Lab: Git and GitHub Basics Using Windows CMD (with Existing HTML and CSS Files)

Prerequisites:

1. Install **Git**: Download and install Git from git-scm.com.
2. Create a **GitHub** account: If you don't have one, sign up at GitHub.com.
3. Your existing `index.html` and `style.css` files.

Steps:

1. Set Up Git on Windows CMD

1. Open **Command Prompt** (CMD).
2. Configure Git with your username and email:

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

2. Initialize a Local Git Repository

1. Navigate to the folder containing your existing `index.html` and `style.css` files:

```
cd path\to\your\project-folder
```

2. Initialize the folder as a Git repository:

```
git init
```

3. Stage and Commit the HTML File

1. Stage the existing HTML file:

```
git add index.html
```

2. Commit the HTML file:

```
git commit -m "Added index.html"
```

4. Create a Repository on GitHub

1. Go to GitHub and create a **new repository** (name it `my-website`).
2. Do **not** initialize the repository with any files.

5. Connect Local Repository to GitHub

1. Add the GitHub repository as the remote:

   ```
   git remote add origin https://github.com/your-username/my-website.git
   ```

6. Push the HTML File to GitHub

1. Push the current commit (HTML file) to GitHub:

   ```
   git push -u origin master
   ```

7. Create a New Branch to Add the CSS File

1. Create a new branch called `add-css` and switch to it:

   ```
   git checkout -b add-css
   ```

2. Stage the existing `style.css` file:

   ```
   git add style.css
   ```

3. Commit the CSS file:

   ```
   git commit -m "Added style.css for page styling"
   ```

8. Merge the `add-css` Branch Into `master`

1. Switch back to the `master` branch:

   ```
   git checkout master
   ```

2. Merge the `add-css` branch into `master`:

   ```
   git merge add-css
   ```

9. Push the Changes to GitHub

1. Push the updated `master` branch (with both `index.html` and `style.css`) to GitHub:

   ```
   git push origin master
   ```

Summary of Commands:

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"

cd path\to\your\project-folder
git init

git add index.html
git commit -m "Added index.html"
```

```
git remote add origin https://github.com/your-username/my-website.git
git push -u origin master

git checkout -b add-css
git add style.css
git commit -m "Added style.css for page styling"

git checkout master
git merge add-css
git push origin master
```

Student Task

Question: Working with GitHub Branches

Task:

1. Create a new branch **directly on GitHub** using the GitHub web interface (e.g., name it `feature-branch`).
2. Fetch this new branch to your **local machine** using Git.
3. Switch to this branch locally, make a **small change** (e.g., modify or create a file), and **commit** the change.
4. **Push** your changes to the remote `feature-branch` on GitHub.
5. **Merge** the `feature-branch` into `master` (or `main`) directly on GitHub using the **Pull Request** feature.

Hints:

- To create a new branch on GitHub, look for the branch dropdown menu (labeled `main` or `master`) and type the name of your new branch.
- After creating the branch, use `git fetch origin` to fetch it on your local machine.
- To switch to the branch locally, use `git checkout branch-name`.
- After committing your changes, use `git push origin branch-name` to push your updates to GitHub.
- To merge the branch, go to the **Pull Requests** tab on GitHub and create a pull request to merge your branch into `master` or `main`.

Solution

Steps to Create a Branch on GitHub

1. **Go to Your Repository**:
   - Navigate to your repository on GitHub (e.g., `https://github.com/your-username/my-website`).
2. **Select the Branch Dropdown**:

   - On your repository page, find the **branch dropdown** located just above your file list (it usually says "main" or "master").
   - Click the dropdown to view the current branches.
3. **Create a New Branch**:
   - In the dropdown, you'll see a search box. Type the name of your new branch (e.g., `add-css`).
   - GitHub will suggest creating a new branch with that name. Click on the suggestion that says **"Create branch: add-css"**.
4. **Switch to the New Branch**:
   - After creating the branch, you'll automatically switch to that branch. You can now make changes to the codebase specific to this branch.

Working with the Branch Locally

After creating the branch on GitHub, you can switch to it locally in your Git setup and continue working:

1. **Fetch the New Branch**:
   - In your local terminal or CMD, fetch the latest changes (including the new branch):

```
git fetch origin
```

2. **Switch to the New Branch**:
   - Switch to the new branch locally:

```
git checkout add-css
```

3. **Make Changes Locally**:
   - Make any changes to your files, commit them, and push the changes to the new branch:

```
git add .
git commit -m "Made changes to CSS"
git push origin add-css
```

Merging the Branch on GitHub

Once you're done with your work on the `add-css` branch, you can merge it into the `master` or `main` branch on GitHub.

1. **Go to the Pull Request Tab**:
   - In your GitHub repository, click on the **Pull Requests** tab.
2. **Create a New Pull Request**:
   - Click **New pull request**.
   - You'll see a comparison view. The base should be `main` or `master`, and the compare branch should be `add-css`.
3. **Submit the Pull Request**:
   - Review the changes and click **Create Pull Request**.

- Once the pull request is approved (if working in a team), you can merge the changes by clicking **Merge Pull Request**.

Summary of Commands for Local Setup:

After creating a branch on GitHub:

```
git fetch origin
git checkout add-css
```

After working on the branch locally:

```
git add .
git commit -m "Changes to CSS"
git push origin add-css
```