Lab: Git Branching and Pull Requests

Lab for the "Software Engineering and DevOps" course @ SoftUni.

1. 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:

```
PS C:\Users\Desktop\demo> git clone https://github.com/SUContent/playground
Cloning into 'playground'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
```

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:

```
PS C:\Users\Desktop\demo\playground> git add .
```

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

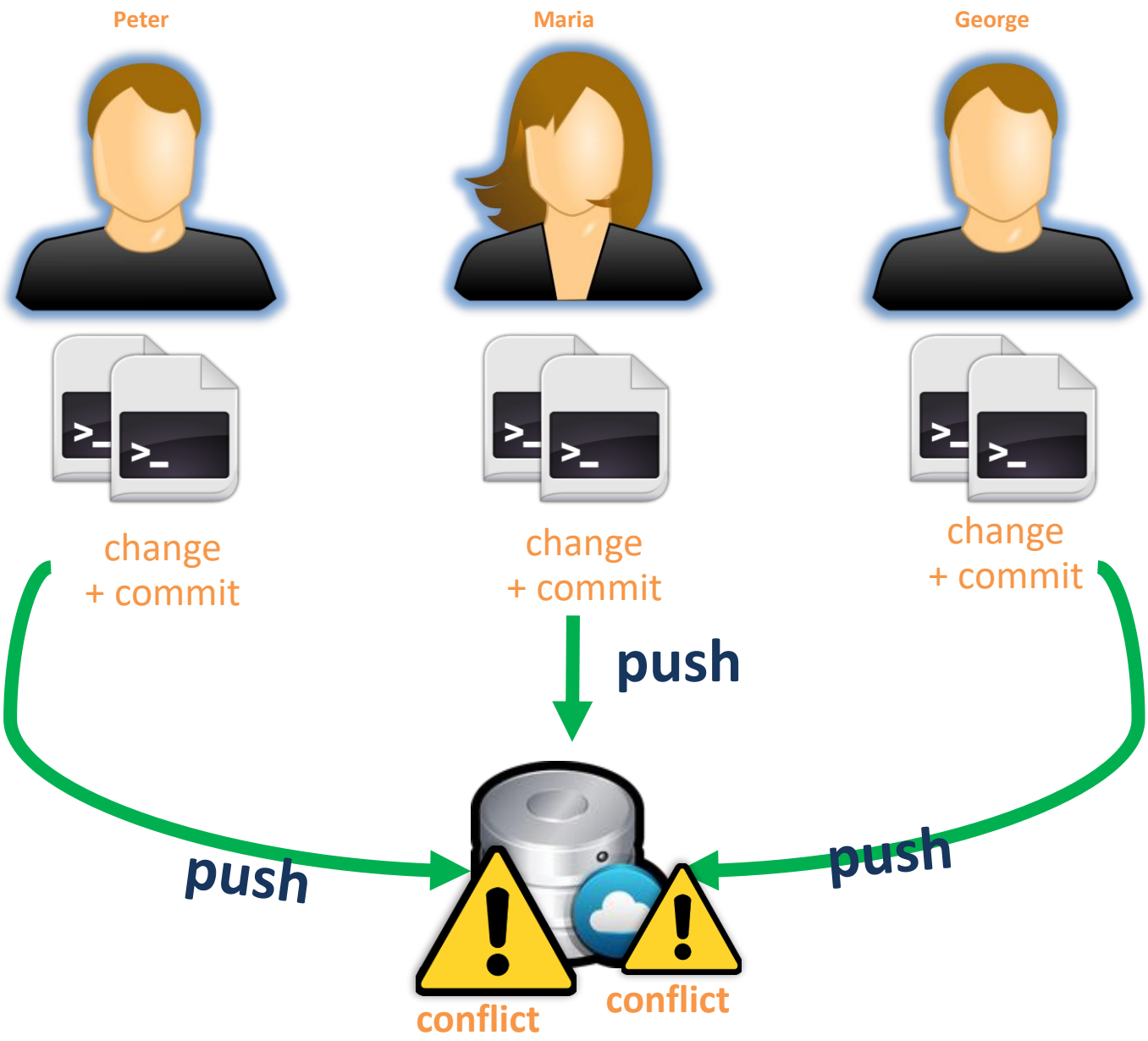
```
PS C:\Users\Desktop\demo\playground> git commit -m "changes"
[main 33630ac] changes
 1 file changed, 1 insertion(+)
 create mode 100644 demo.txt
```

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

```
PS C:\Users\Desktop\demo\playground> git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 312 bytes | 312.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/SoftUni/playground
   9b3bd01..33630ac  main -> main
```

2. 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**.



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**

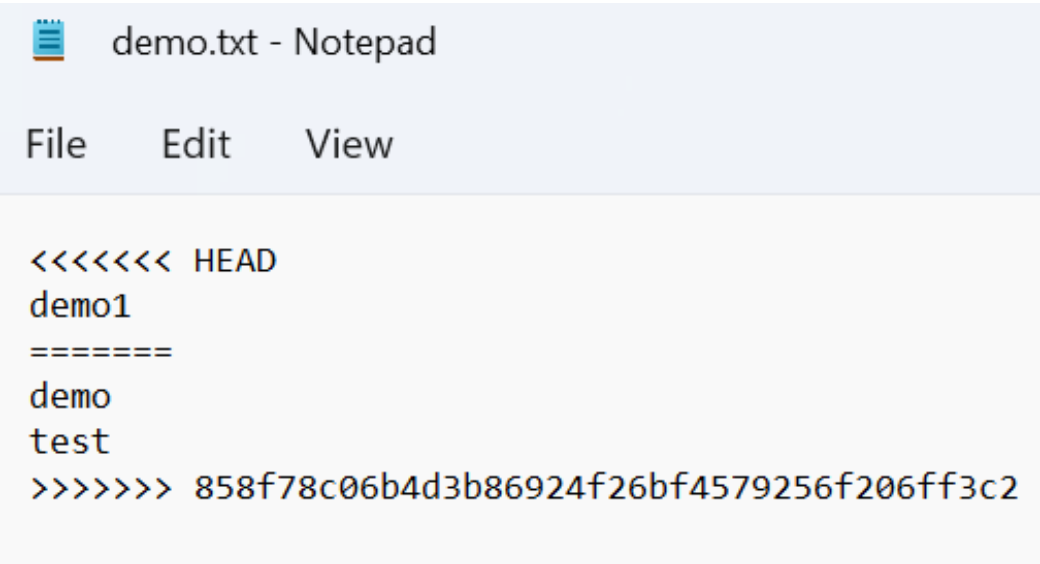
```
C:\Users\Desktop\demo>cd playground
```

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

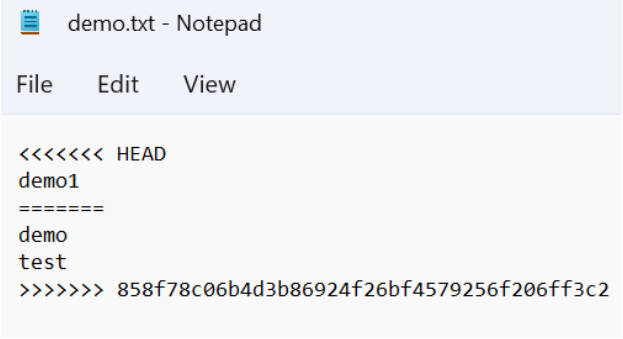
```
C:\Users\Desktop\demo\playground>git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   demo.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

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

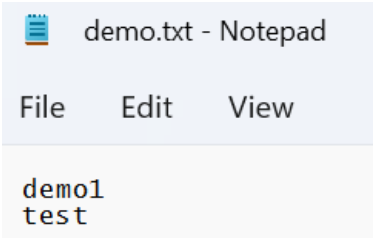
4. **Open** a **text editor** and **navigate** to the **file** with **merge conflicts**
5. To see the **beginning** of the **merge conflict** in the file, search the file for the conflict marker **`<<<<<<<`**
   - You'll see the changes from the **HEAD** after the line **`<<<<<<< HEAD`**
   - Next, you'll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



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



→



8. Add or stage the changes

```
C:\Users\Desktop\demo\playground>git add .
```

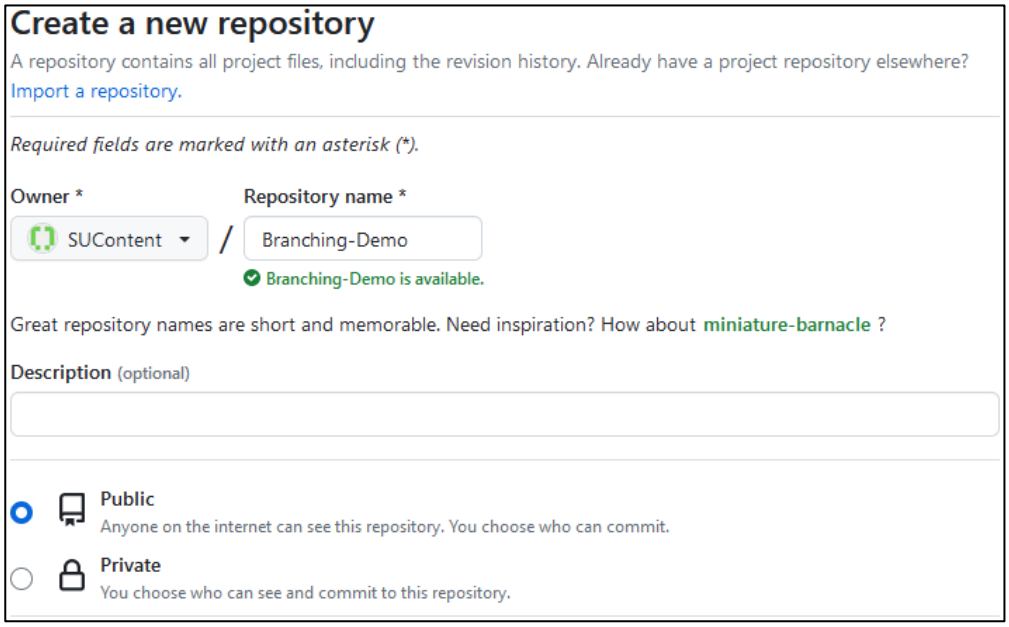
9. Finally, commit the changes with a comment

```
C:\Users\Desktop\demo\playground>git commit -m "Resolved merge conflict."
[main 3c8fa03] Resolved merge conflict.
```

3. Git Branches

Step 1: Create and Clone Repo

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

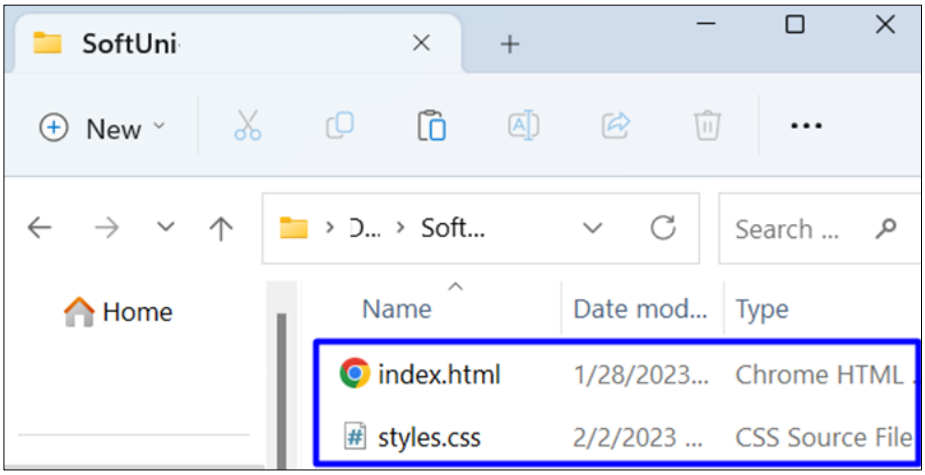



```
PS C:\Users\     \Softuni> git clone https://github.com/SUContent/Branching-Demo
Cloning into 'Branching-Demo'...
warning: You appear to have cloned an empty repository.
```

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:

```
PS C:\Users\     \Softuni\Branching-Demo> git status
On branch main

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```



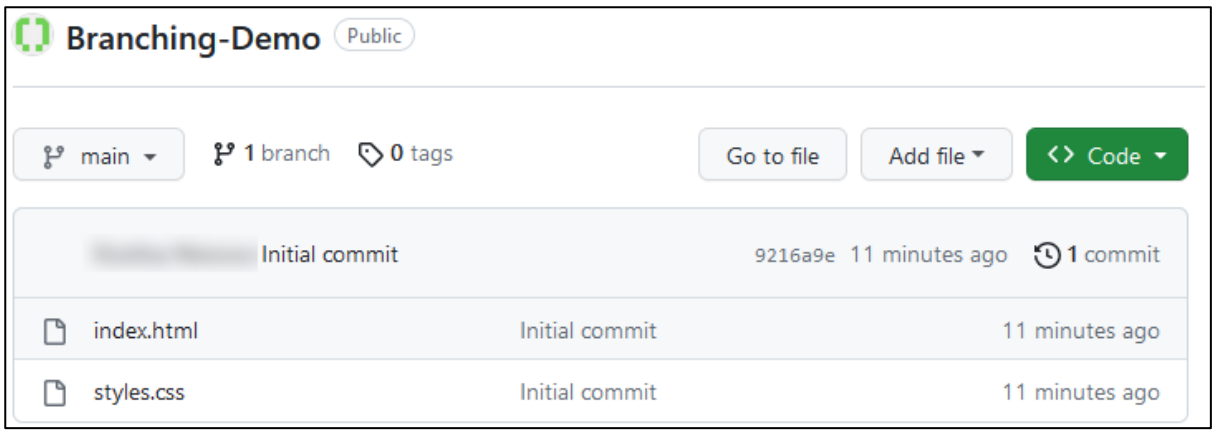
After that, **add** and **commit** all the chages with **`git add .`** and **`git commit`**:

```
PS C:\Users\vikto\Softuni\Branching-Demo> git add .
PS C:\Users\vikto\Softuni\Branching-Demo> git commit -m "Initial commit"
[main (root-commit) 9216a9e] Initial commit
 2 files changed, 22 insertions(+)
 create mode 100644 index.html
 create mode 100644 styles.css
```

Step 3: Push to GitHub

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

```
PS C:\Users\     \Softuni\Branching-Demo> git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 640 bytes | 640.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/SUContent/Branching-Demo
 * [new branch]      main -> main
```



Step 4: Create a New Branch

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

```
PS C:\Users\      \Softuni\Branching-Demo> git branch add-title
PS C:\Users\      \Softuni\Branching-Demo> git branch
  add-title
* main
PS C:\Users\      \Softuni\Branching-Demo> git checkout add-title
Switched to branch 'add-title'
PS C:\Users\      \Softuni\Branching-Demo> git branch
* add-title
  main
PS C:\Users\      \Softuni\Branching-Demo>
```

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

```
index.html M
index.html > ...
1   <!DOCTYPE html>
2   <html lang="en">
3
4   <head>
5       <meta charset="UTF-8">
6       <meta http-equiv="X-UA-Compatible" cont
7       <meta name="viewport" content="width=de
8       <title>Hello World</title>
9   </head>
10
11  <body>
12      <div>
13          <h1>Hello World</h1>
14          <p>Lorem ipsum dolor sit amet conse
15              doloribus suscipit reprehenderi
16              autem. Commodi, ullam?</p>
17      </div>
18  </body>
19
20  </html>
```

Step 5: Commit New Branch Changes

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

```
PS C:\Users\      \Softuni\Branching-Demo> git add .
PS C:\Users\      \Softuni\Branching-Demo> git commit -m "Added title"
[add-title 6c9c3be] Added title
 1 file changed, 1 insertion(+)
```

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**:

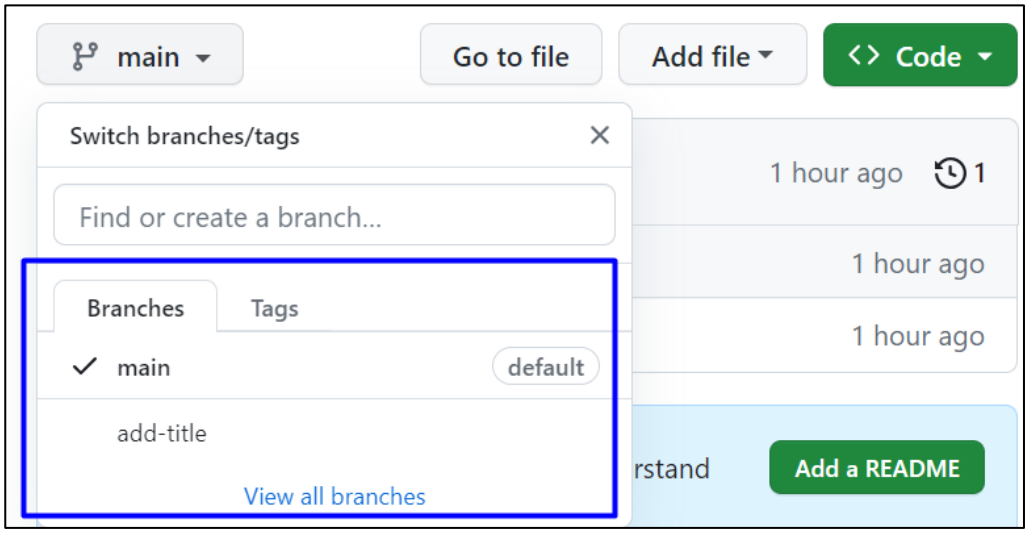
```
PS C:\Users\     \Softuni\Branching-Demo> git push
fatal: The current branch add-title has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin add-title

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.
```

Step 6: Add Upstream and Push Changes

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

```
PS C:\Users\     \Softuni\Branching-Demo> git push --set-upstream origin add-title
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 339 bytes | 339.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'add-title' on GitHub by visiting:
remote:      https://github.com/SUContent/Branching-Demo/pull/new/add-title
remote:
To https://github.com/SUContent/Branching-Demo
 * [new branch]      add-title -> add-title
branch 'add-title' set up to track 'origin/add-title'.
```

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`**"

```
PS C:\Users\     \Softuni\Branching-Demo> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\     \Softuni\Branching-Demo> git merge add-title
Updating 9216a9e..6c9c3be
Fast-forward
 index.html | 1 +
 1 file changed, 1 insertion(+)
PS C:\Users\     \Softuni\Branching-Demo> git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/SUContent/Branching-Demo
   9216a9e..6c9c3be  main -> main
PS C:\Users\vikto\Softuni\Branching-Demo>
```

Step 8: Delete Branch

Delete the local branch:

```
PS C:\Users\     \Softuni\Branching-Demo> git branch
  add-title
* main
PS C:\Users\     \Softuni\Branching-Demo> git branch -d add-title
Deleted branch add-title (was 6c9c3be).
PS C:\Users\     \Softuni\Branching-Demo> git branch
* main
PS C:\Users\vikto\Softuni\Branching-Demo>
```

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

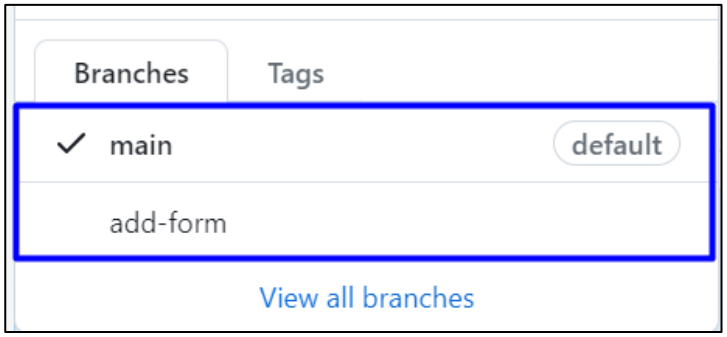
```
PS C:\Users\     \Softuni\Branching-Demo> git push origin -d add-title
To https://github.com/SUContent/Branching-Demo
 - [deleted]         add-title
```

4. Creating a Pull Request

Step 1: Create Branch, Make Changes and Push

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

```
PS C:\Users\     \Softuni\Branching-Demo> git branch add-form
PS C:\Users\     \Softuni\Branching-Demo> git checkout add-form
Switched to branch 'add-form'
```

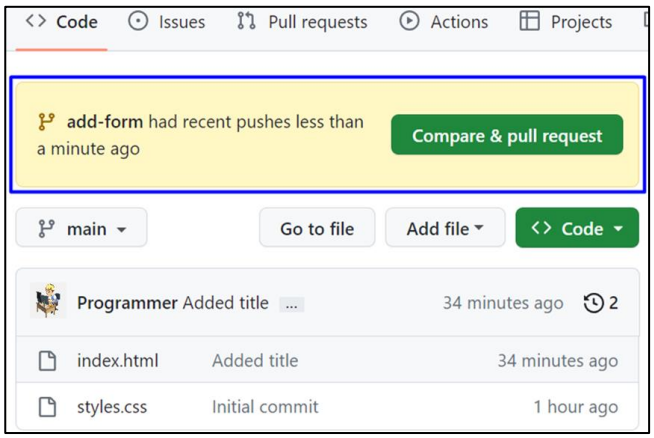


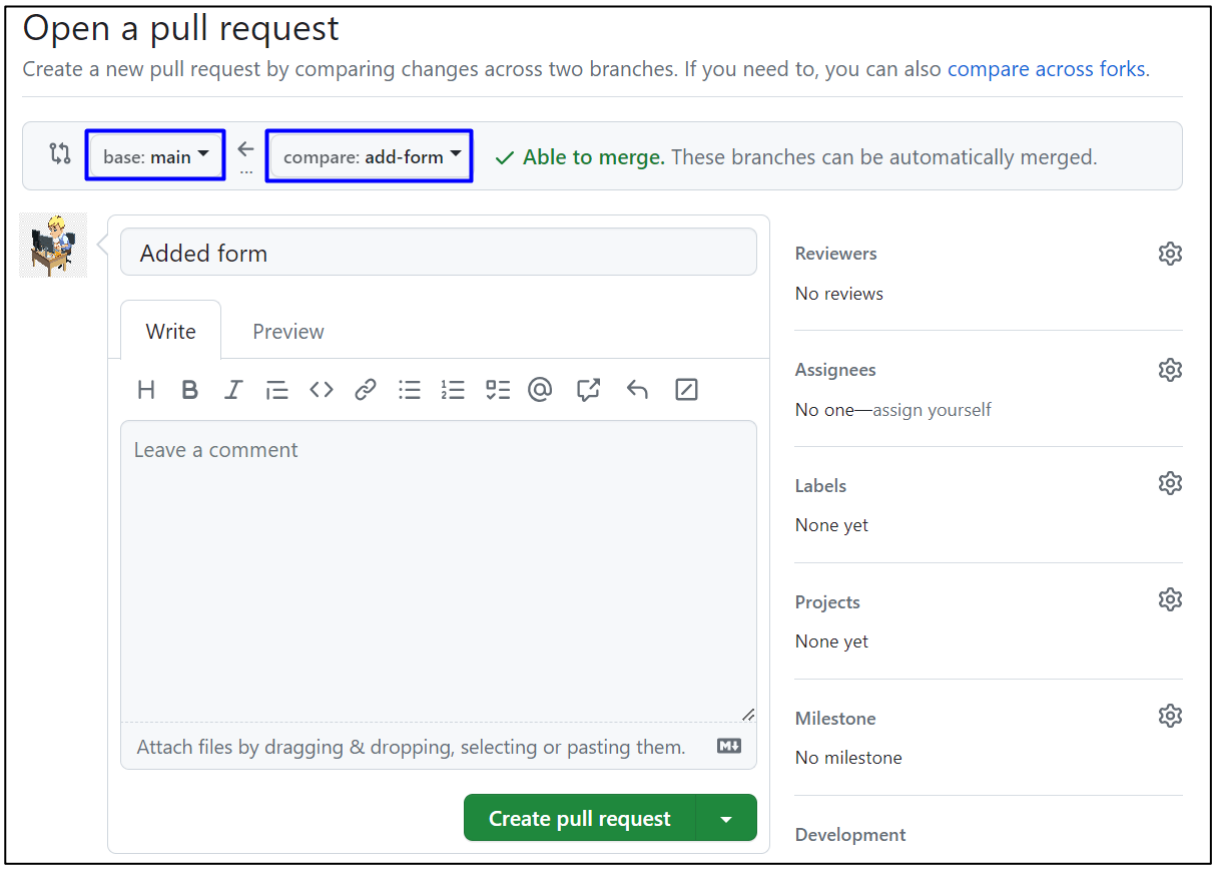
Now, **add** an HTML form in the **`index.html`** file, commit and push the changes to the **remote GitHub repo:**

```html
<!DOCTYPE html>
<html lang="en">

<head> ···
</head>

<body>
    <div> ···
    </div>
    <form action="/action_page.php">
        <label for="fname">First name:</label><br>
        <input type="text" id="fname" name="fname" value="John"><br>
        <label for="lname">Last name:</label><br>
        <input type="text" id="lname" name="lname" value="Doe"><br><br>
        <input type="submit" value="Submit">
    </form>
</body>
```

Step 2: Open a Pull Request in GitHub

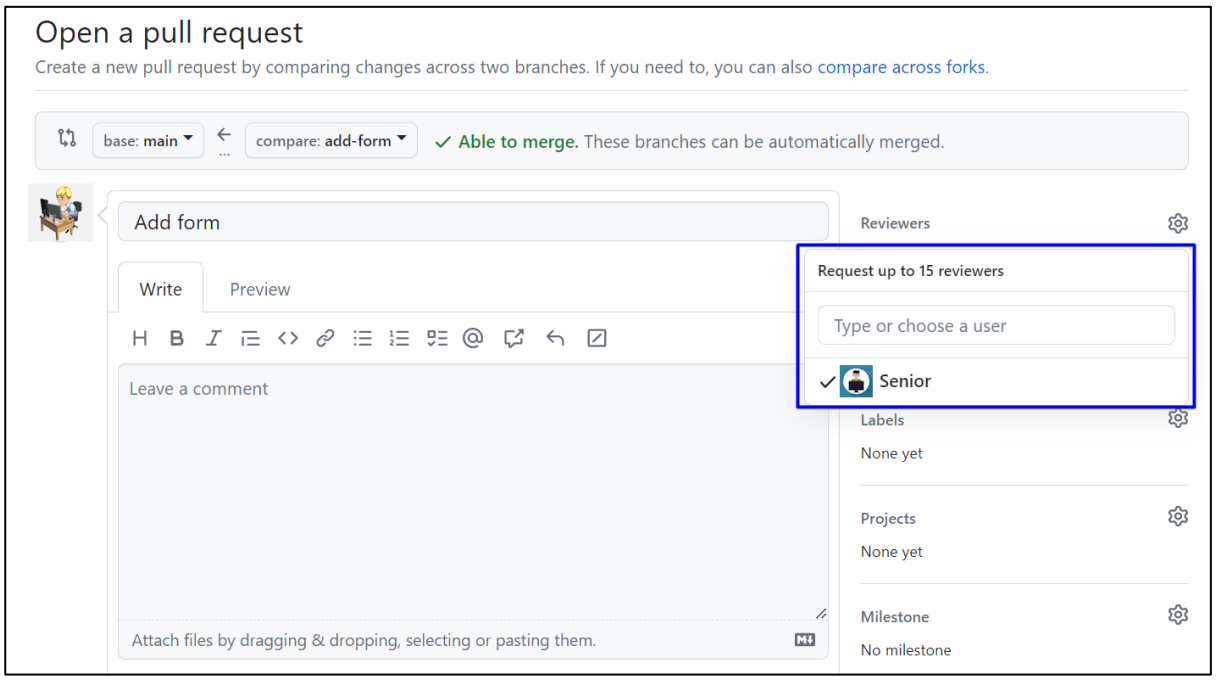
Now it's time to open a pull request from the "main" to the "add-form" branch:





Step 3: Request a Review (Optional)

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



Step 4: Team Discussion