Day 1 Source Control Lab Assignment

**Objective**

Apply Git and GitHub skills from the Day 1 lab to create and manage a front-end project (HTML/CSS/JS), tracking multiple versions, uploading it to GitHub, and answering theoretical questions about Git’s structure and workflows.

Instructions

**Part 1: Create and Track a Front-End Project**

* **Set Up a New Project:**

Create a new folder for a front-end project.

Add an HTML file (e.g., a webpage with a heading and paragraph), a CSS file (e.g., styling for colors or fonts), and a JavaScript file (e.g., a button that shows an alert when clicked). Initialize the project as a Git repository to start version control.

Create a file to ignore unnecessary items (e.g., temporary files, node\_modules folder) to keep the repository clean.

**Save Version 1:**

Track all project files (HTML, CSS, JS) and save the initial state as the first version of your project.

Ensure the ignore file is included to exclude irrelevant files.

Make Changes and Save Version 2:

Modify the project (e.g., change the CSS to add a hover effect or update the JavaScript to change the alert message).

Save these changes as the second version of your project.

If you encounter issues (e.g., extra files showing up or incorrect changes), troubleshoot and fix them (e.g., undo a bad change or update the ignore file).

**Part 2: View History and Upload to GitHub**

View the project’s history to confirm both versions (initial and updated) are recorded.

Compare the differences between the two versions to see what changed (e.g., new CSS rules).

**Upload to GitHub:**

Create a new public repository on GitHub for your project.

Link your local project to this repository and upload all changes.

Set up the project to be hosted online using GitHub Pages, so the webpage is accessible via a live URL.

**Part 3: Theoretical Questions**

* What is the purpose of a Git repository, and how does it help front-end developers manage a project like an HTML/CSS/JS webpage?
* Explain the difference between tracked and untracked files in Git, and why might untracked files cause issues in a front-end project?
* Why is it important to use a file to ignore certain items (e.g., node\_modules) in a Git repository, especially for front-end development?
* How does viewing a project’s history benefit a front-end developer when working on multiple versions of a webpage?
* What is the role of GitHub in relation to a local Git repository, and how does hosting a project on GitHub Pages add value for a front-end developer

Good Luck