# Lab: Git Branching and Pull Requests

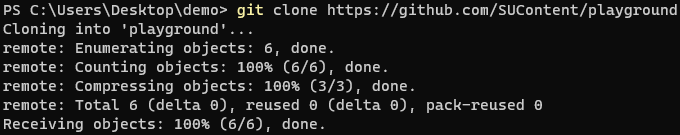
Lab for the ["Software Engineering and DevOps"](https://softuni.bg/trainings/4333/software-engineering-and-devops-october-2023) course @ SoftUni.

## Using Git Commands

First, let's start by opening a CLI, for example PowerShell or Terminal.

After that, let's try to clone an existing Git repository. The full repo URL is below:  
<https://github.com/SUContent/playground>

Use the following command:

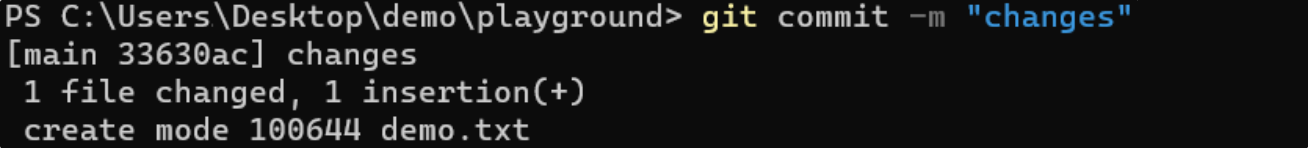


Now, let's try to make local changes. In order to do that, we'll make some changes in the Readme.md file.

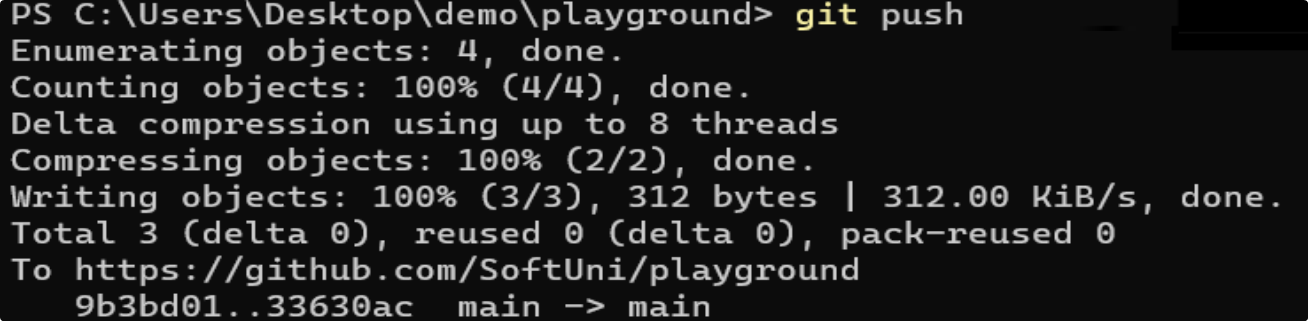
After we are done with the changes, it's time to add (prepare) the files for commit. We'll do this using the following command:



After that, we should commit added files to the local repository using the command below:



Finally, we should push all committed changes to the remote repository.



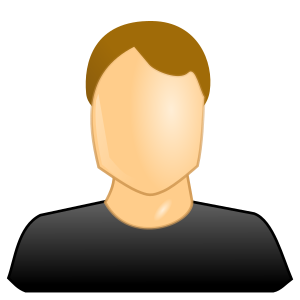
## Git Conflict Scenario

Let's imagine that three developers work on a **shared project** with Git. **All** of them try to change and push the **same file**. By doing so, a **conflict** will occur on pushing the **changes**.

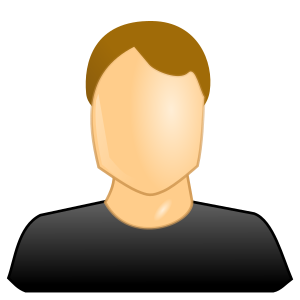




Maria



George



Peter

**push**

**push**

**push**



**conflict**



**conflict**



change  
+ commit



change  
+ commit



change  
+ commit

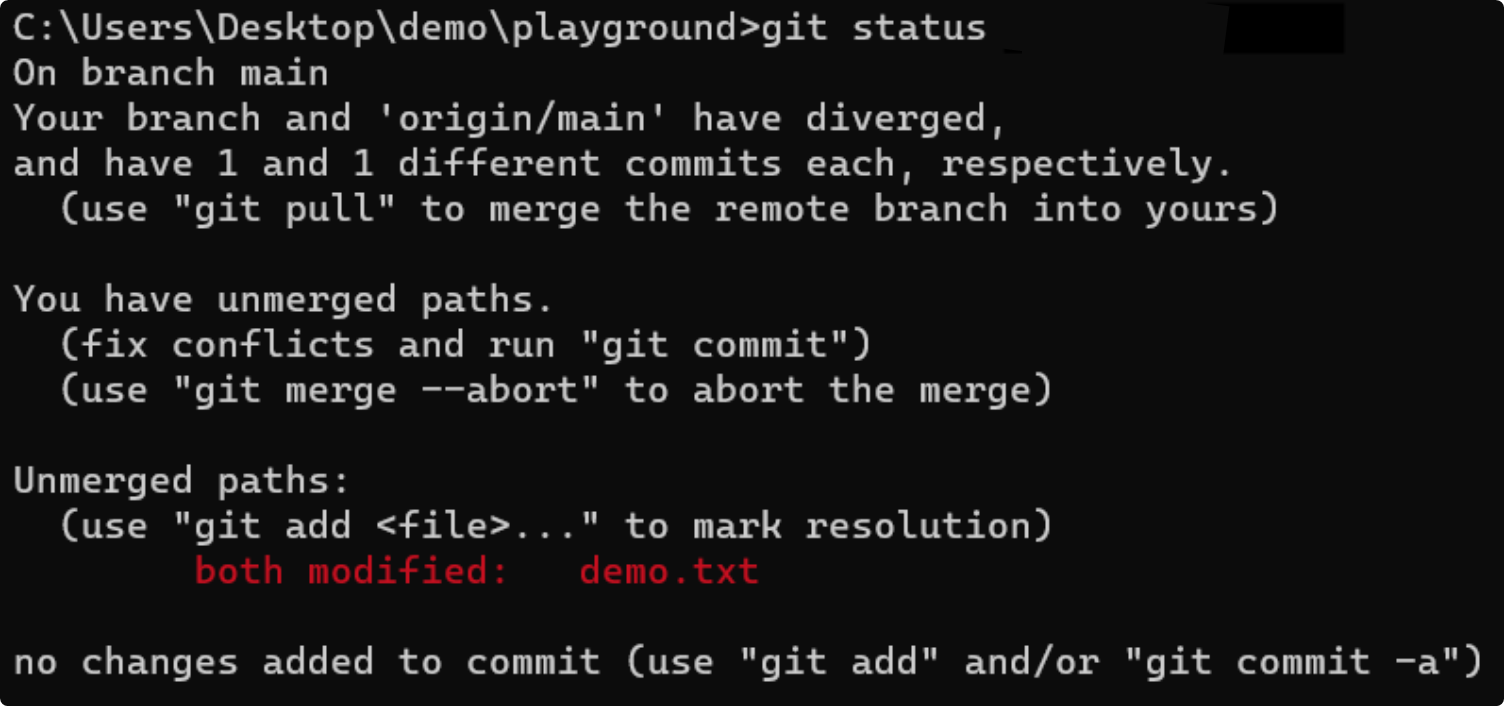
To **resolve** a **merge** **conflict** caused by **conflicting** **line** **changes**, we must choose which **changes** to **incorporate** in a new commit.

There a several steps that we should follow:

1. Open **Git Bash**
2. Navigate into the **local** **Git** **repository** that has the **merge** **conflict**

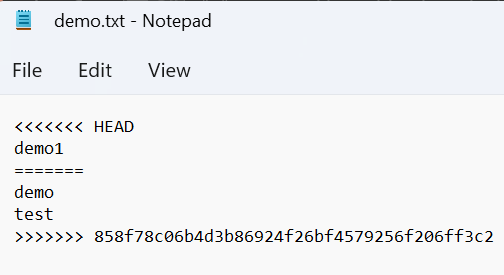


1. **Display** a **list** of the **files** **affected** by the **merge** **conflict**

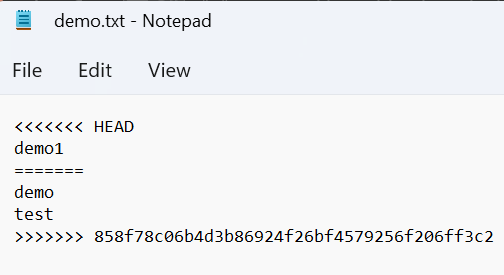
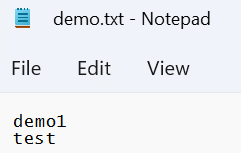


In the example above, the **demo.txt** file has a merge conflict.

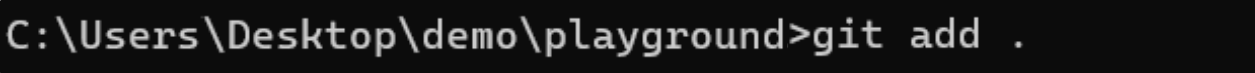
1. **Open** a **text** **editor** and **navigate** to the **file** with **merge** **conflicts**
2. To see the **beginning** of the **merge** **conflict** in the file, search the file for the conflict marker **<<<<<<<**
   * You'll see the changes from the **HEAD** after the line **<<<<<<<** **HEAD**
   * Next, you'll see **=======**, which divides your changes from the changes in the other branch,   
     followed by **>>>>>>> name**



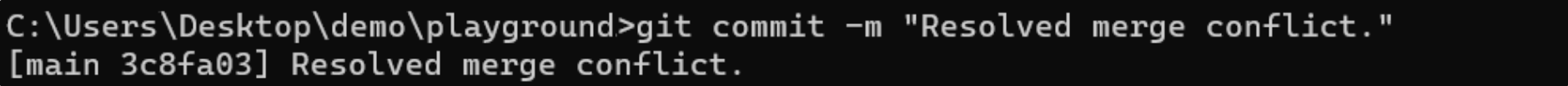
1. Decide if you want to keep **only** **your** **changes**, keep only the **other** **changes**, or make a **new** **change**, which incorporates **both** **changes.**
2. **Delete** the **conflict** **markers** **<<<<<<<**, **=======**, **>>>>>>>** and make the **changes** you want in the **final** **merge**

 **→** 

1. Add or stage the changes



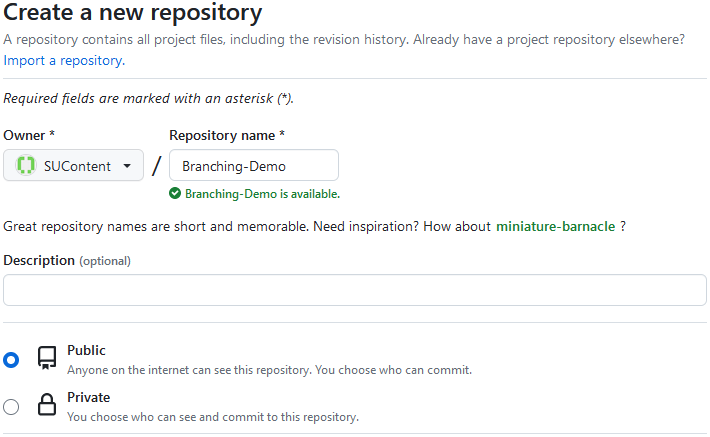
1. Finally, commit the changes with a comment



## Git Branches

### Step 1: Create and Clone Repo

Create an **empty GitHub repo and then clone the repo to work in it locally:**

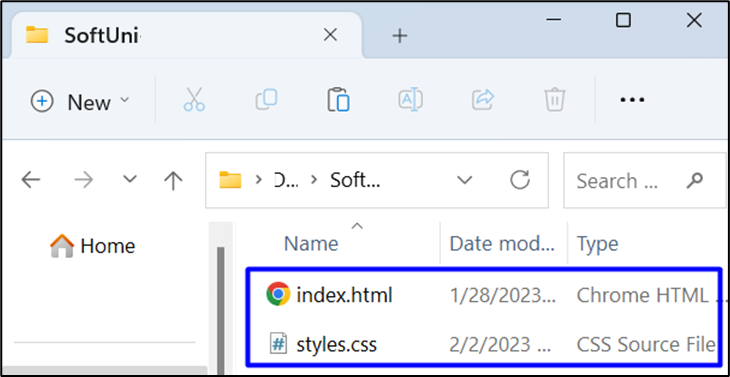


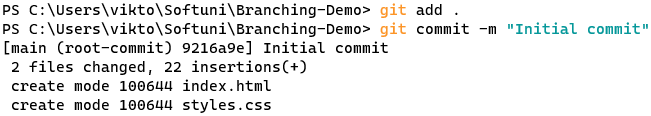
****

### Step 2: Add and Commit Files

After that, **add** the files from the **lab resources** to the local repo folder. You can use **git status** to check the working directory state:

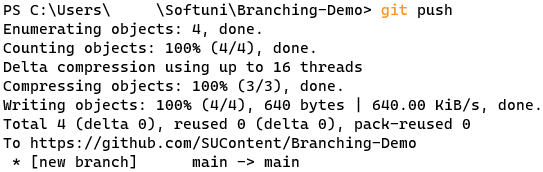


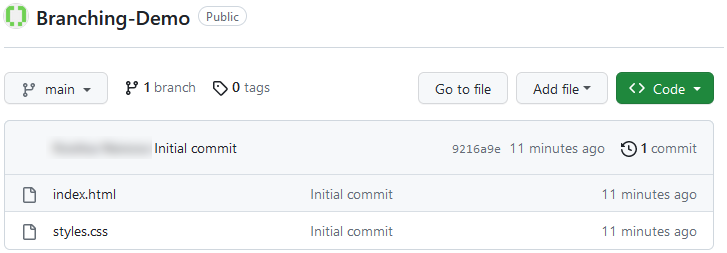


After that, **add** and **commit** all the chages with **git add .** and **git commit:**

### Step 3: Push to GitHub

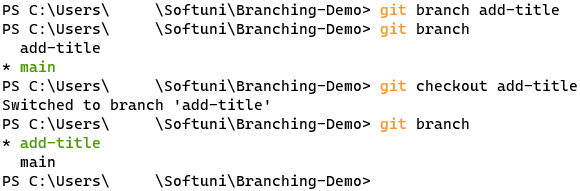
Next step is to **push** to the **remote** **repository**, using **git push**:





### Step 4: Create a New Branch

Now it's time to **create** and **switch** to a **new branch,** called **"add-title":**

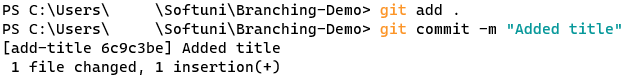


Make some changes in the **index.html** file (you can add an **<h1>** tag with title):

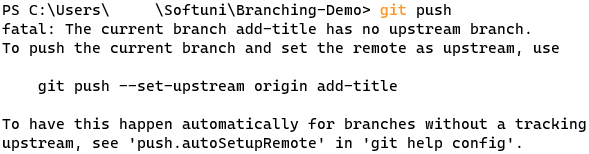


### Step 5: Commit New Branch Changes

Add and **commit** to the local repo:

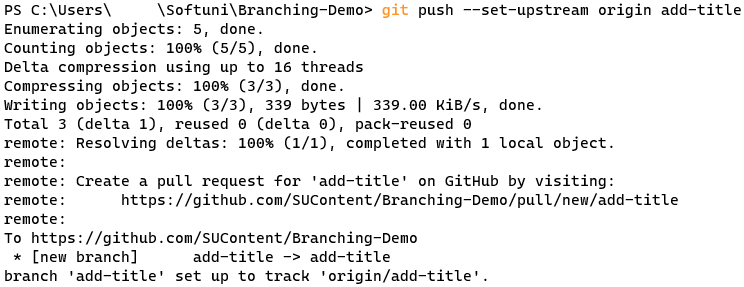


And then, **push** to the **remote** **repo**. An **error** should occur, as this branch is created only **locally** and you don't have it in your **remote GitHub repo**:

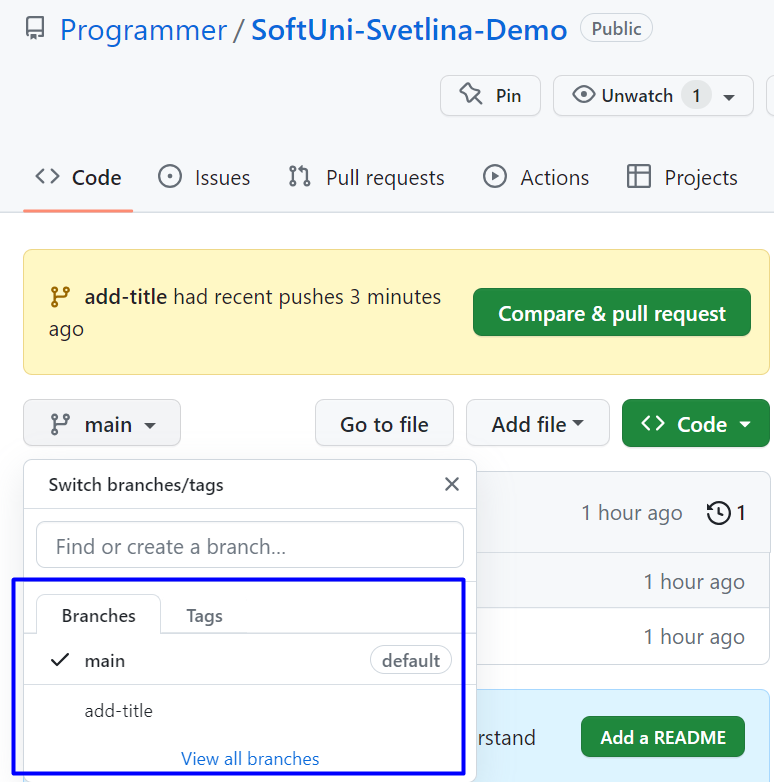


### Step 6: Add Upstream and Push Changes

Now, **add upstream** and **push again** using this command:

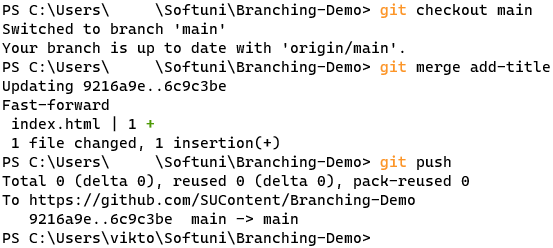


You should have your new "**add-title**" branch in the remote repo:



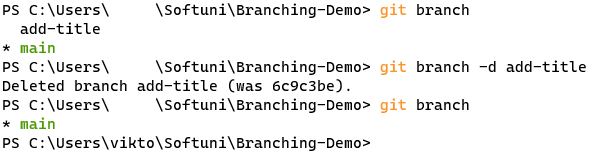
### Step 7: Merge Branches

Now, **switch** to the "**main**" branch and **merge** it with "**add-title**"



### Step 8: Delete Branch

**Delete the local branch:**



And **delete** the **remote** **GitHub branch**:

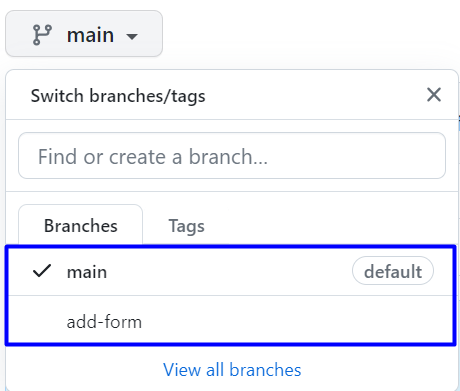


## Creating a Pull Request

### Step 1: Create Branch, Make Changes and Push

Like in the previous task, create a **new** **branch** "**add-form**":



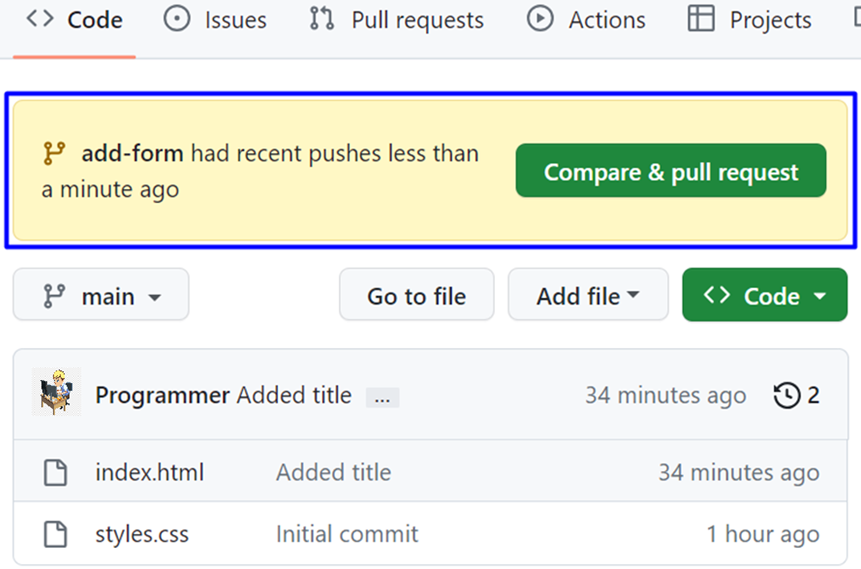


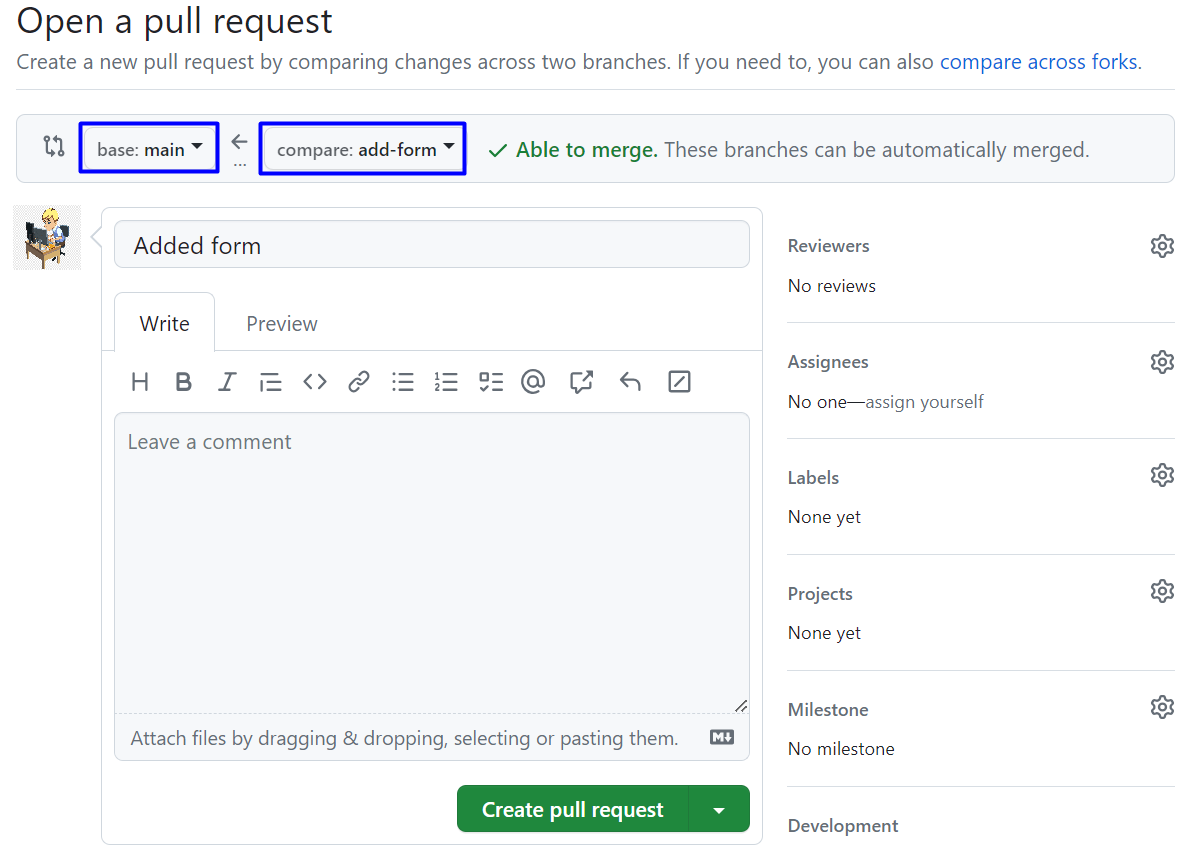
Now, **add** an HTML form in the **index.html** file, commit and push the changes to the **remote** **GitHub repo:**



### Step 2: Open a Pull Request in GitHub

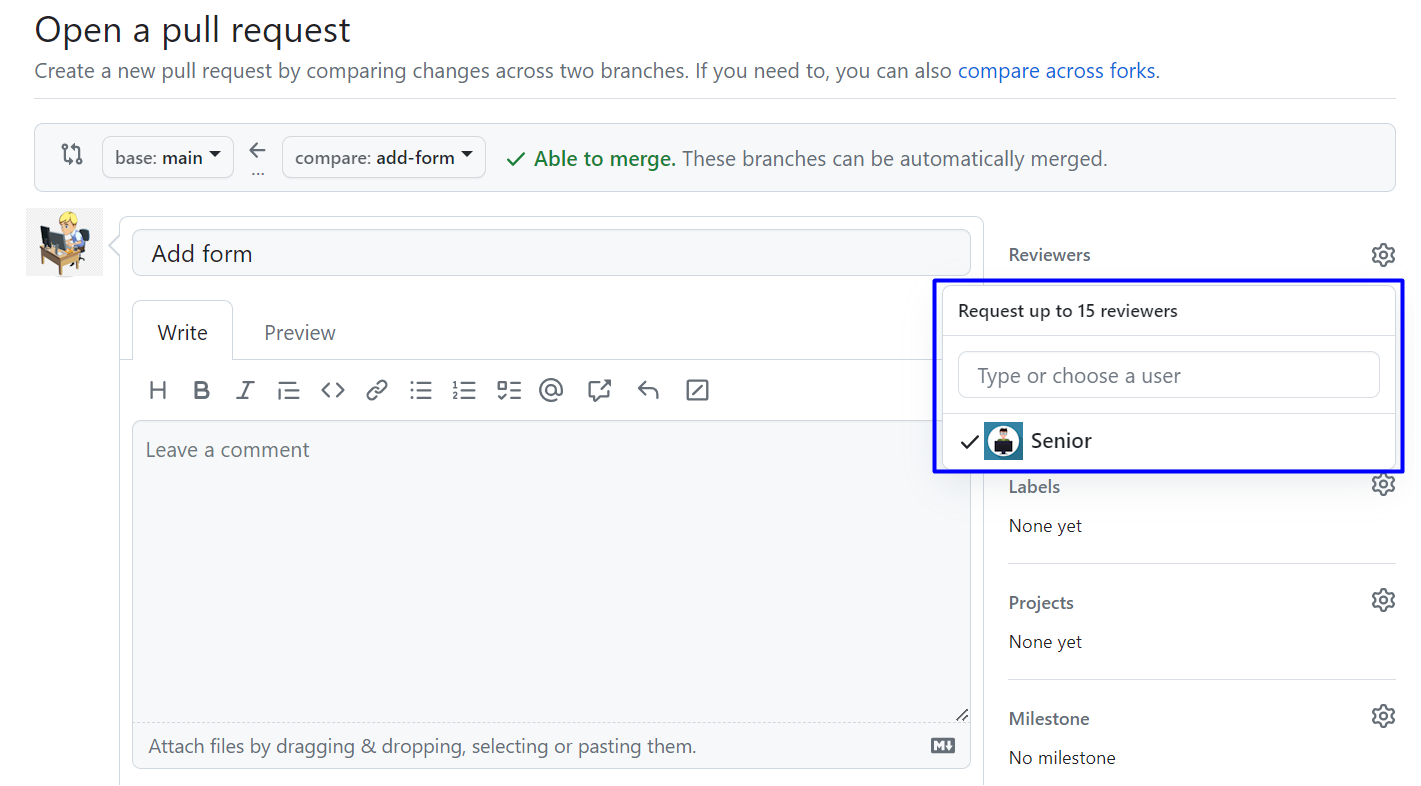
Now it's time to open a pull request from the "main" to the "add-form" branch:





### Step 3: Request a Review (Optional)

You have the option to request a review of your changes:



### Step 4: Team Discussion

