**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

**Set Up a Local Git Repository**:

Initialize Git repository locally and version control your static website

Name: Yasvanth Rajan E Department: AML



**Introduction and Overview**

Version control is a critical component in modern software development, allowing teams and individuals to track changes, collaborate efficiently, and maintain a history of their work. Git, a distributed version control system, is widely used for its flexibility, speed, and powerful features. This Proof of Concept (POC) document outlines the process of setting up a local Git repository and using it to version control a static website. This will enable efficient management of website updates and facilitate collaboration among developers.

**Objectives**

* Understand the basics of Git and version control.
* Learn how to initialize a Git repository locally.
* Apply version control to a static website.
* Perform basic Git operations such as adding, committing, and viewing changes.

**Importance**

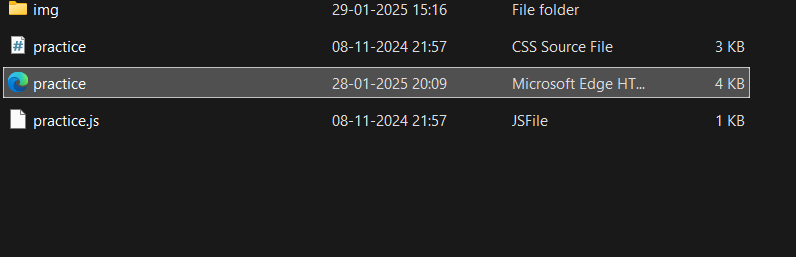
* **Enhanced Collaboration:** Enables multiple developers to work on the same project without overwriting each other's work.
* **Change Tracking:** Maintains a detailed history of code changes, making it easy to identify and revert to previous versions if needed.
* **Backup and Recovery:** Provides a secure backup of code that can be restored in case of data loss.
* **Efficient Project Management:** Helps manage different versions of the project efficiently, facilitating smooth feature development and bug fixing.

**STEPS:**

* **PRE-REQUISITE**: Ensure Git is installed on your local machine. You can verify this by running git --version.

**STEP 1: Create a Directory with Website Files:**

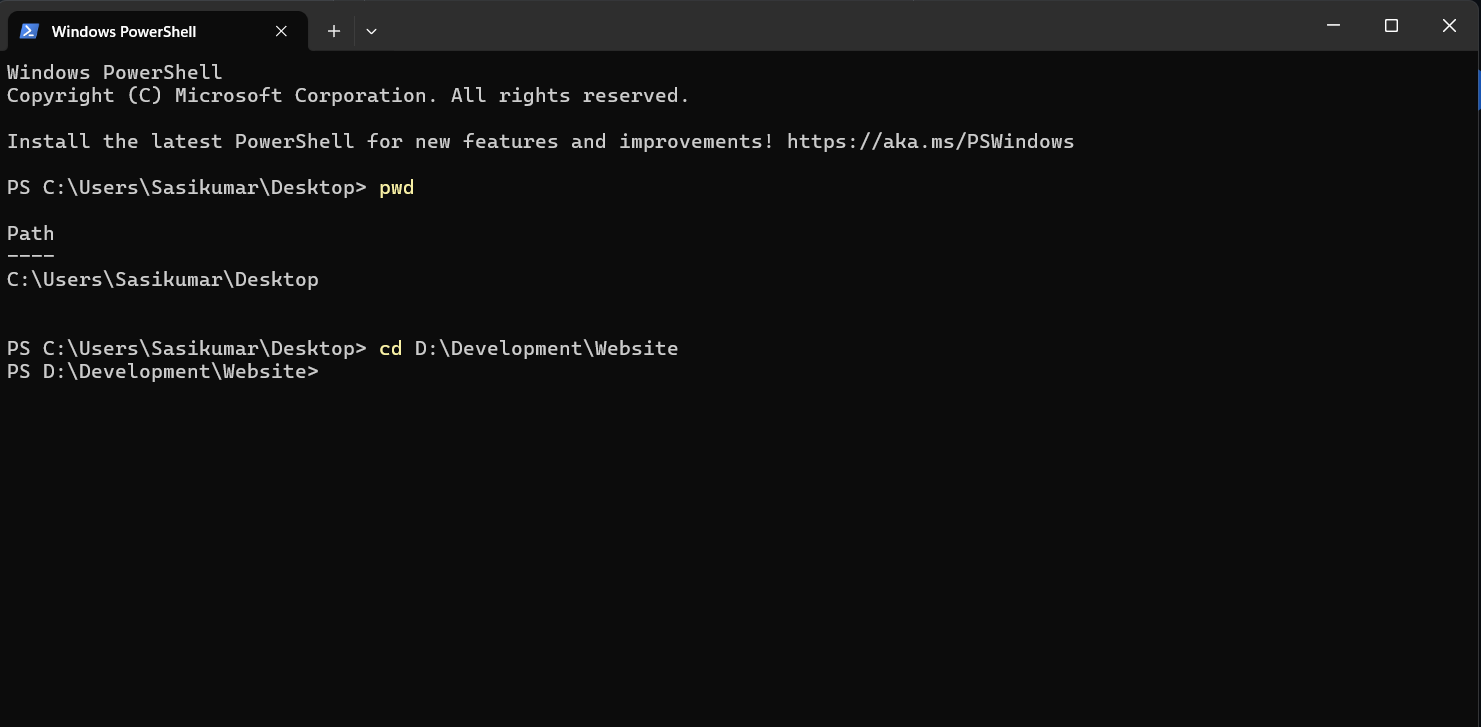
* Create a folder on your local machine containing all your static website files (HTML, CSS, JS, etc.).



**STEP 2: Navigate to the Directory:**

* Open the terminal or command prompt.
* Use the cd command to change the current directory to the folder containing your website files.

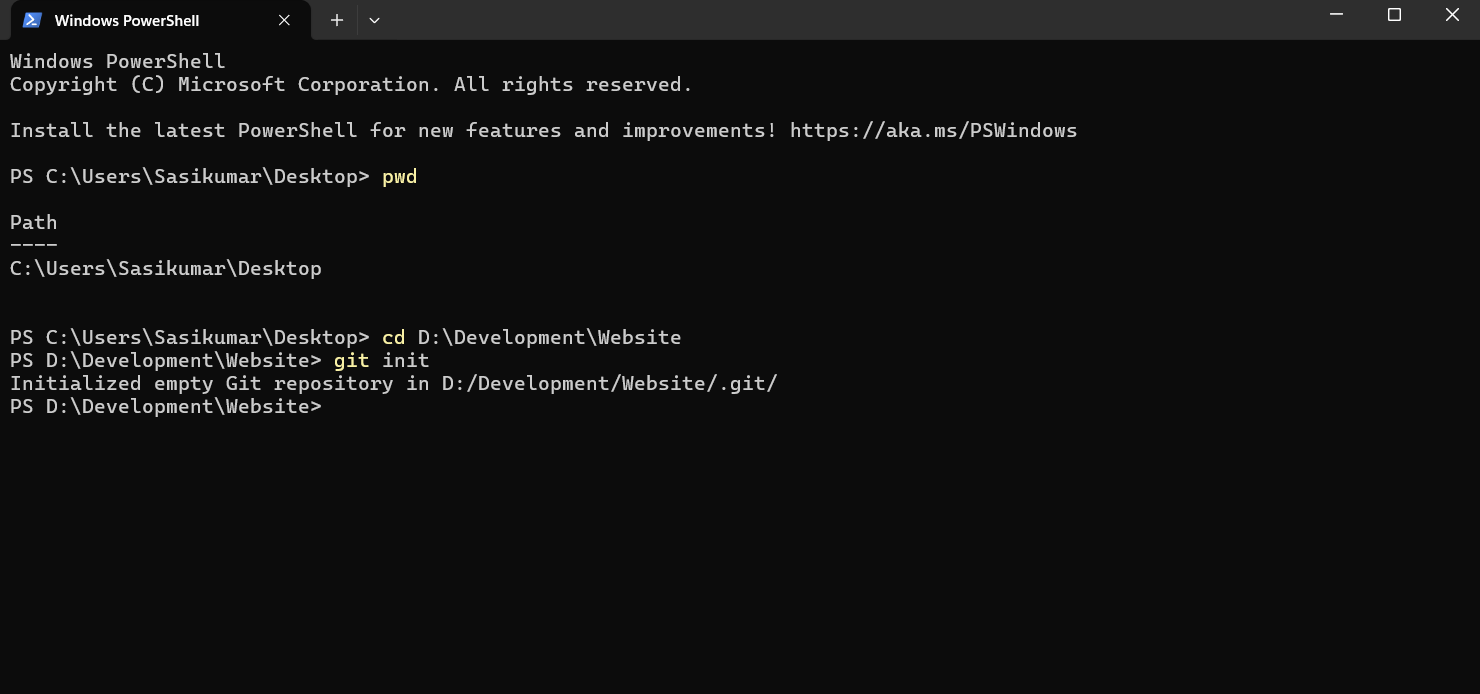
cd path/to/your/website-folder



**STEP 3: Initialize a Local Git Repository:**

* Run the following command to initialize an empty Git repository in your project folder:

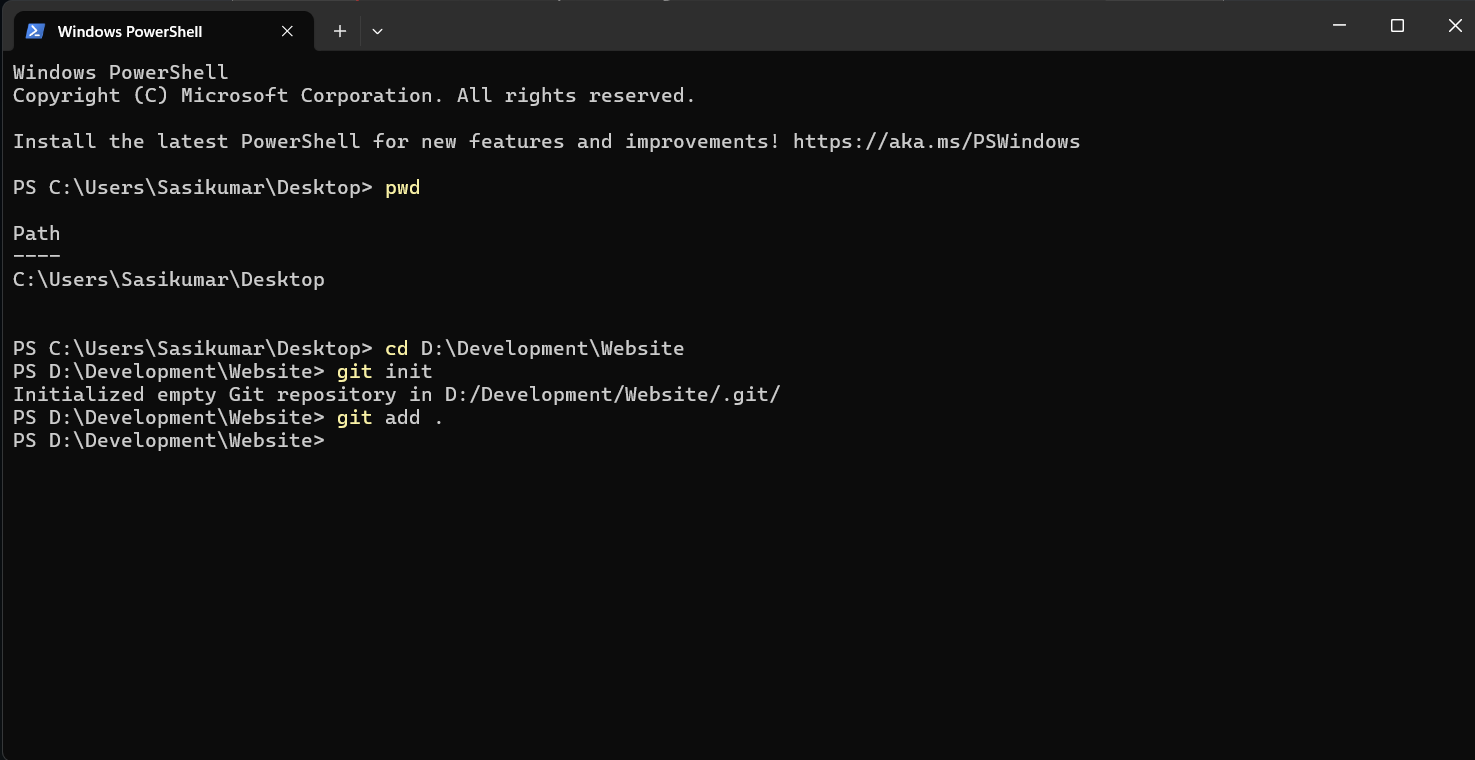
git ini**t**

****

**STEP 4: Add Files to the Repository:**

* Add all files to the Git staging area using:

git add .

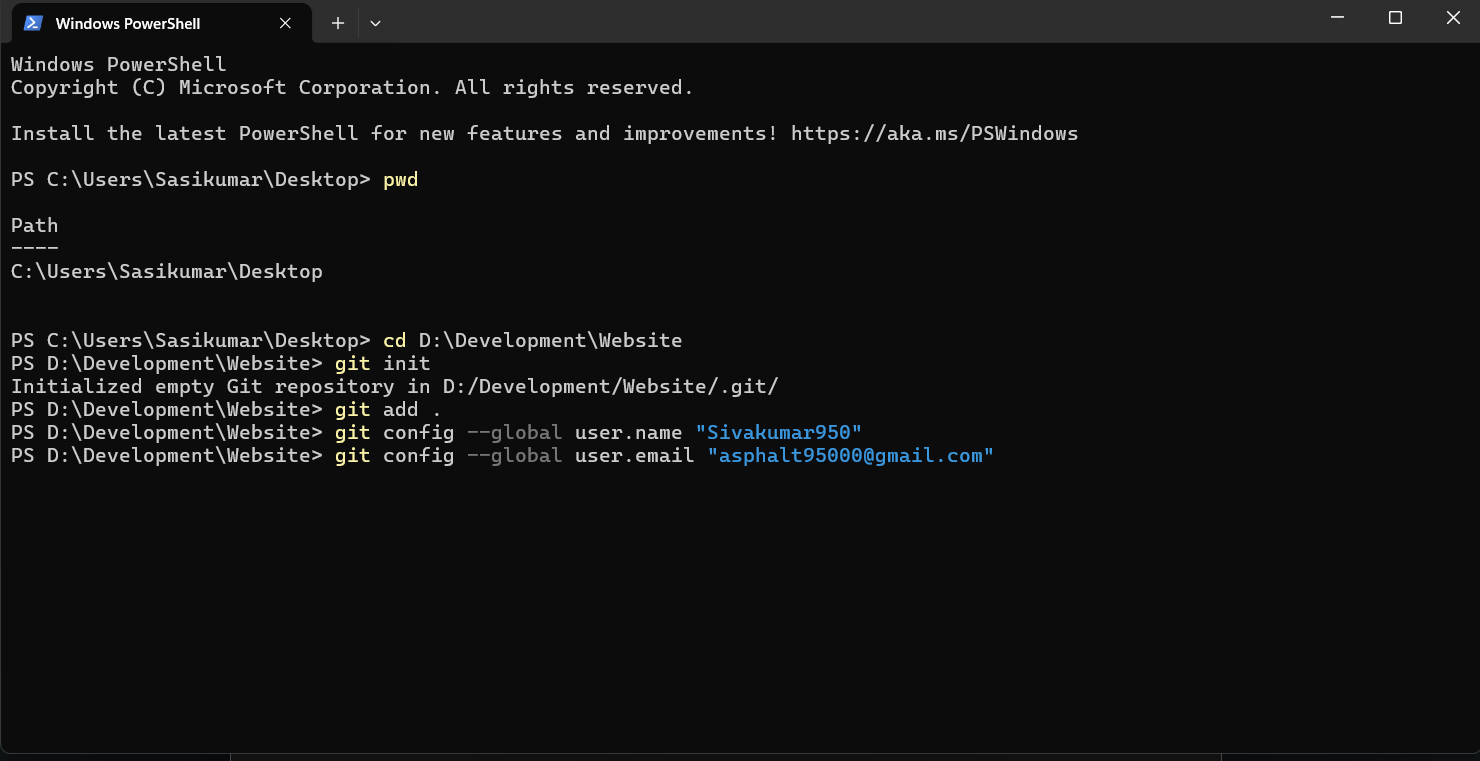


**STEP 5: Configure Git Settings:**

* Set up your Git username and email (if not configured already):

git config --global user.name "Your Name"

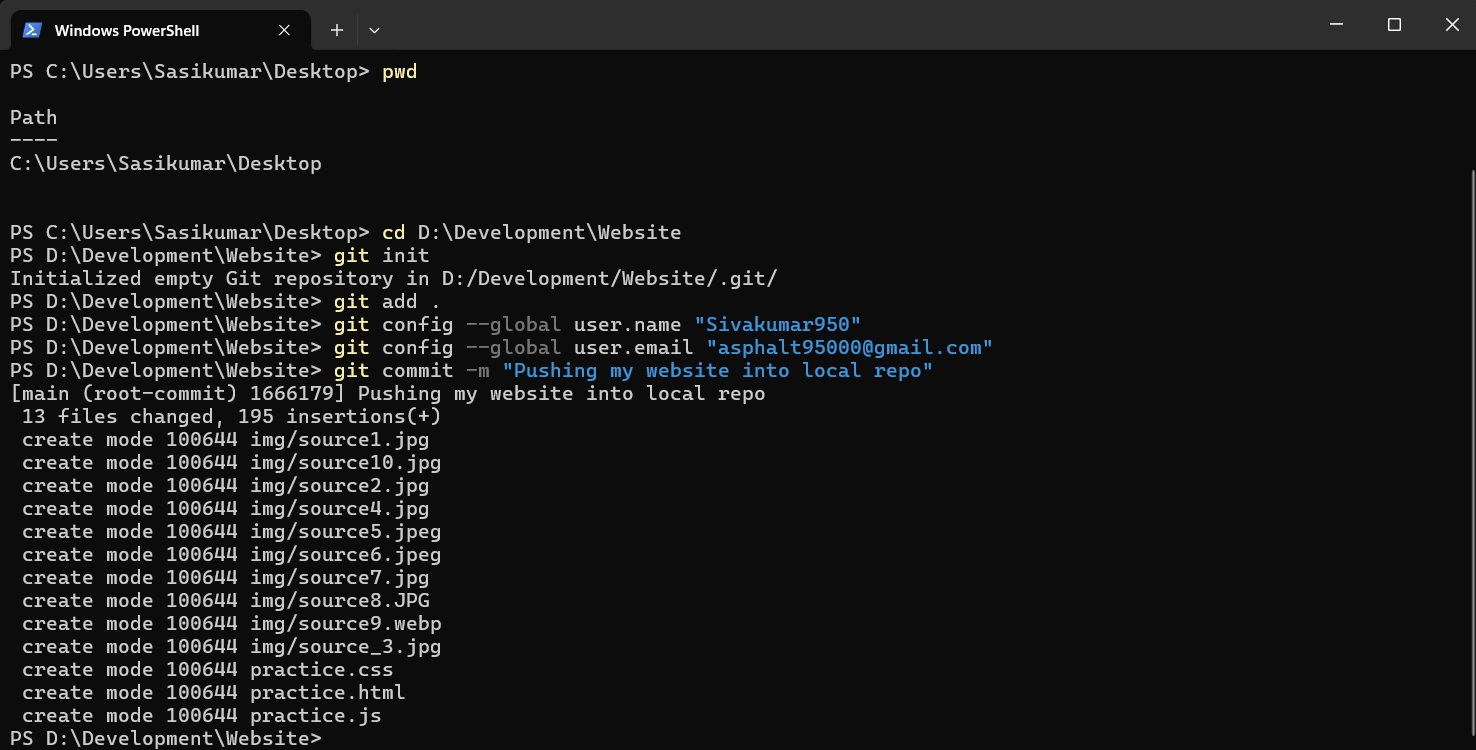
git config --global user.email [your.email@example.com](mailto:your.email@example.com)



**STEP 6: Commit the Files:**

* Commit the added files with a meaningful commit message:

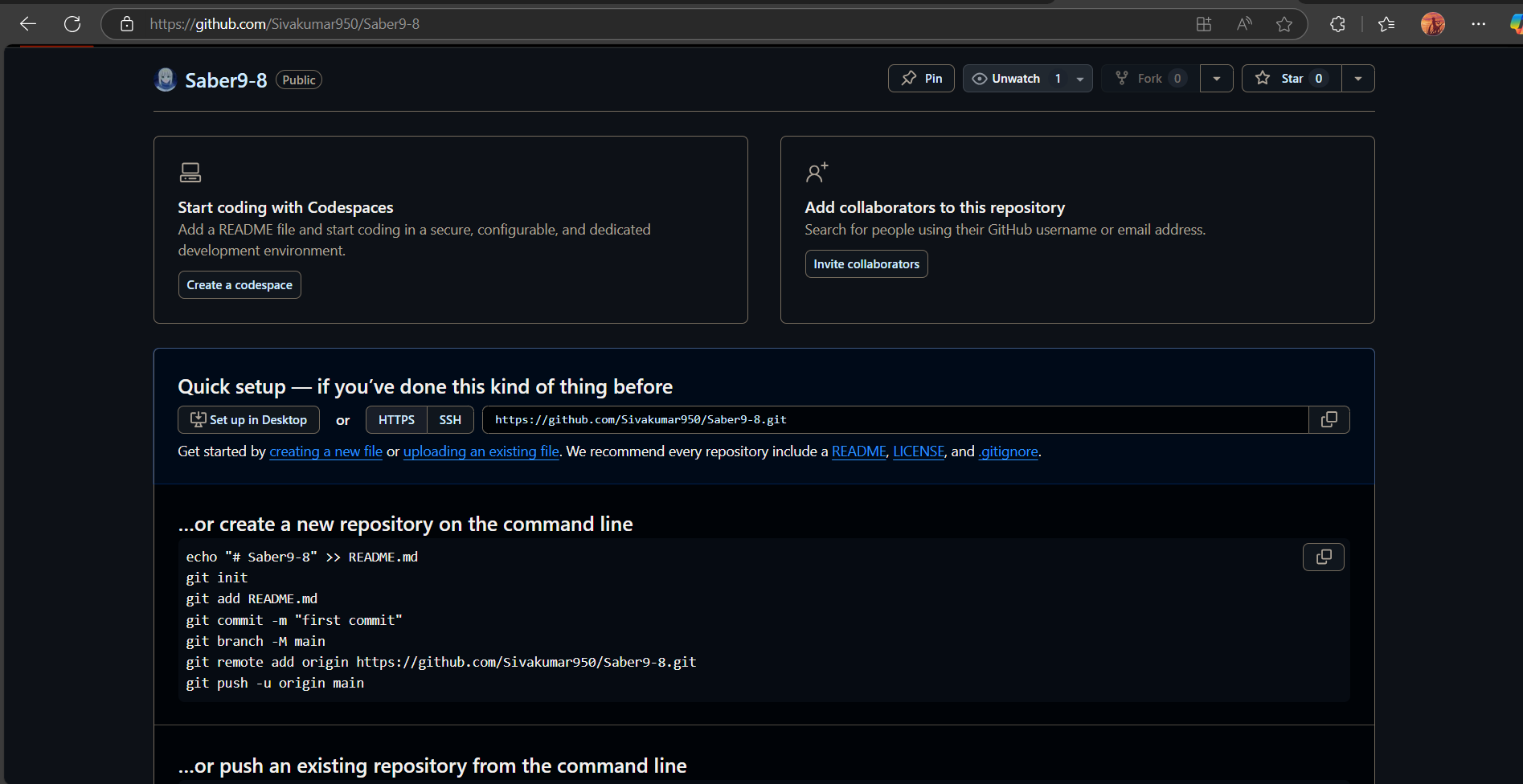
git commit -m "Initial commit with website files"

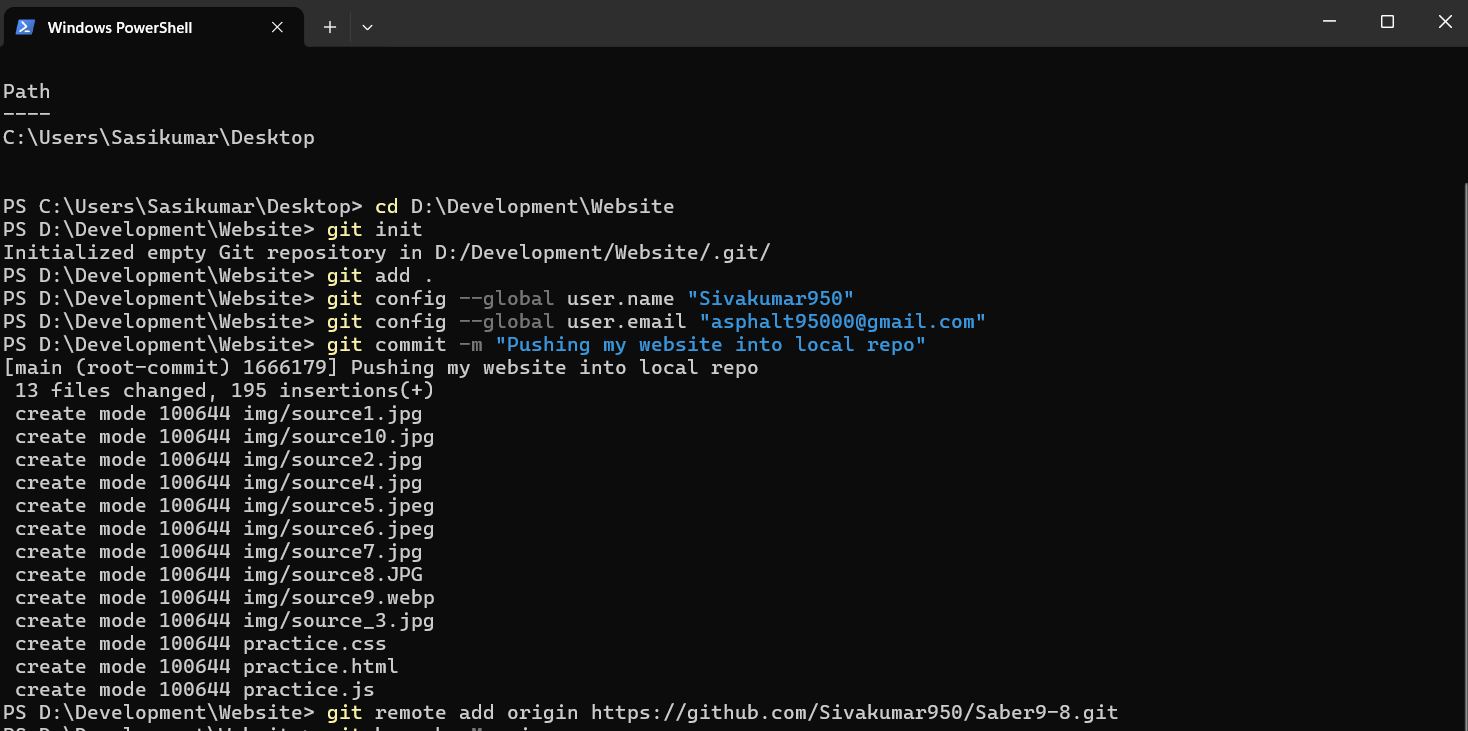


**STEP 7: Add Remote Repository:**

* Copy the Git remote command provided when you create a repository on GitHub and paste it into your terminal:

git remote add origin <https://github.com/yourusername/your-repo.git>

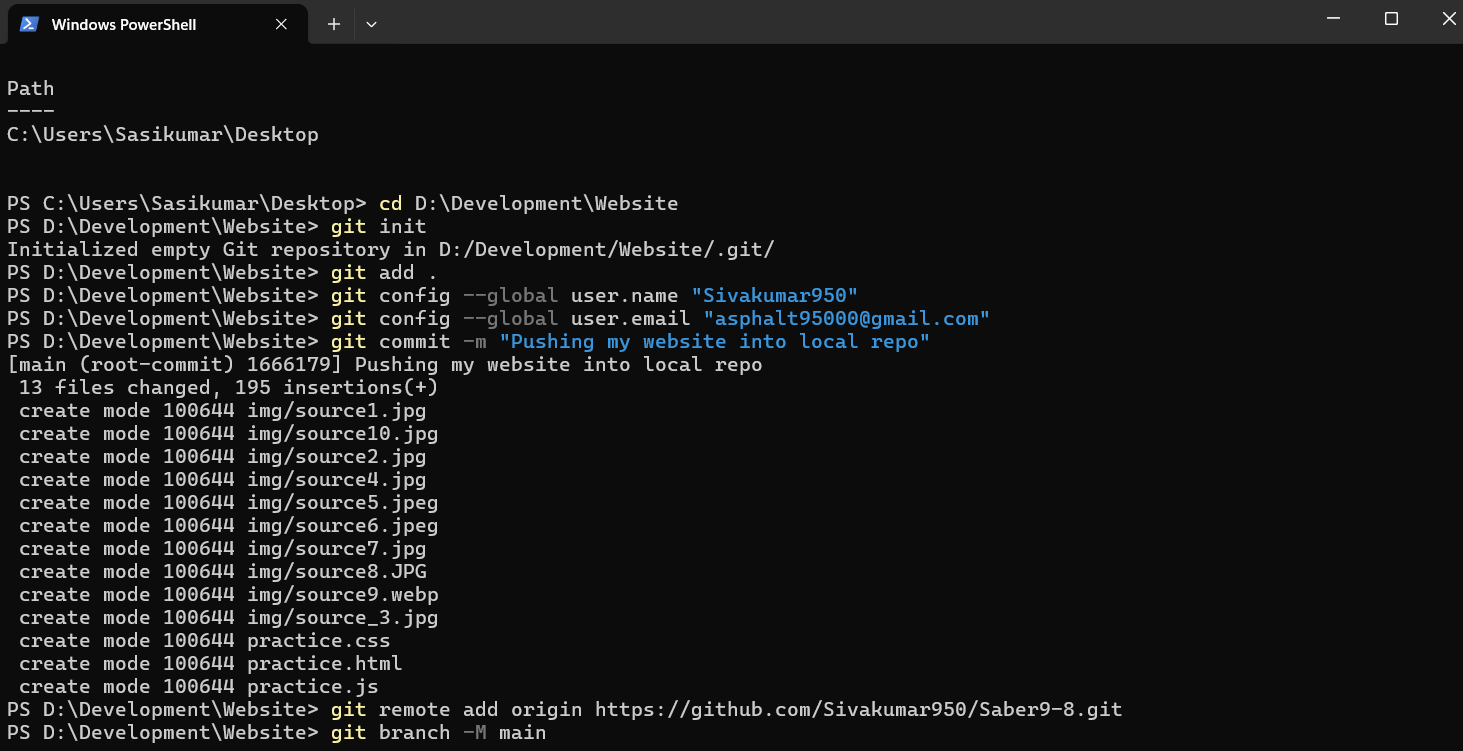




**STEP 8: Create the Main Branch:**

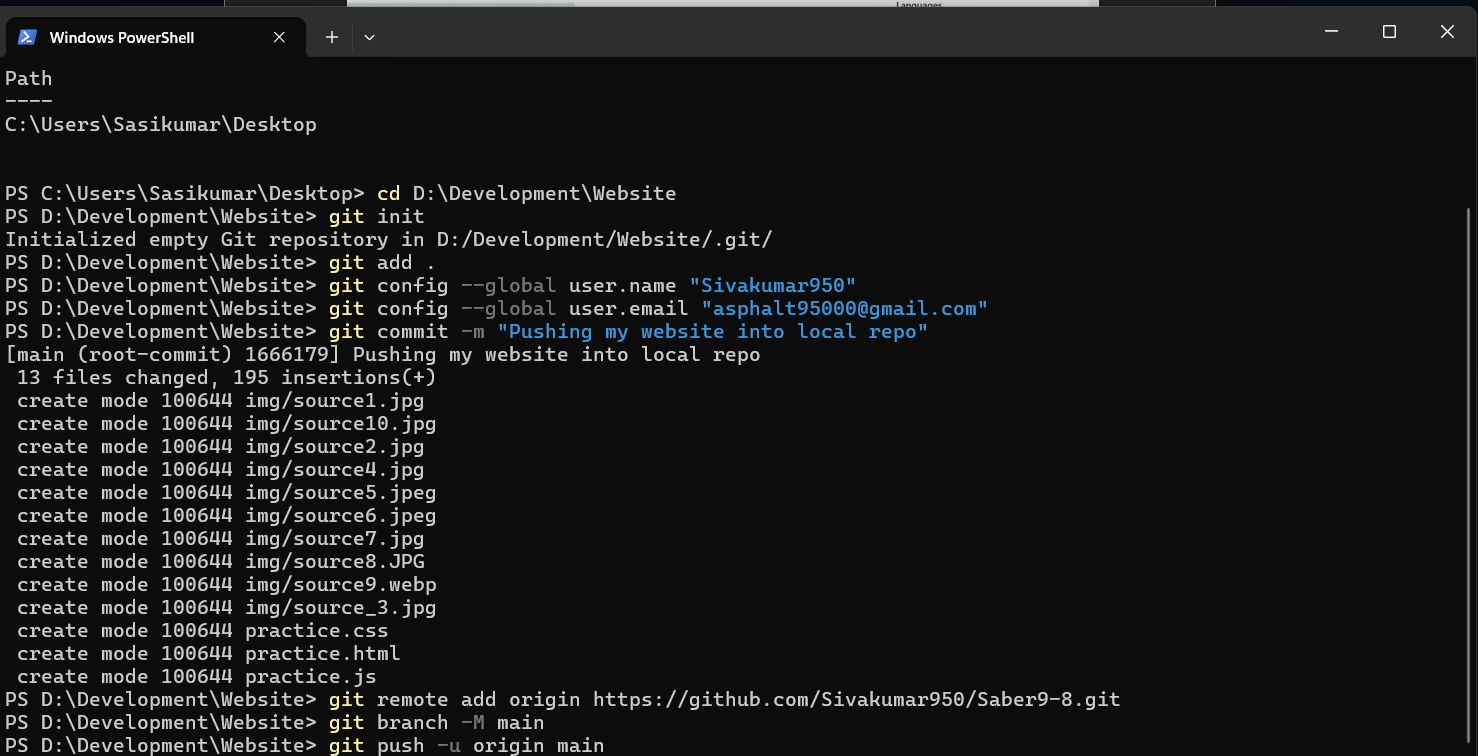
* Create and switch to the main branch:

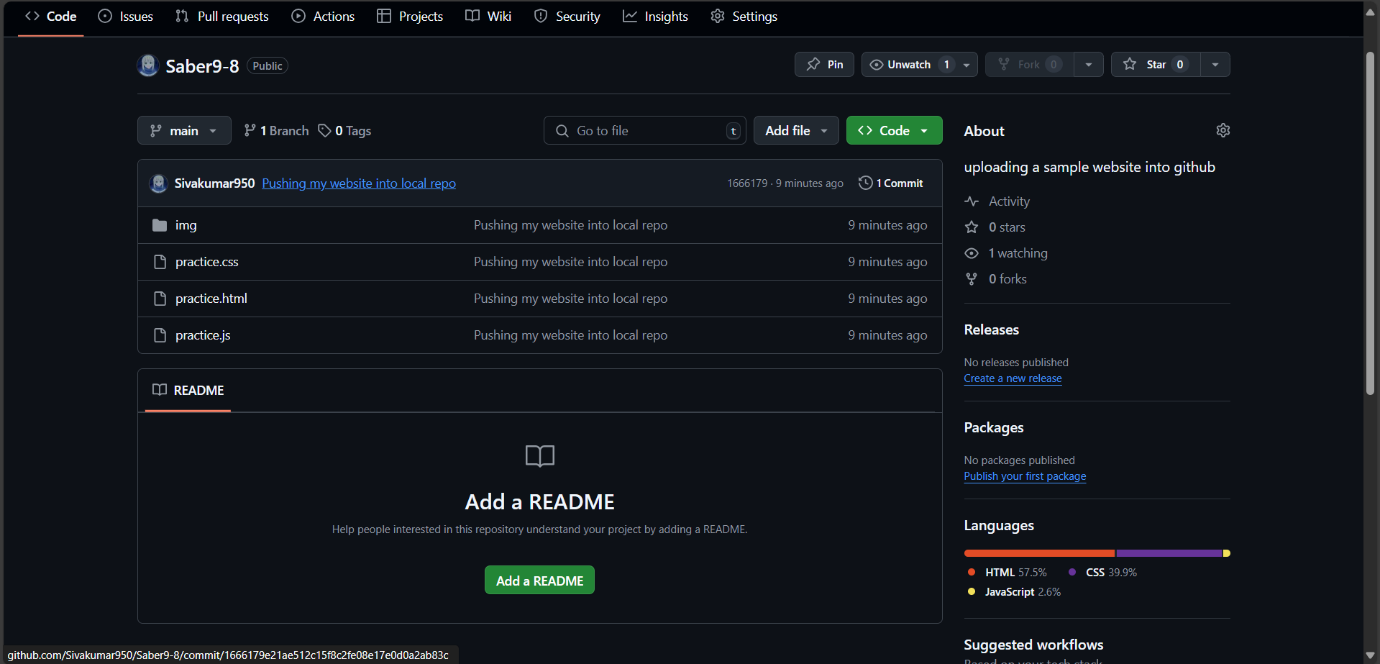
git branch -M main



**STEP 9: Push Files to GitHub Repository:**

* Copy the git push command provided when you created the GitHub repository.
* Paste and execute the command in your terminal to push your committed files to the remote GitHub repository.
* Once the command executes successfully, your static website files will be uploaded to your GitHub repository**.**

****

****