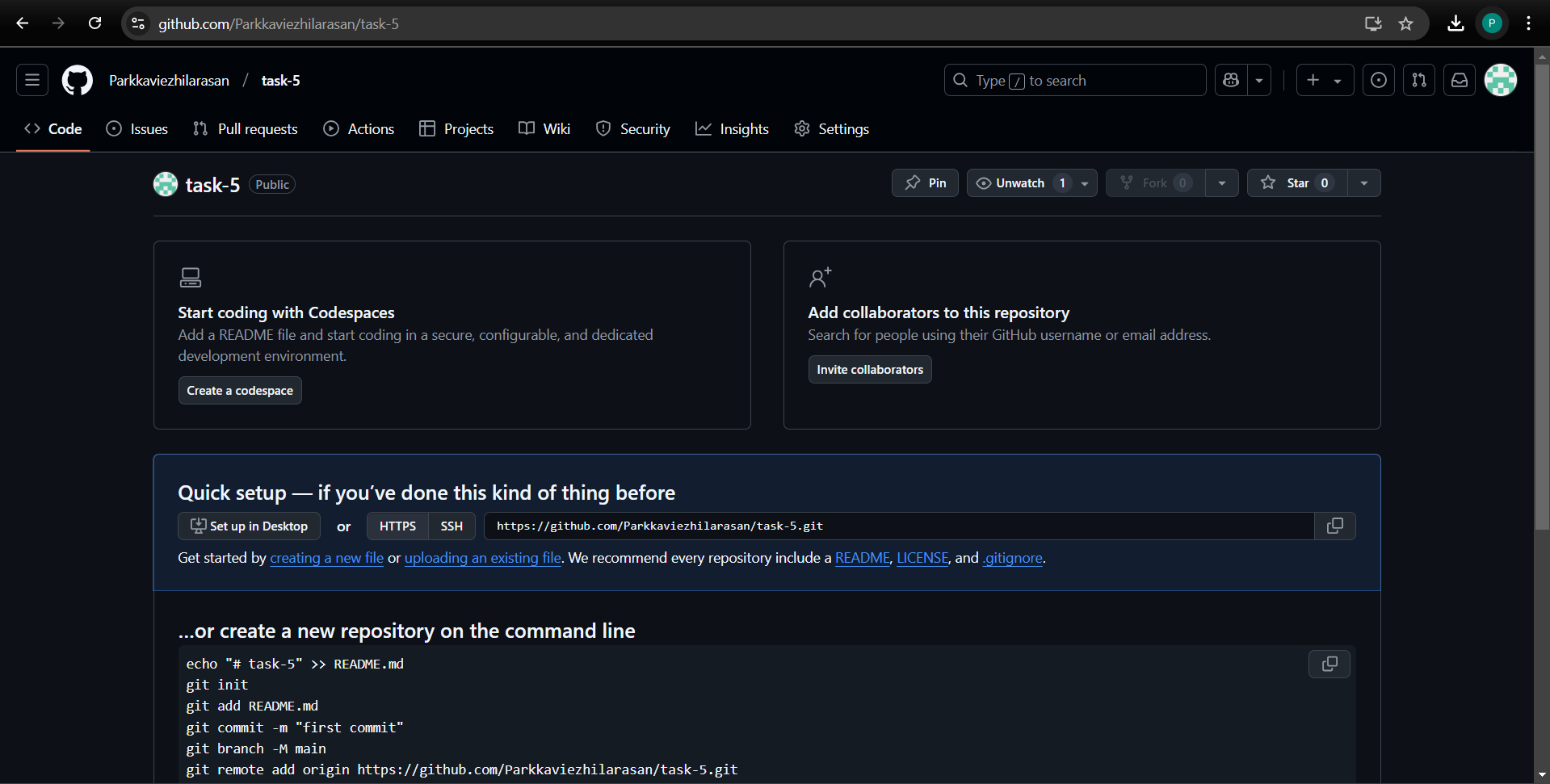
1. **Add Your Website Files:** Upload your HTML, CSS, JavaScript, and other assets to the repository. Ensure that your main HTML file is named index.html, as this will serve as the entry point to your site.
2. **Configure GitHub Pages:** Navigate to the repository's settings, find the "Pages" section, and select the branch and folder (if any) that contains your website files. GitHub will then build and deploy your site.

[docs.github.com](https://docs.github.com/articles/creating-project-pages-manually?utm_source=chatgpt.com)

1. **Access Your Site:** After deployment, your site will be accessible at https://username.github.io for user or organization sites, or https://username.github.io/repository-name for project sites.

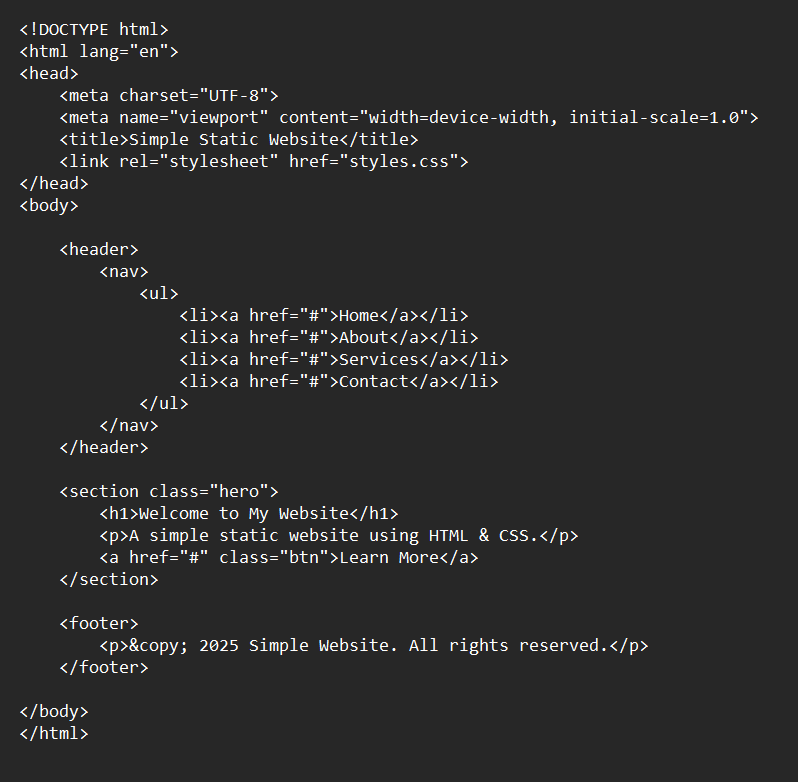
1. Create a GitHub Repository

* Go to [GitHub](https://github.com).
* Click on the + icon in the top right corner and select **New** repository.
* Give your repository a name (e.g., my-static-website).
* Choose Public.
* Select Initialize this repository with a README.
* Click Create repository.



2. Prepare Your Static Website

* Create a folder on your local machine and add your HTML, CSS, and JavaScript files.
* Make sure the main HTML file is named index.html, as this will be the homepage of your website.



3.Command

-Git init: To create a.git folder inside your folder, which tells Git to start tracking files.

-Create a simple file to start the repo

-Add the File to git

-Save this changes in Git with commit –m messages

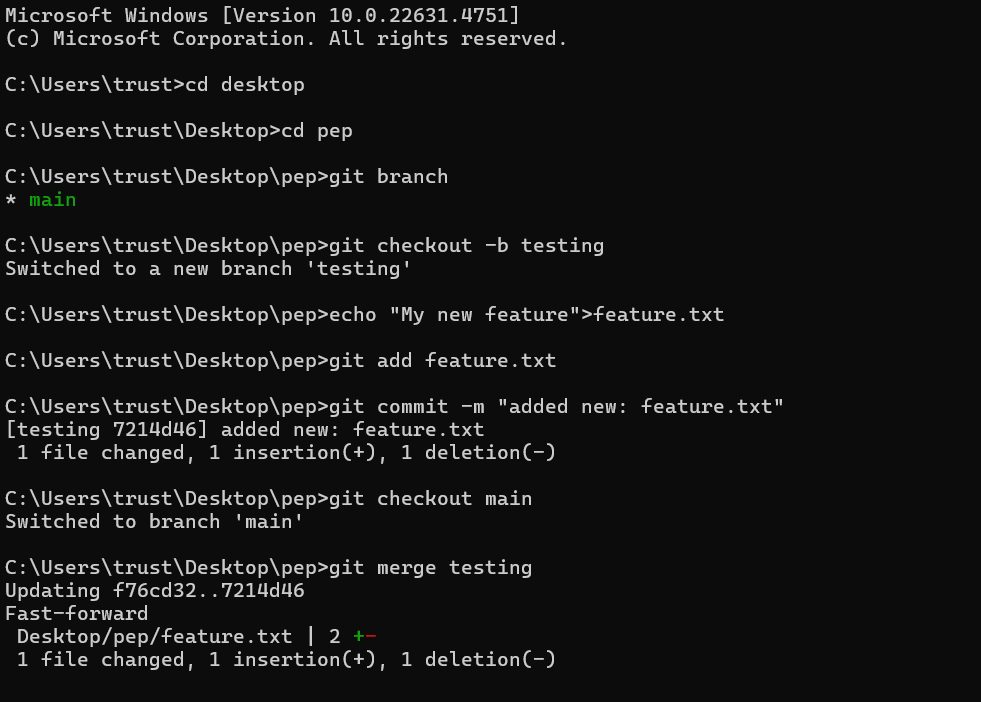
-Create and switch to a new bramch called testing-feature

- Add new file to our feature

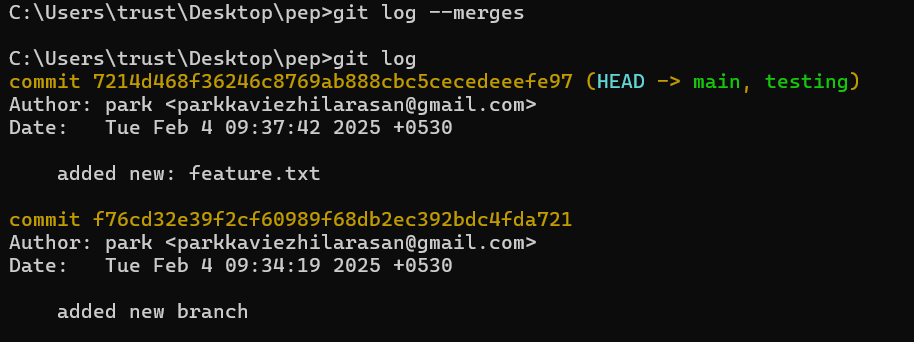
-Use git add . command to add all the files

- Commit the changes

- Merge changes from testing-feature to master



**3.Chechking the log**

****