

Using Visual Studio Code with GitHub in a team

Work as a pair, with one student as StudentA and another as StudentX

Steps for StudentA and StudentX are indicated as **A** and **X** respectively

While one student is doing the steps, the other student should see what is being done

If there are an odd number of students in the class, one group can have 3 students

The third student will do steps similar to Student X after all the steps are completed

Overview of steps that will be done

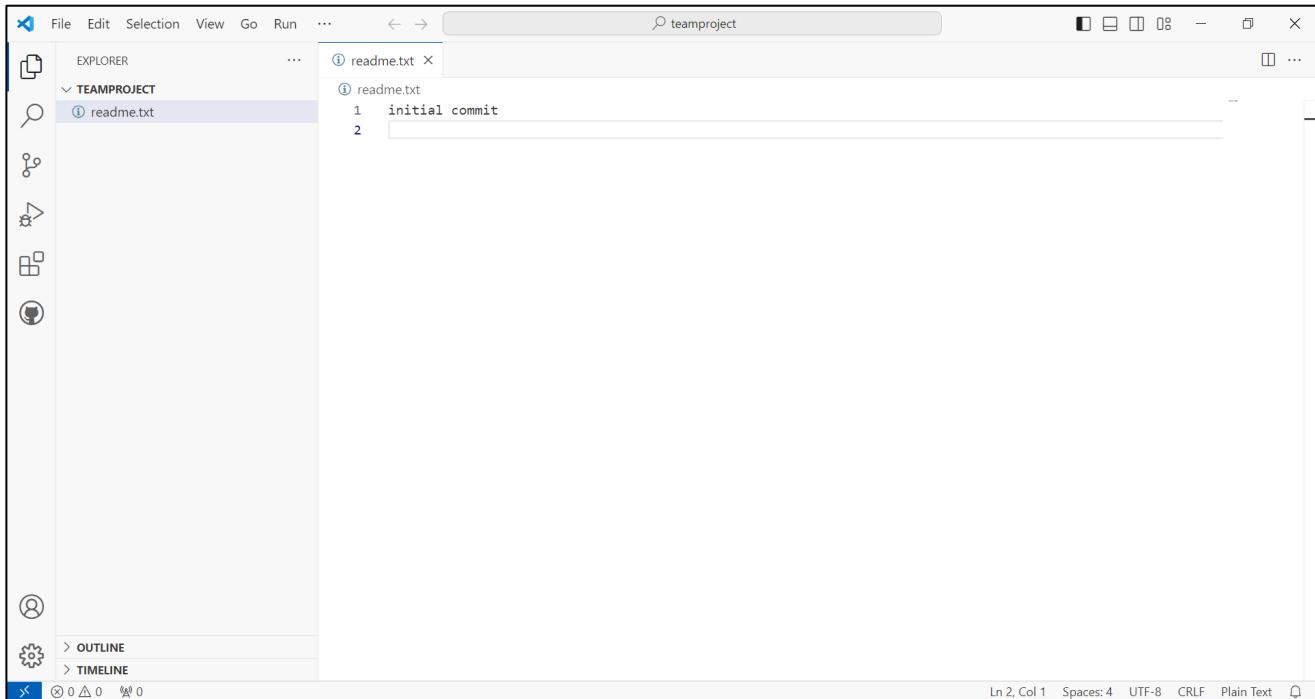
Sequence	Student A	Student X
1	Create a folder for the project	
2	Create a local repository	
3	Stage changes	
4	Commit changes	
5	Publish branch	
6	View repository on GitHub	
7	Invite collaborators	
8		Accept invite
9		Get the project web URL
10		Clone the repository
11		Create a branch
12		Publish the branch
13		Make changes, stage changes, commit changes, sync changes
14		Checkout main
15		Merge branch
16	Create a branch	
17	Publish the branch	
18	Make changes, stage changes, commit changes, sync changes	
19	Checkout main	
20	Merge branch	
21	Resolve conflicts	
22	Sync changes	

Create a folder for the project

- A 1. Create a folder **teamproject**

Open the folder in Visual Studio Code

Add a text file called **readme.txt** with the text **initial commit**

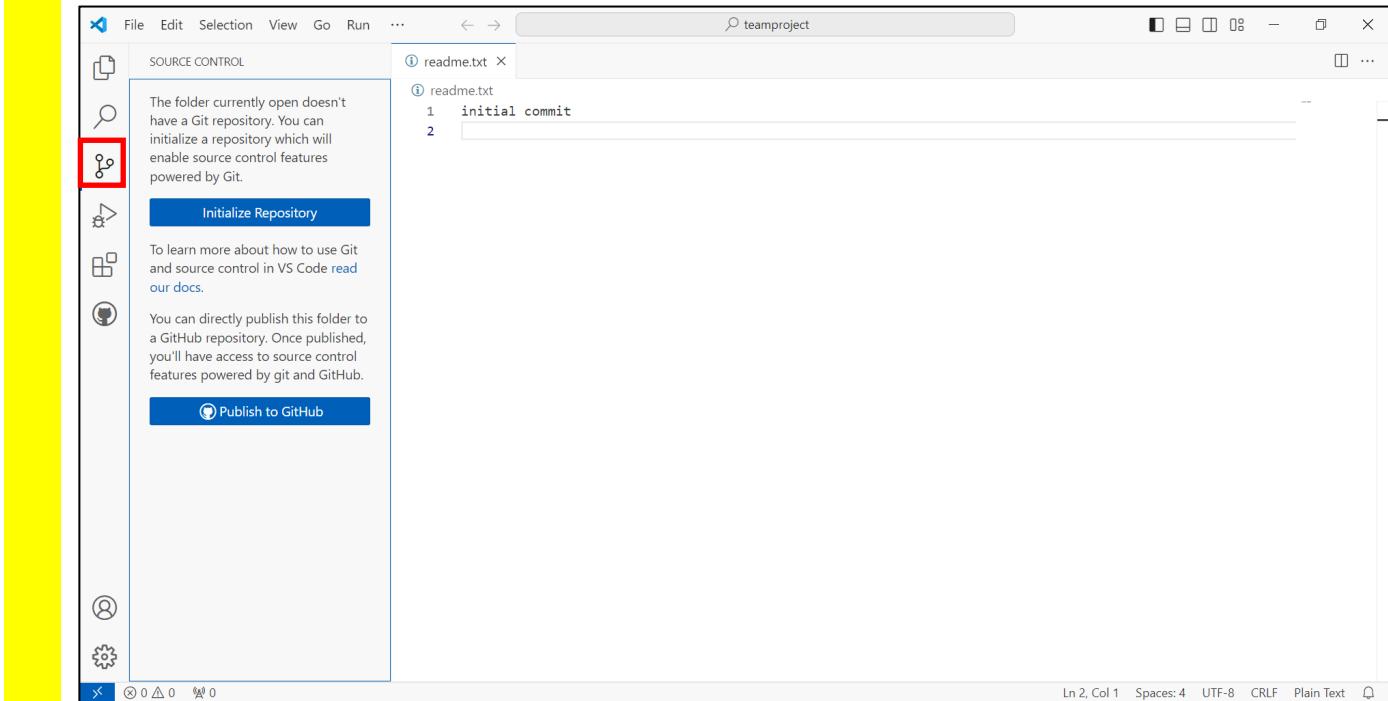


At this point, the project only exists on StudentA's computer

As per normal, you can add folders and source files to your project. For this example, we will work on a single text file, but the same principles apply to other files in your projects.

Create a local repository

A 2. Click the Source Control button and Initialize Repository

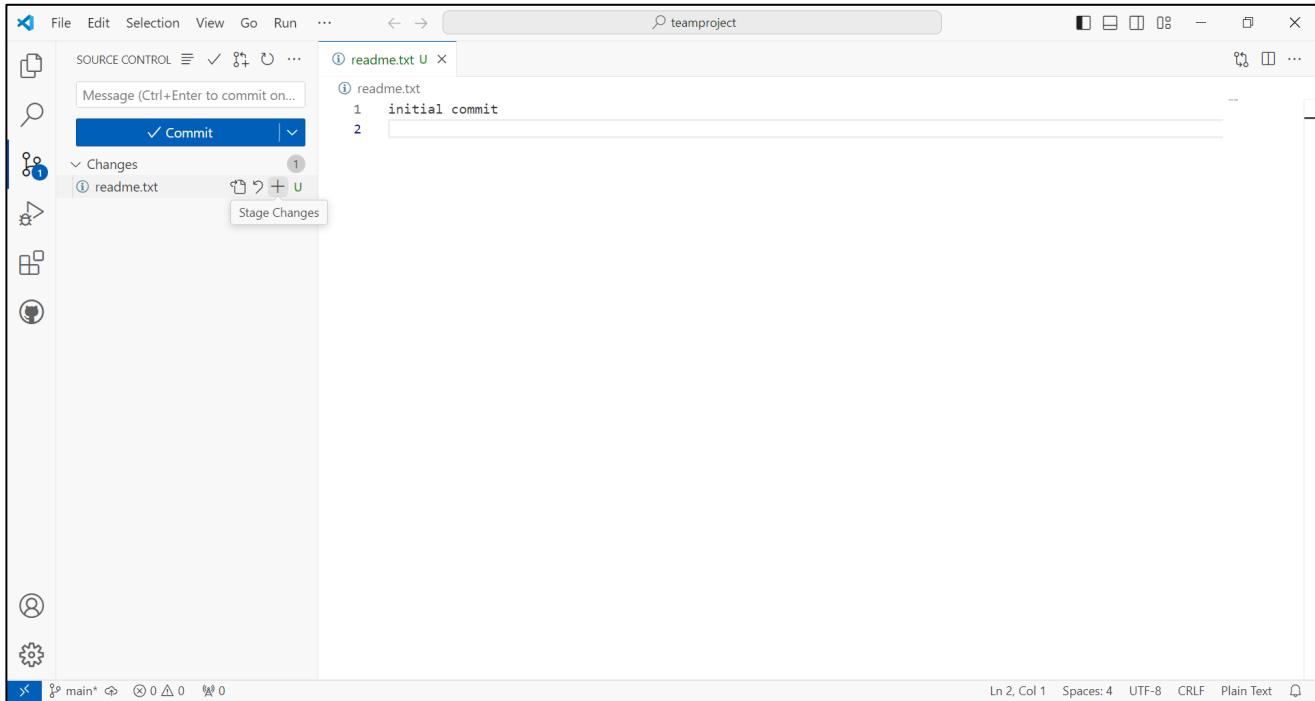


After creating a project, you will need to initialise a **local repository** at the root of your project folder. This will enable version control features for the project.

Stage changes

The bottom left corner of Visual Studio Code shows that you are working on the main branch of the local repository.

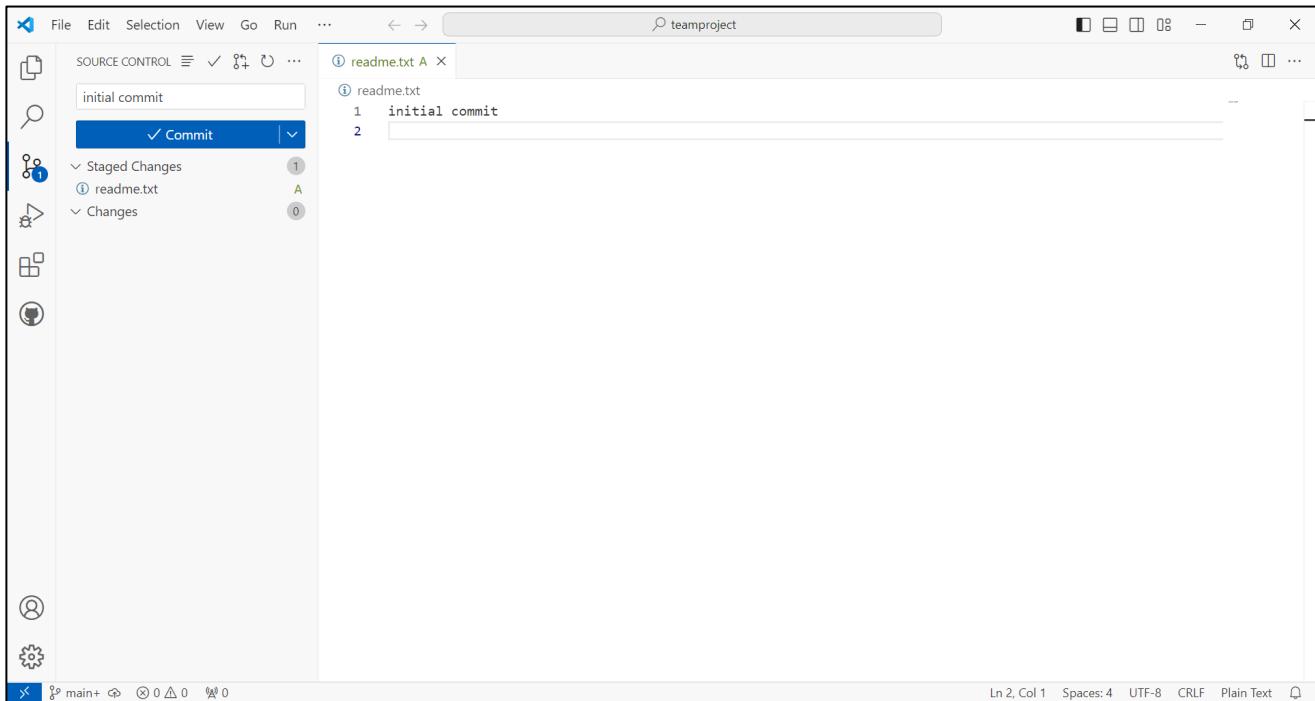
- A 3. Click on the + button next to `readme.txt` to stage changes for that file



After a file has been created, modified or deleted, you will need to stage it. This prepares the file for committing.

Commit changes

- A 4. Set the commit message as **initial commit** and press **Commit**

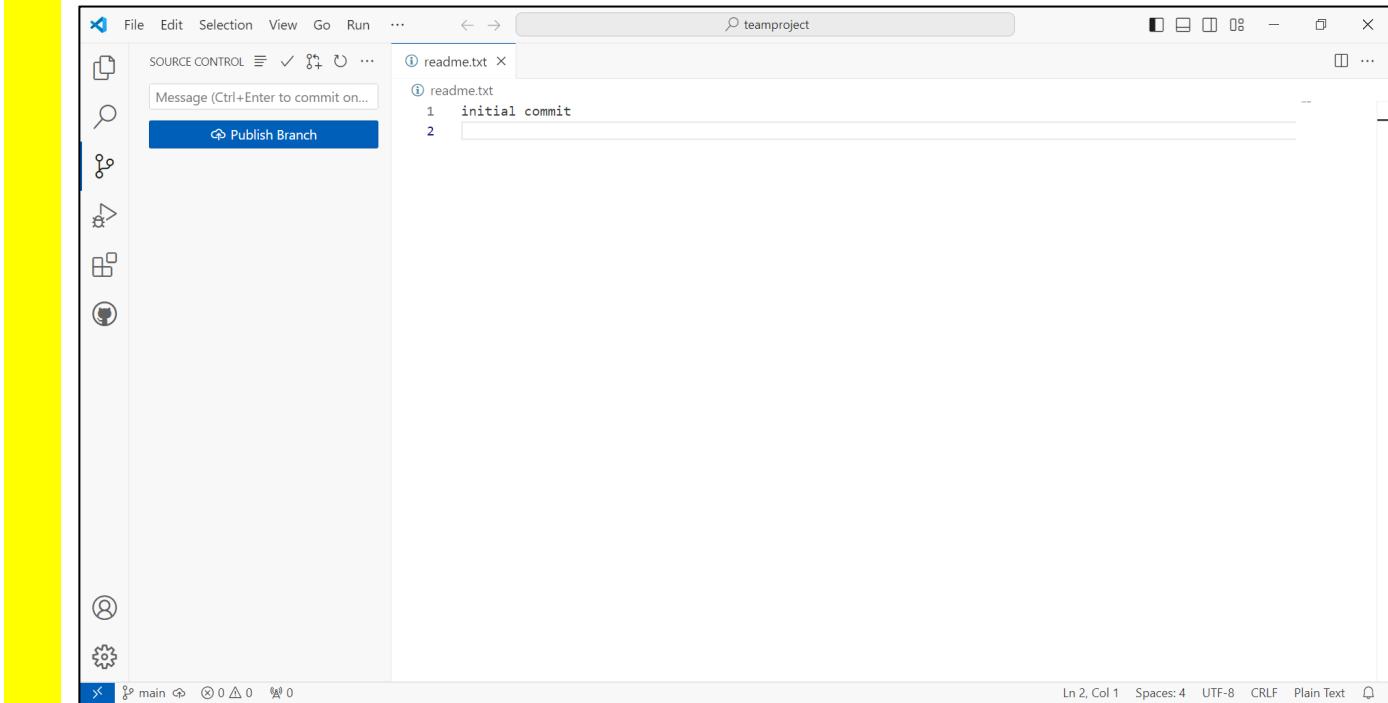


The commit message should be used to provide a meaningful explanation of what was committed.

For example, if you were working on a feature, the commit message should mention the feature that was developed.

Publish branch

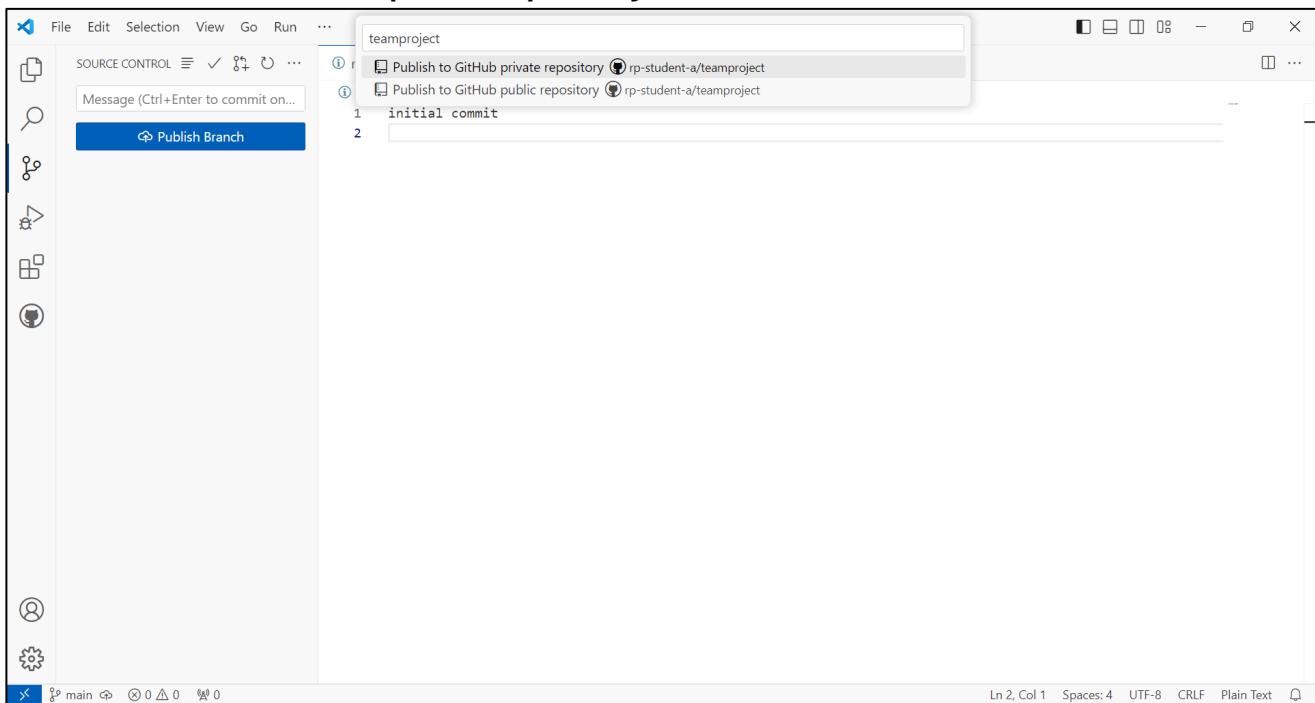
A 5. Click Publish Branch



At this point, the repository only exists on Student A's computer.

Publish branch will publish the project to the main branch of the **remote repository** on GitHub.

A 6. Click Publish to GitHub private repository

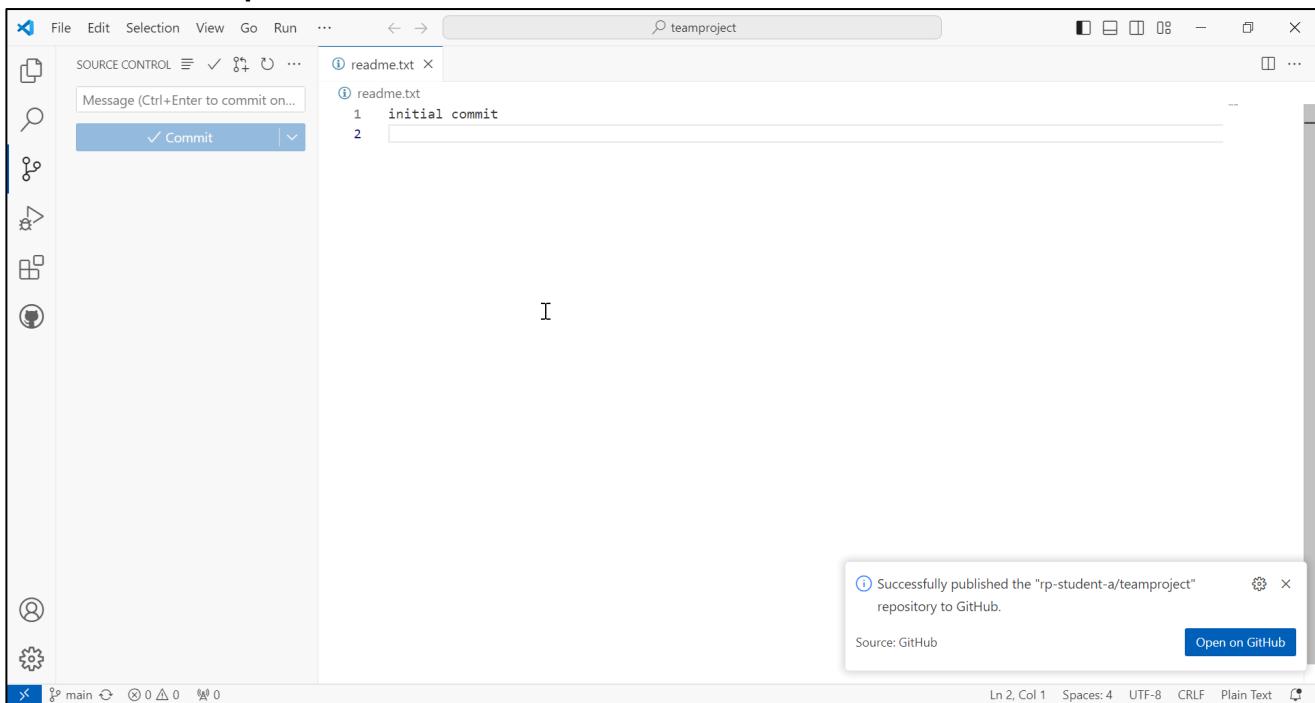


A repository can be **private** or **public**.

Public repositories can be viewed by anyone.

Private repositories can be viewed by collaborators.

A 7. A dialog should appear to indicate that the repository was successfully published to GitHub. Click on the Open on GitHub button to launch GitHub in a browser.

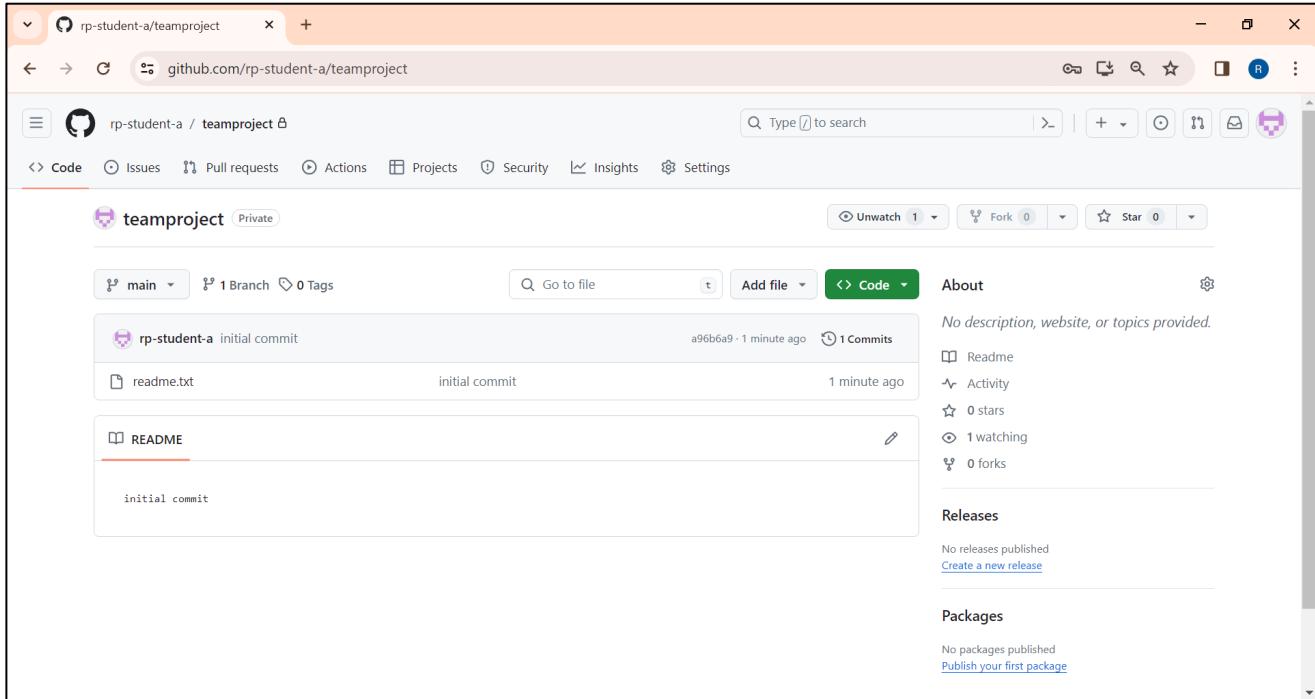


View repository on GitHub

- A 8. The repository has been created on GitHub

Click on **Settings**

For the next step, we will be adding collaborators, which can be done from the Settings page



The screenshot shows a GitHub repository page for 'teamproject'. The repository is private, as indicated by the 'Private' badge. It contains one branch ('main') and one commit ('initial commit' by 'rp-student-a'). The commit was made 1 minute ago and includes files 'readme.txt' and 'README'. On the right side, there is an 'About' section with a note: 'No description, website, or topics provided.' It also lists 'Readme', 'Activity', '0 stars', '1 watching', '0 forks', and sections for 'Releases' and 'Packages', both of which are currently empty.

The remote repository can act as a **backup** for your project.

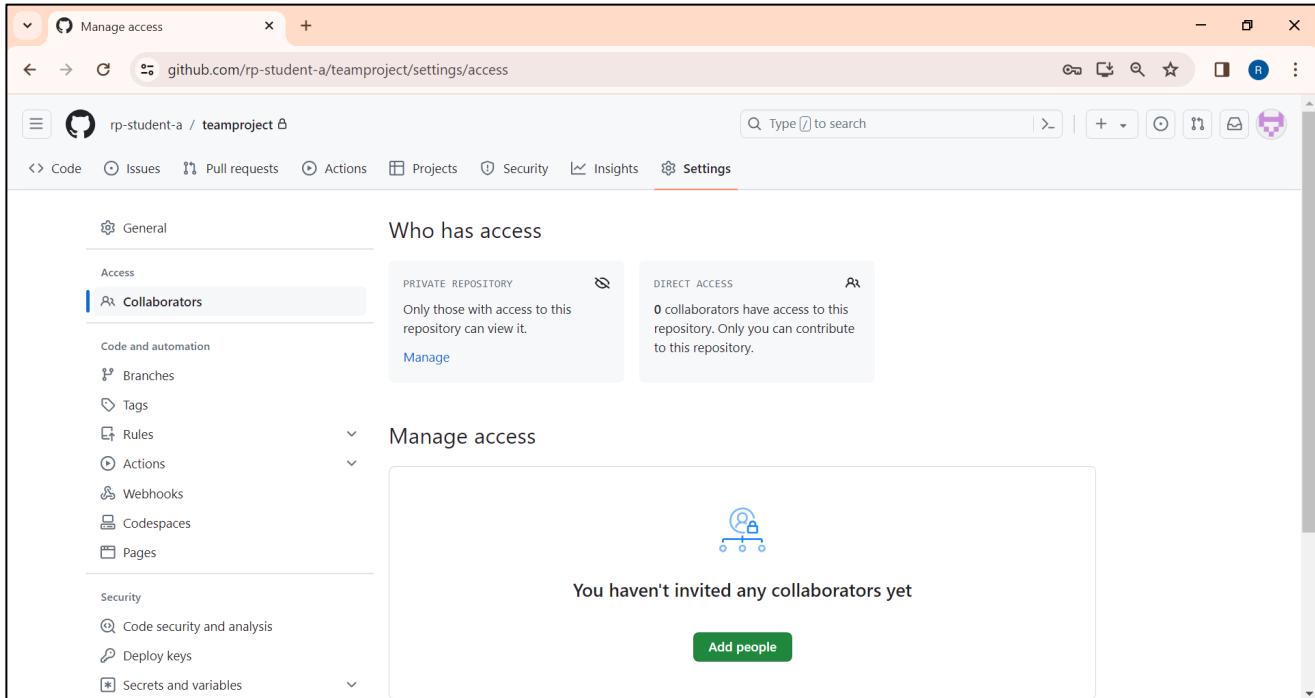
You can **retrieve** the project files that you have published by cloning the repository.

You should publish your projects (e.g. project assignments, final year project) to a **private repository**.

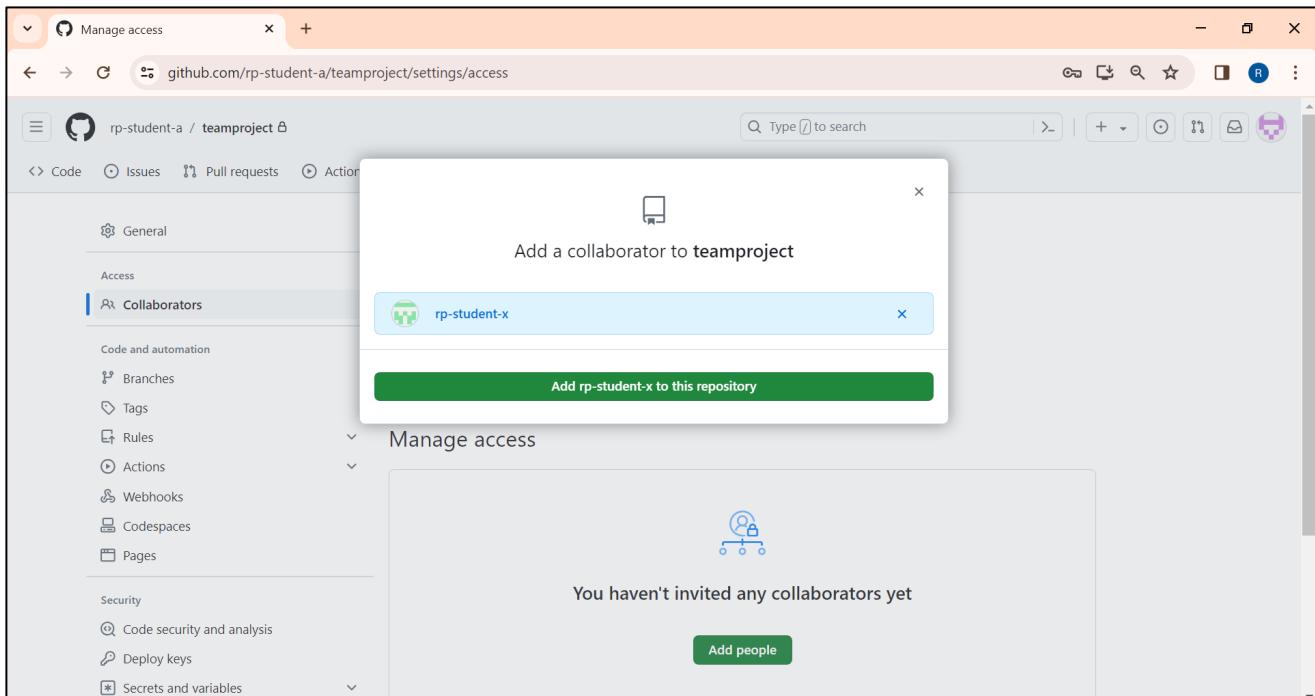
Do not publish to a public repository or else anyone will have access to your projects. It is **your responsibility** to ensure that your projects are not accessed by unauthorised people.

Invite collaborators

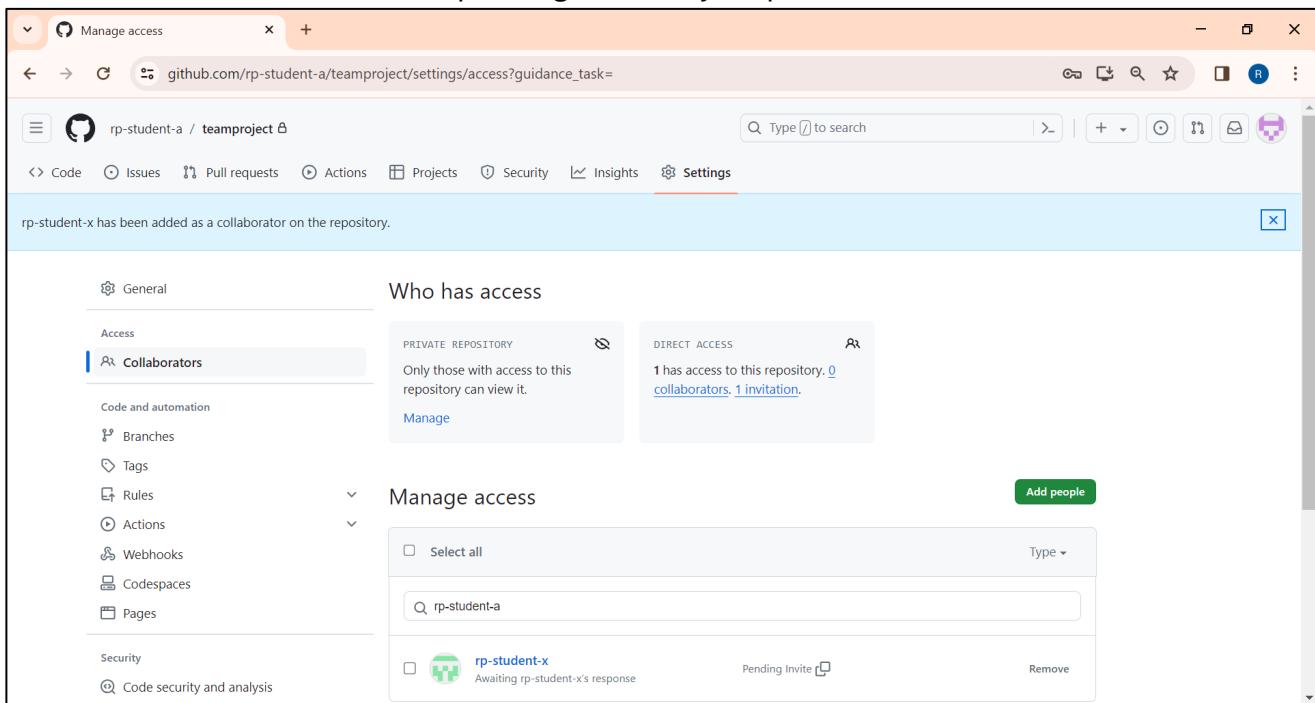
- A 9. On the **Settings** page, click **Collaborators** from the left panel
 Click on **Add people**



- A 10. Enter the GitHub id for your partner and click on the **Add to this repository** button



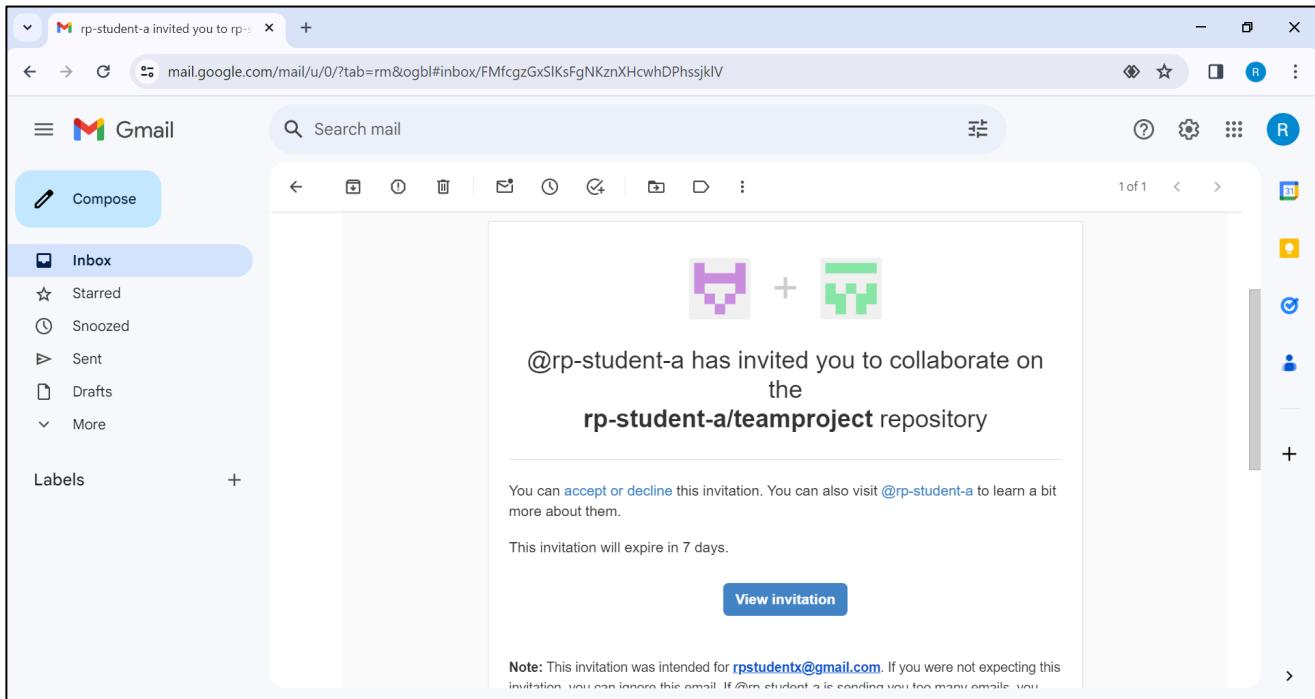
A 11. You should see that there is a pending invite for your partner



The screenshot shows the GitHub repository settings page for 'rp-student-a / teamproject'. The 'Access' section is selected, specifically the 'Collaborators' tab. On the right, under 'Who has access', it shows 'PRIVATE REPOSITORY' and 'DIRECT ACCESS'. Under 'DIRECT ACCESS', it says '1 has access to this repository: 0 collaborators, 1 invitation.' Below this, the 'Manage access' section lists 'rp-student-a' and 'rp-student-x'. 'rp-student-x' is shown with a status of 'Awaiting rp-student-x's response' and a 'Pending Invite' button. There is also a 'Remove' button next to the invite.

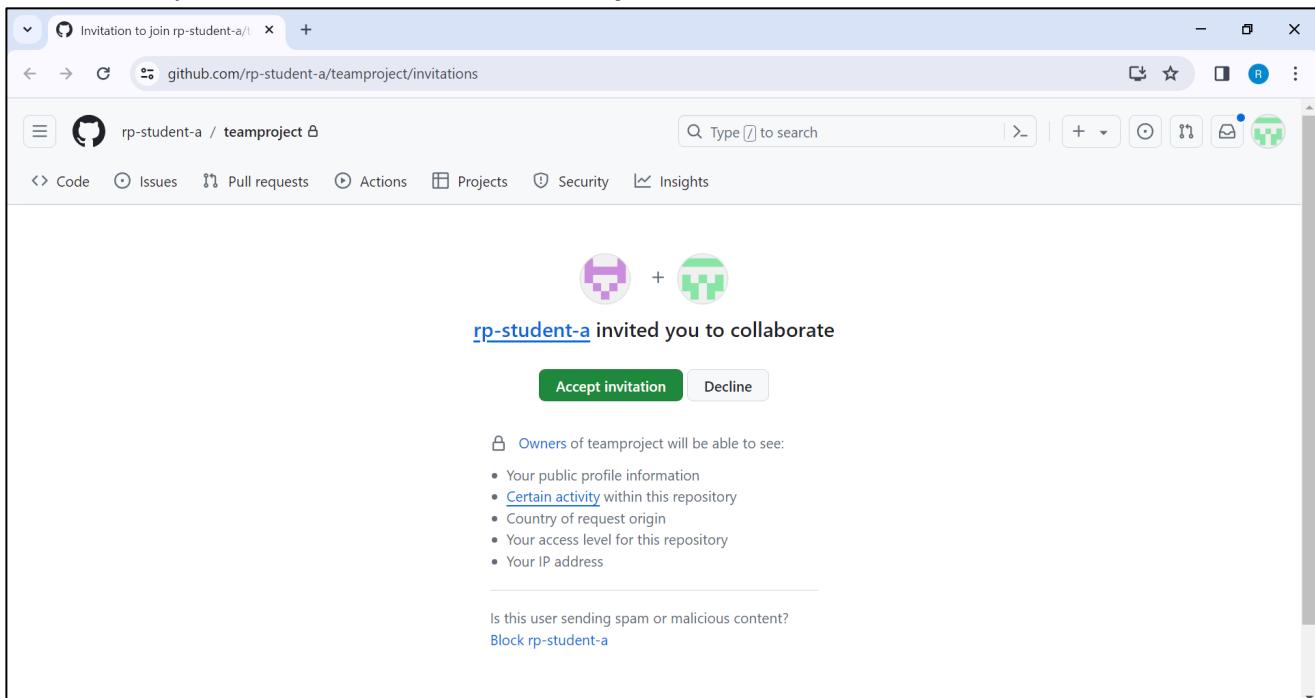
Accept invite

X 12. Open the email for StudentX to see the invitation, click on **View invitation**



The screenshot shows a Gmail inbox with one unread email. The subject line is "rp-student-a invited you to rp...". The email body contains the text "@rp-student-a has invited you to collaborate on the **rp-student-a/teamproject** repository". Below this, there is a note: "You can [accept](#) or [decline](#) this invitation. You can also visit [@rp-student-a](#) to learn a bit more about them." and "This invitation will expire in 7 days." At the bottom is a blue "View invitation" button. The sidebar on the left shows the "Inbox" tab selected, along with other options like Starred, Snoozed, Sent, Drafts, and More.

X 13. GitHub opens in a browser, click on **Accept invitation**



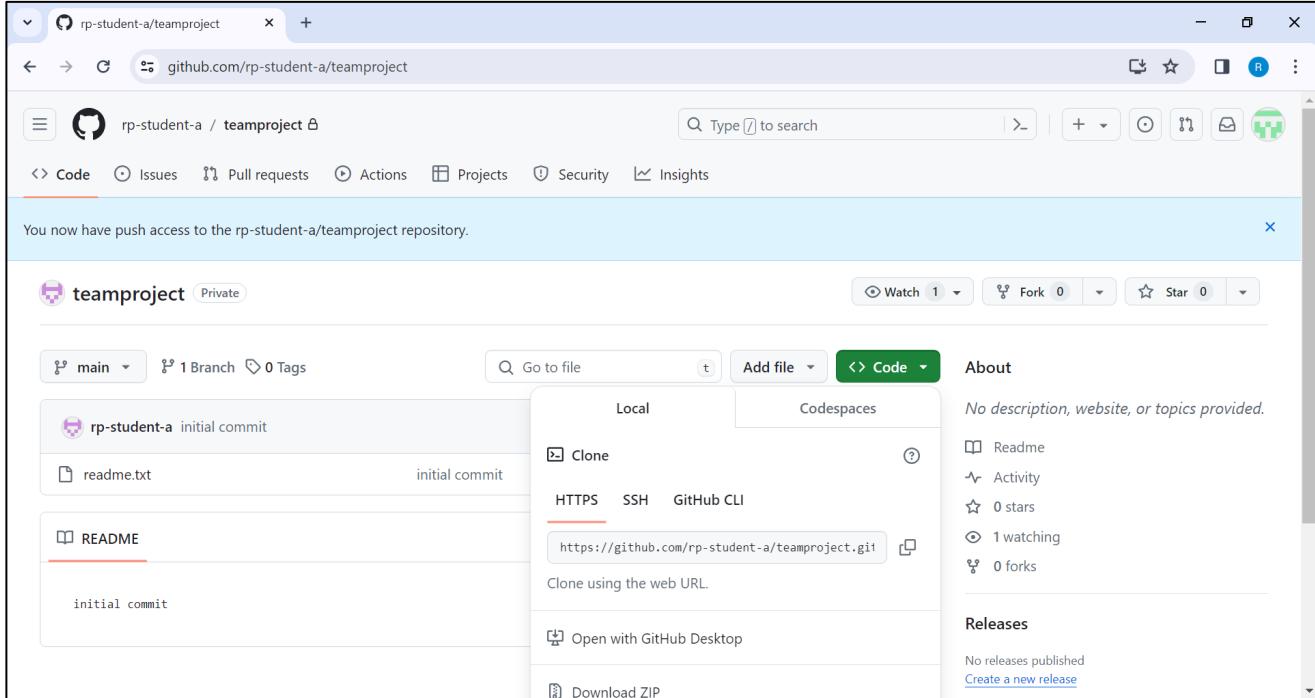
The screenshot shows a GitHub invitation page. The top bar indicates the URL is "github.com/rp-student-a/teamproject/invitations". The main content displays the message "rp-student-a invited you to collaborate". Below this are two buttons: a green "Accept invitation" button and a white "Decline" button. A note below the buttons states: "Owners of teamproject will be able to see:" followed by a bulleted list: "Your public profile information", "Certain activity within this repository", "Country of request origin", "Your access level for this repository", and "Your IP address". At the bottom of the page is a link "Block rp-student-a". The GitHub navigation bar at the top includes Code, Issues, Pull requests, Actions, Projects, Security, and Insights.

Get the project web URL

- X 14. Click on the **Code** button, then click on the copy button to copy the project web URL

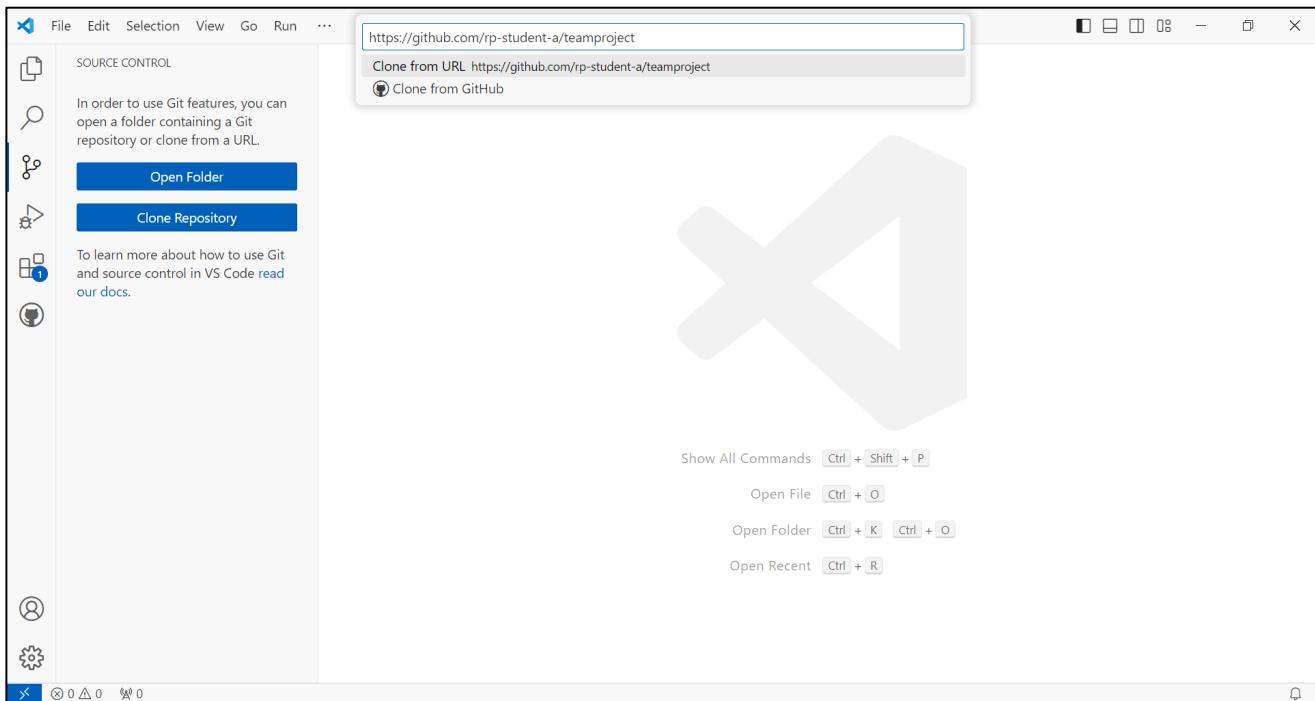
The project web URL in the example below is

<https://github.com/rp-student-a/teamproject.git>

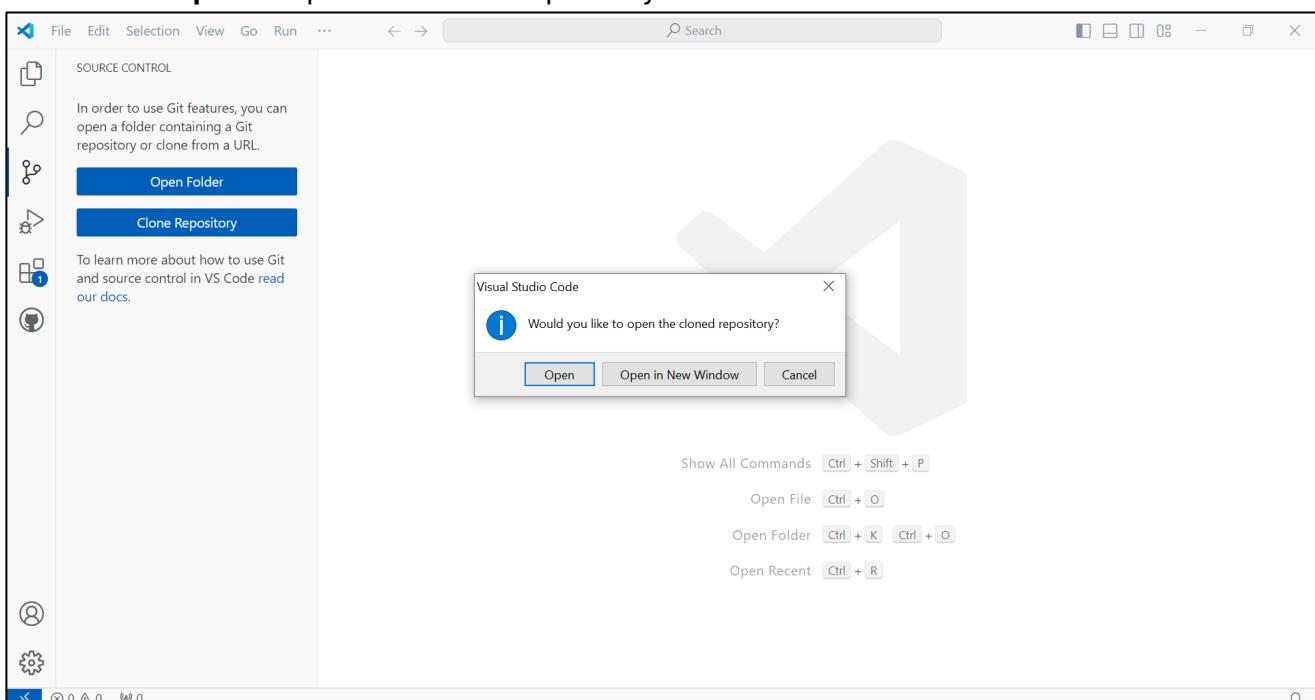


Clone repository

- X 15. Click on the source control button, and **Clone Repository**
 Enter the project web URL and click **Clone from URL**
 Select a folder to clone the project to



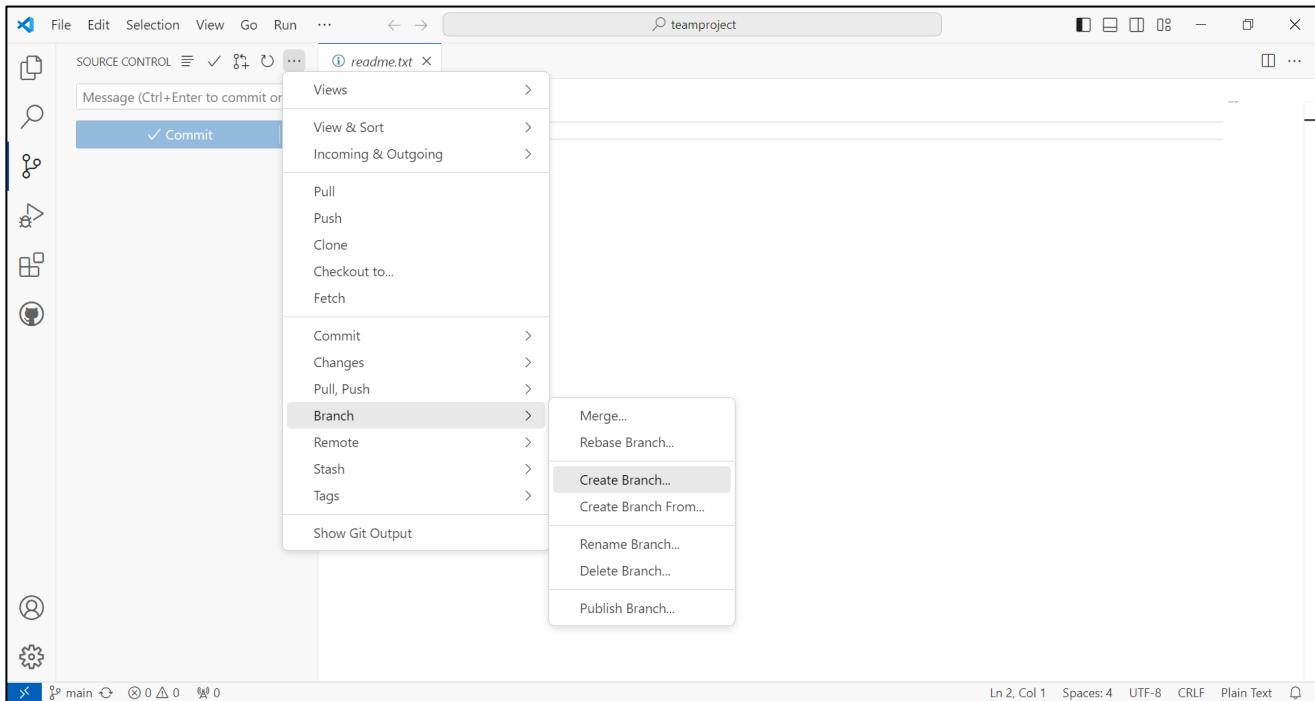
- X 16. Click on **Open** to open the cloned repository in Visual Studio Code



At this point, the project also exists on StudentX's computer

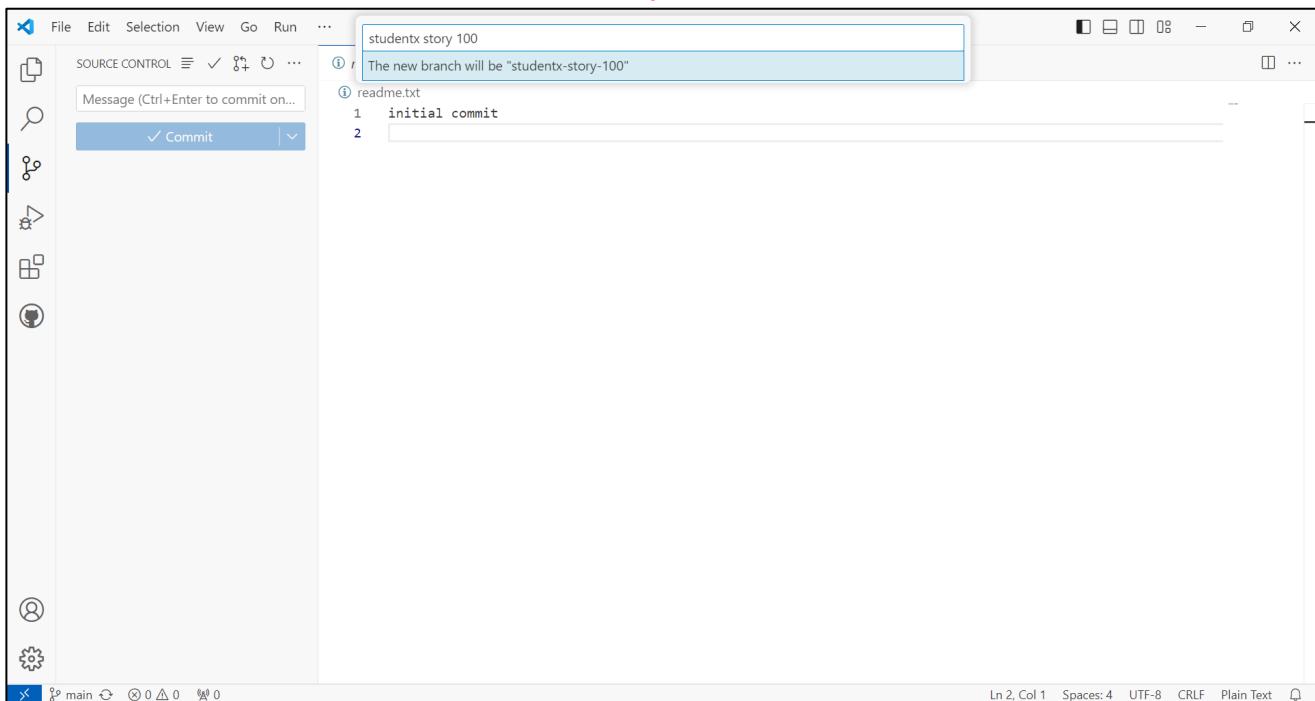
Create branch

X 17. Click on the ellipsis, select **Branch, Create Branch...**



Before you make any changes to the files, create a branch with a meaningful name, for example, the name of the feature you are working on. In agile development, this corresponds to a story (more on that in later lessons).

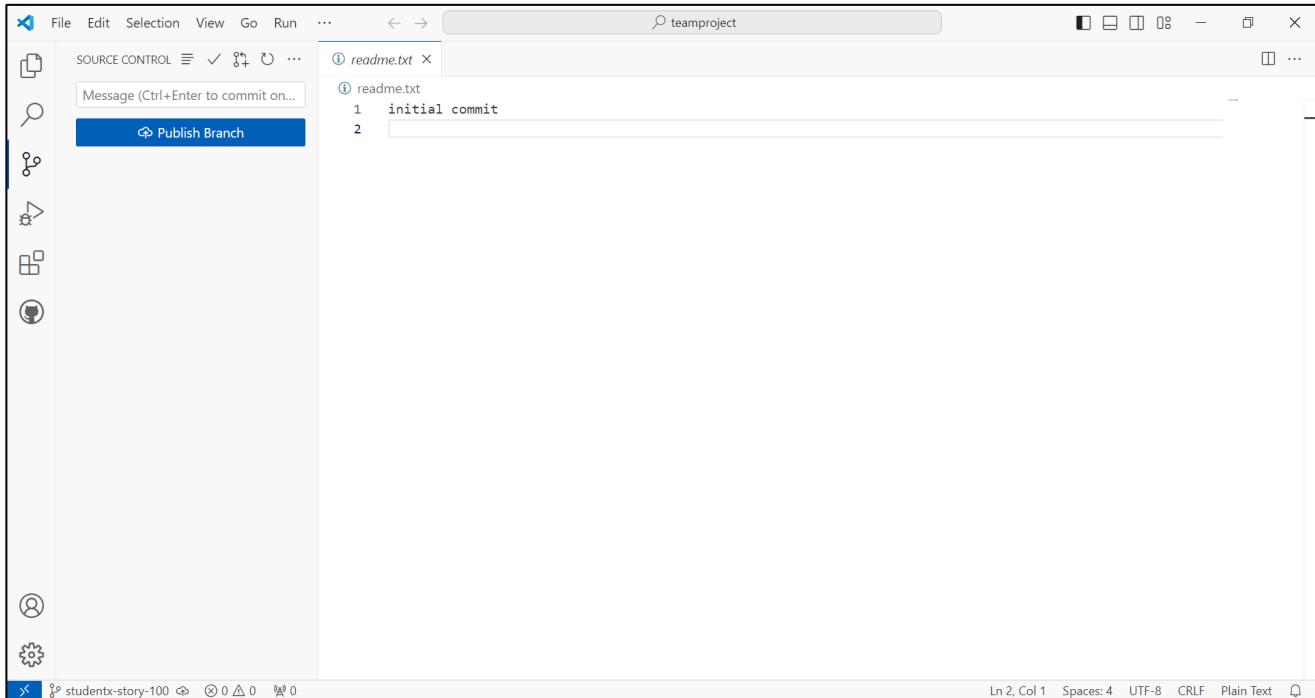
X 18. Enter the branch name as **studentx_story_100**
The new branch will be called **studentx-story-100**



Publish branch

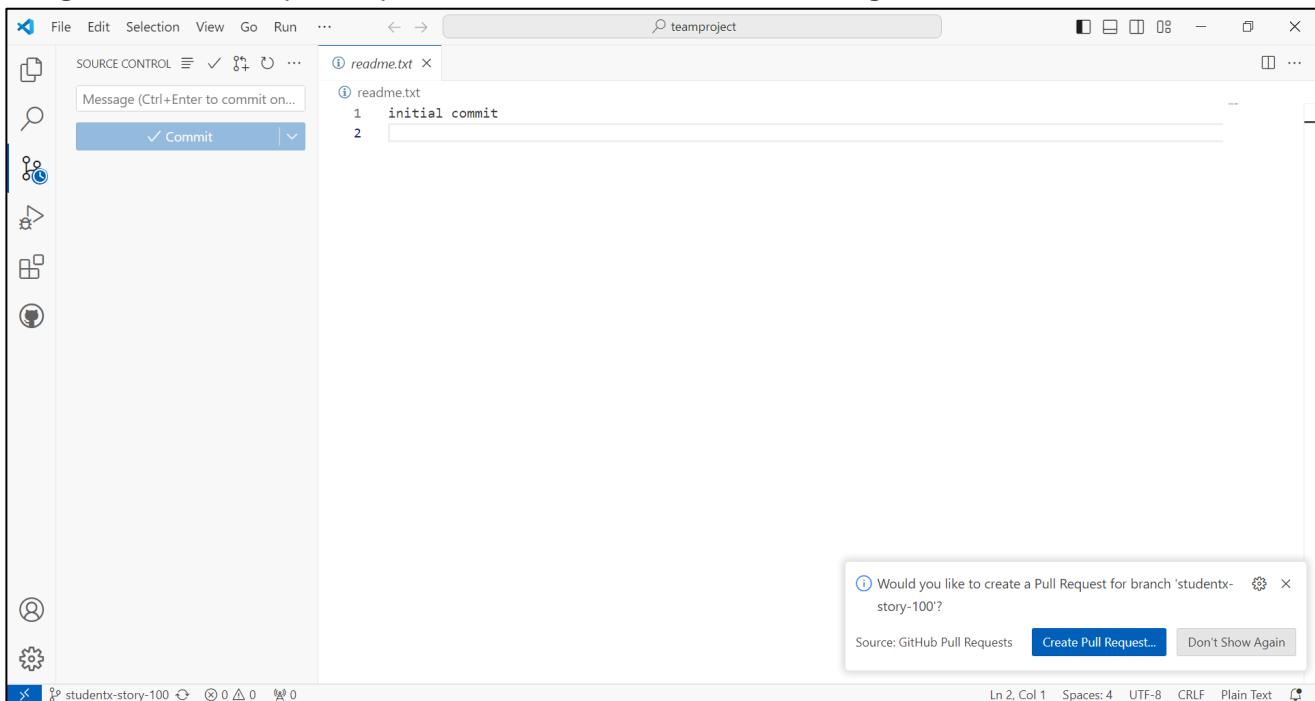
The bottom left corner in Visual Studio Code shows you are on branch **studentx-story-100**
 This branch only exists in the local repository on StudentX's computer
 We will need to publish it to the remote repository on GitHub

X 19. Click on Publish Branch



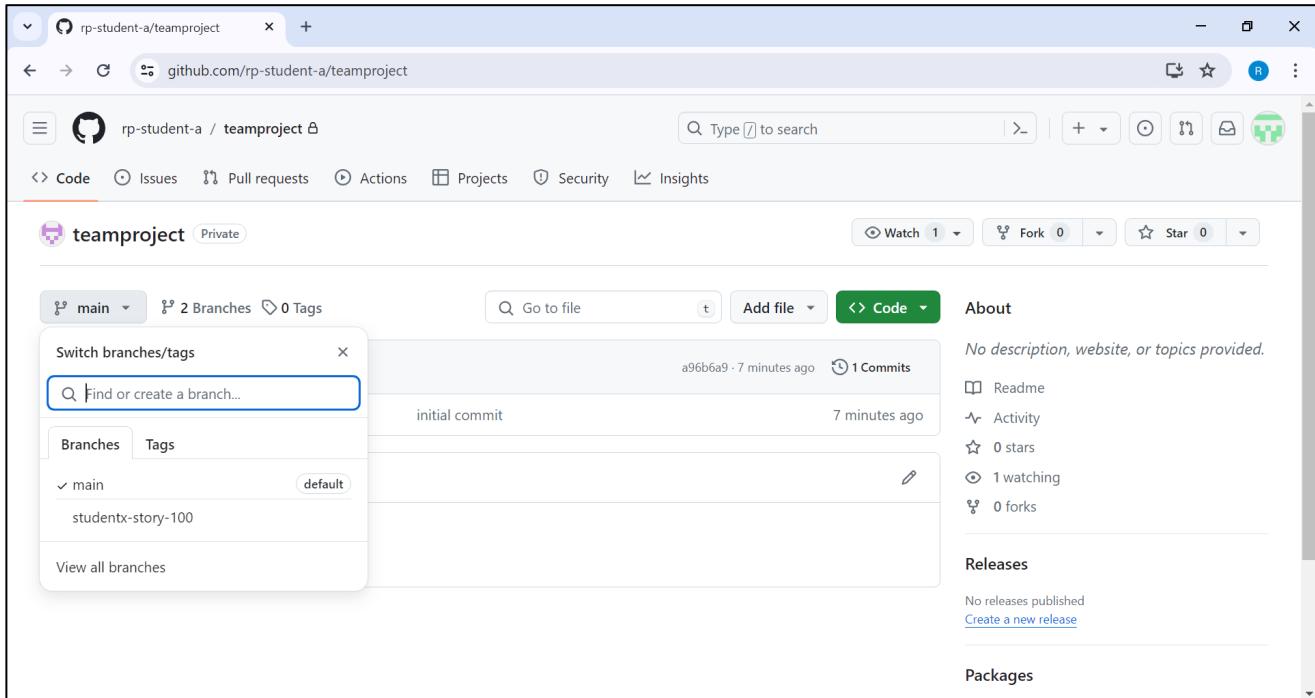
X 20. You should see a dialog to Create Pull Request...

Ignore it for now, pull requests will not be covered at this stage



X 21. View the repository in GitHub in a browser

Click on the dropdown next to **main**, you should see that there are 2 branches - **main** and **studentx-story-100**



The screenshot shows a GitHub repository page for 'rp-student-a/teamproject'. On the left, a dropdown menu is open under the 'main' branch, showing two branches: 'main' (selected) and 'studentx-story-100'. The main content area displays a single commit titled 'initial commit' made 7 minutes ago. The right sidebar contains sections for 'About' (no description), 'Activity' (0 stars, 1 watching, 0 forks), 'Releases' (no releases), and 'Packages'.

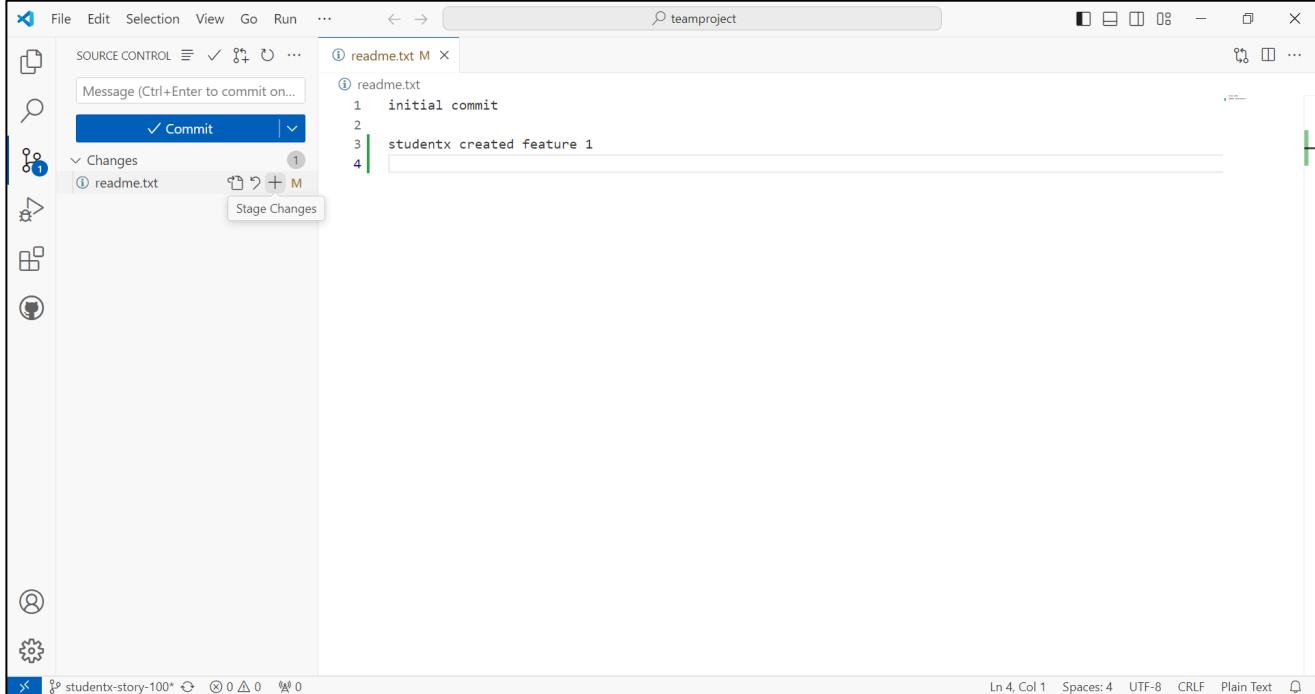
The project from the **main** branch should only contain the stable and fully tested code

The project from the **studentx-story-100** branch contains the code that StudentX is working on for the feature defined by story 100. Before the feature is completed, this branch may contain code that is only partially complete.

Make changes, stage changes, commit changes, sync changes

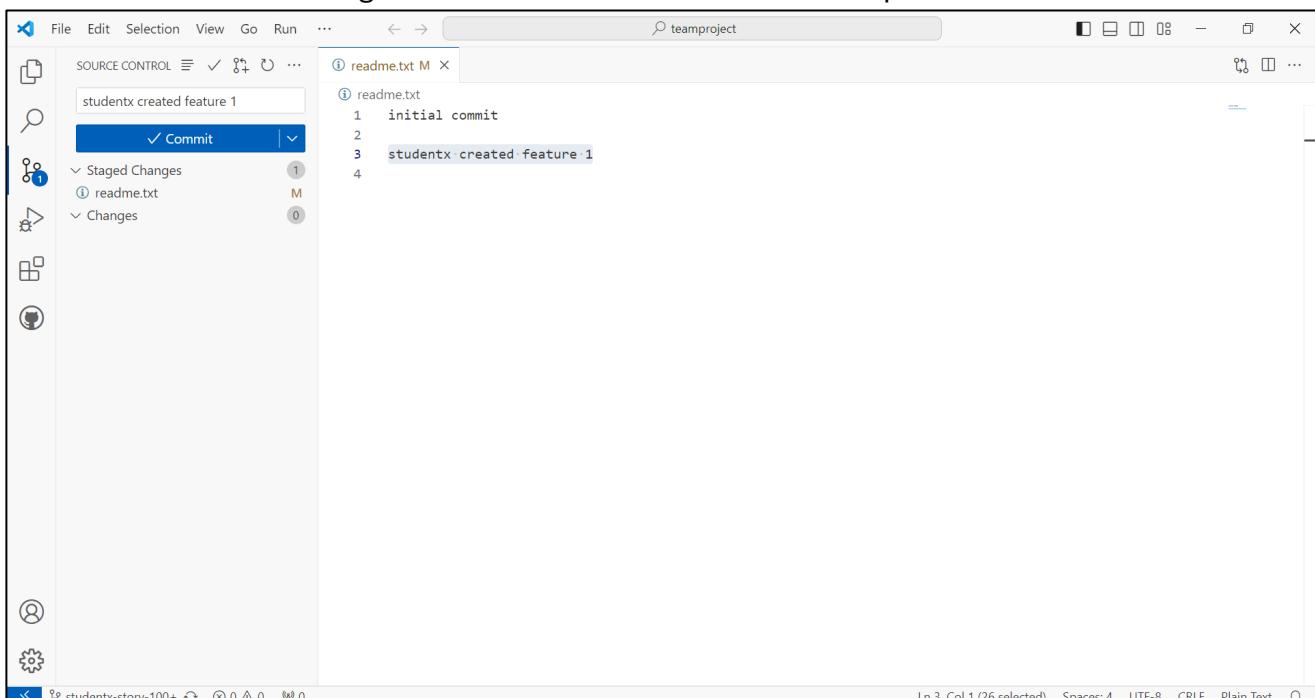
- X 22. Modify `readme.txt` by adding the text `studentx created feature 1` on line 3 and add a new line at line 4

Click on the `+` button next to `readme.txt` to stage changes for that file



```
initial commit
studentx created feature 1
```

- X 23. Set the commit message as `studentx created feature 1` and press **Commit**

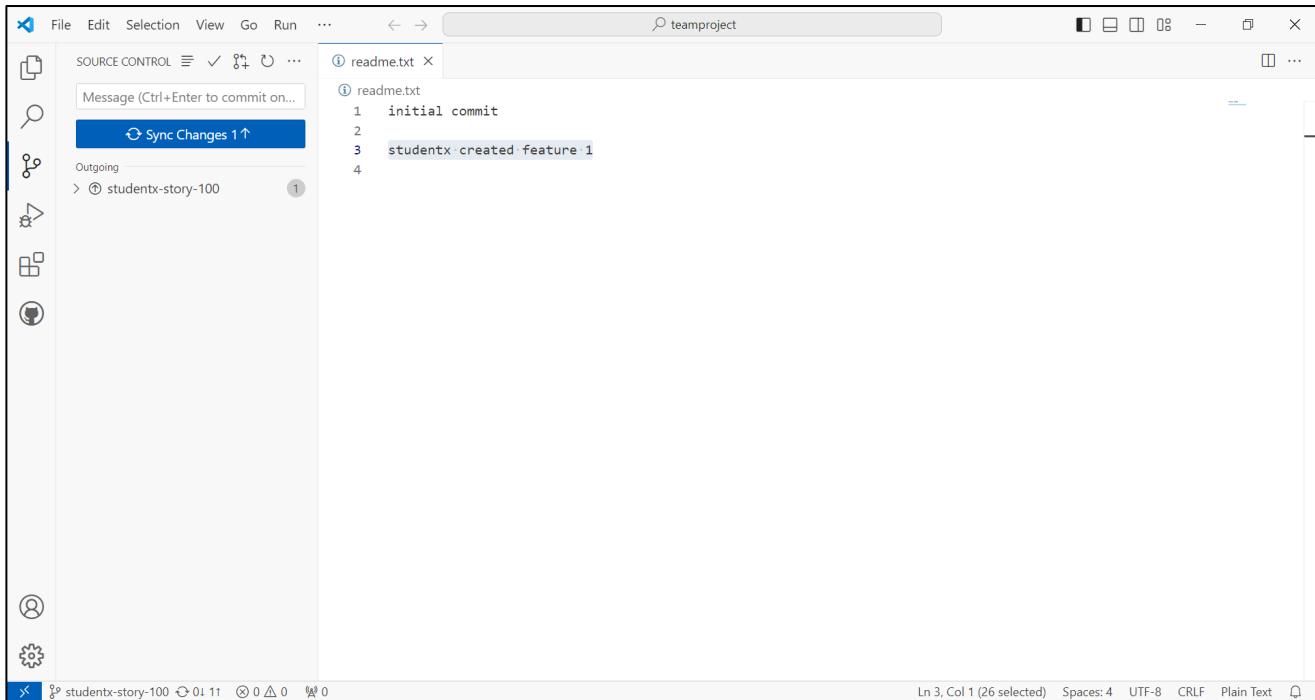


```
initial commit
studentx created feature 1
```

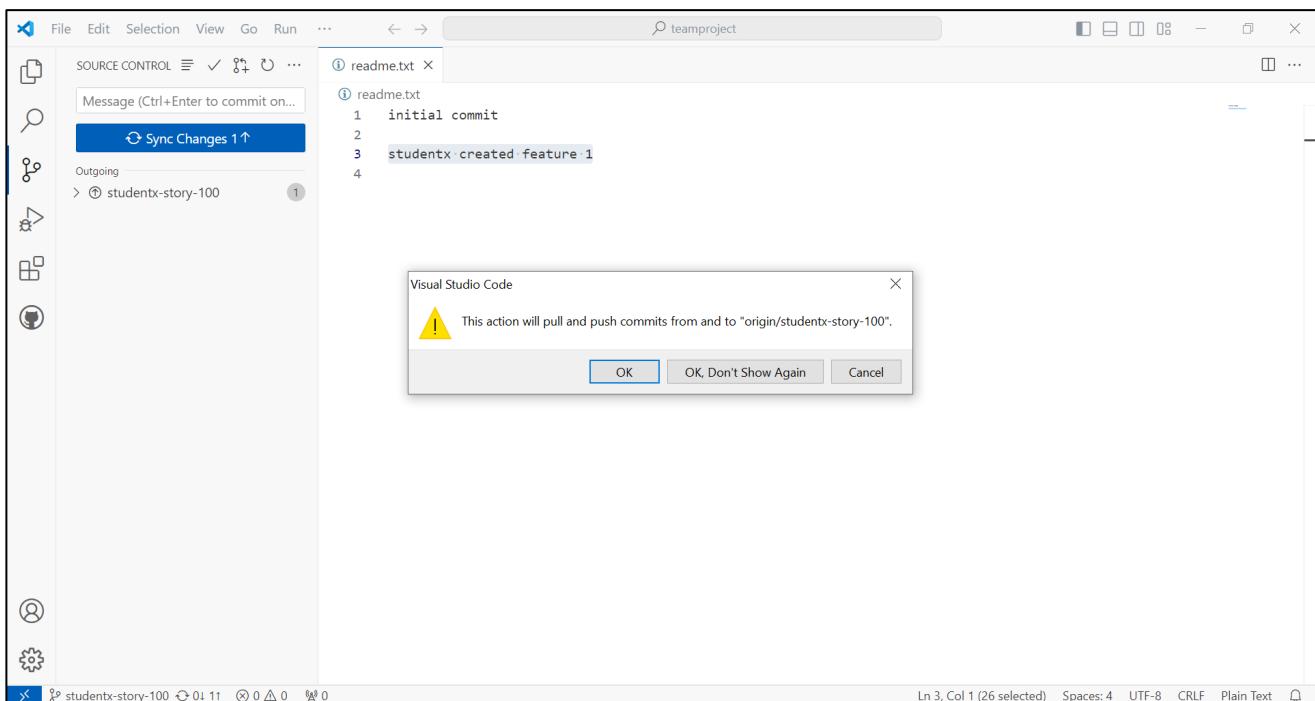
The changes are committed to the local repository

We need to sync the changes in the local repository to the remote repository

X 24. Click on Sync Changes



X 25. The dialog informs you that sync changes will pull and push commits from and to **origin/studentx-story-100** Click OK



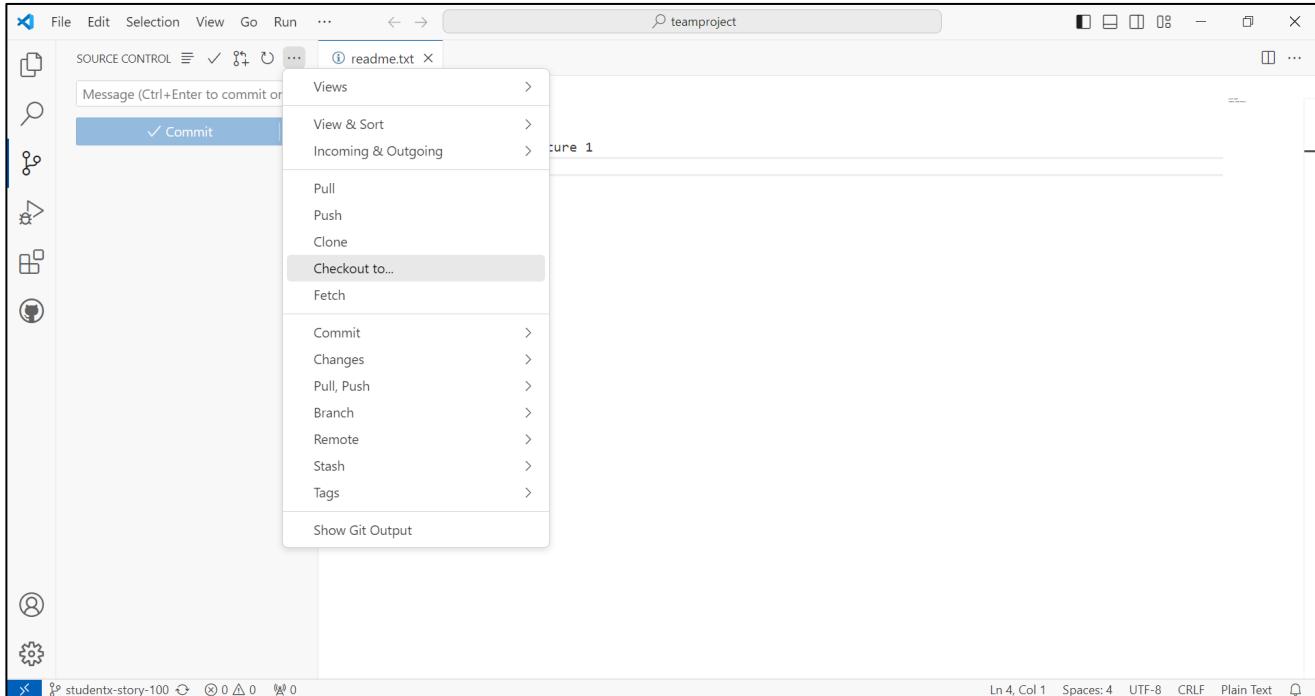
pull will pull the changes from the remote repository to the local repository
push will push the changes from the local repository to the remote repository

Checkout main

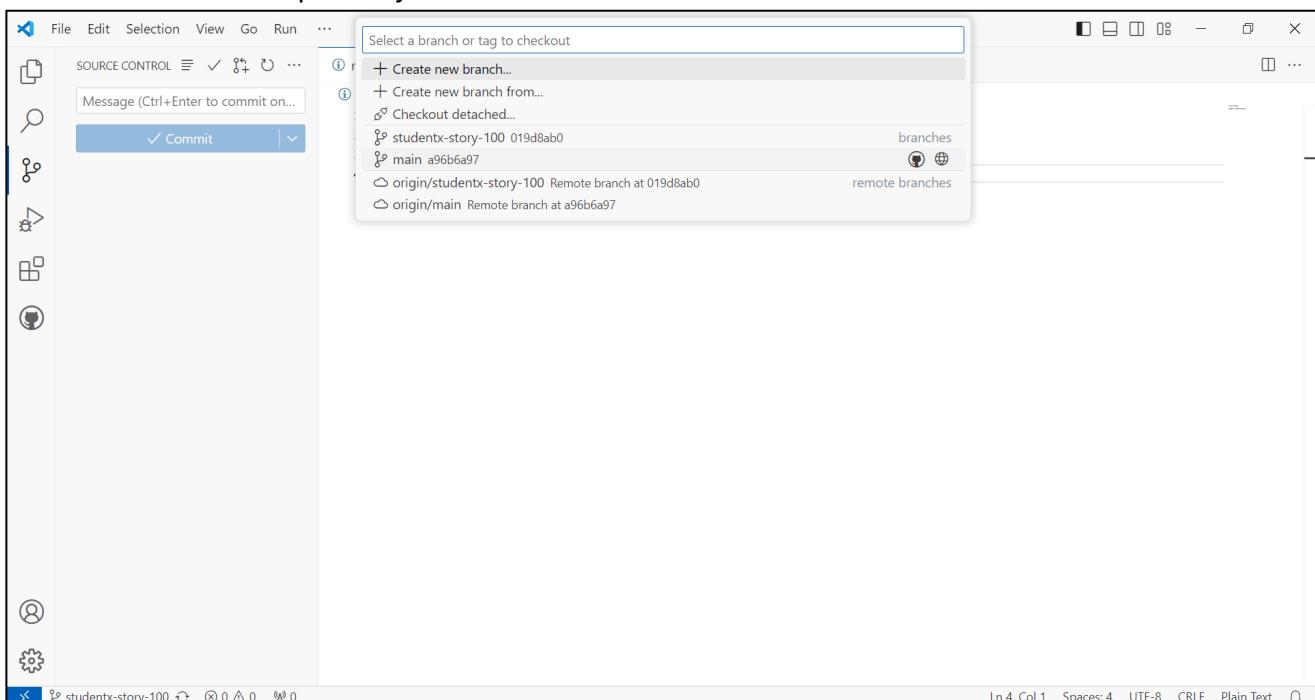
Once StudentX has completed the feature, it is time to merge the changes to the main branch
 There are several steps to this process

- 1) checkout to the local repository branch main
- 2) merge from the story branch to the main branch

X 26. Click the ellipsis, select **Checkout to..**



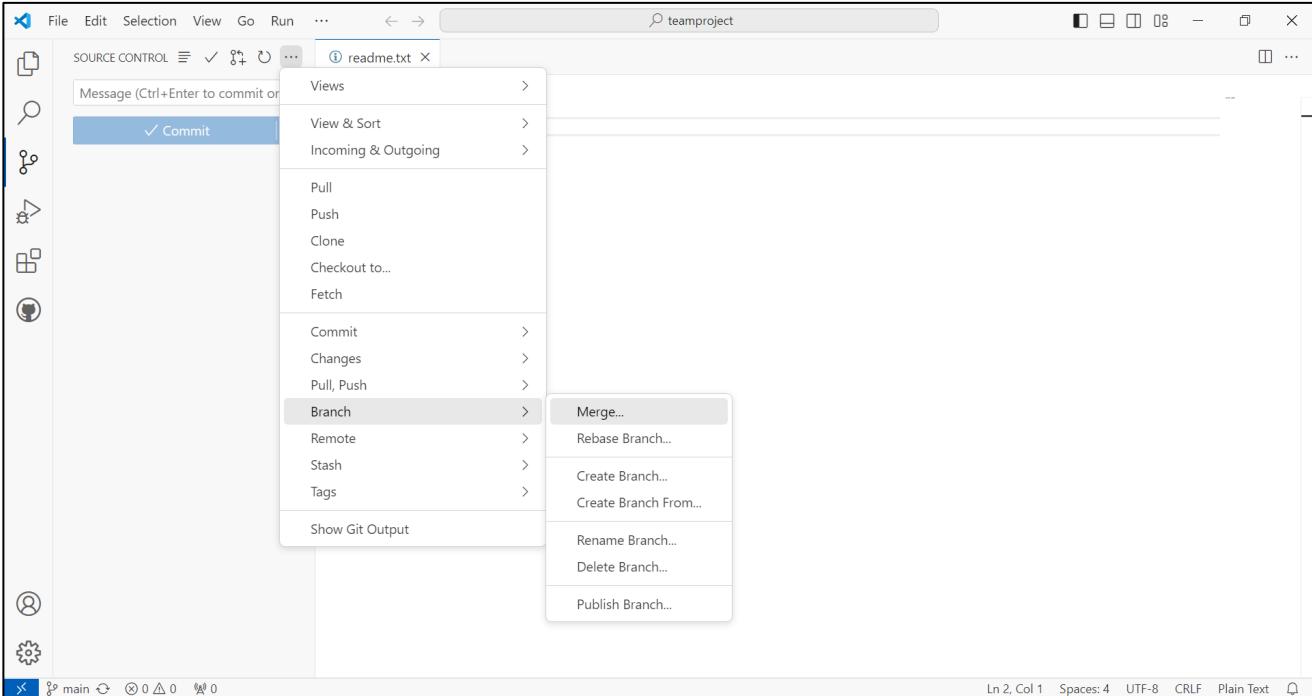
X 27. Select the local repository branch **main**



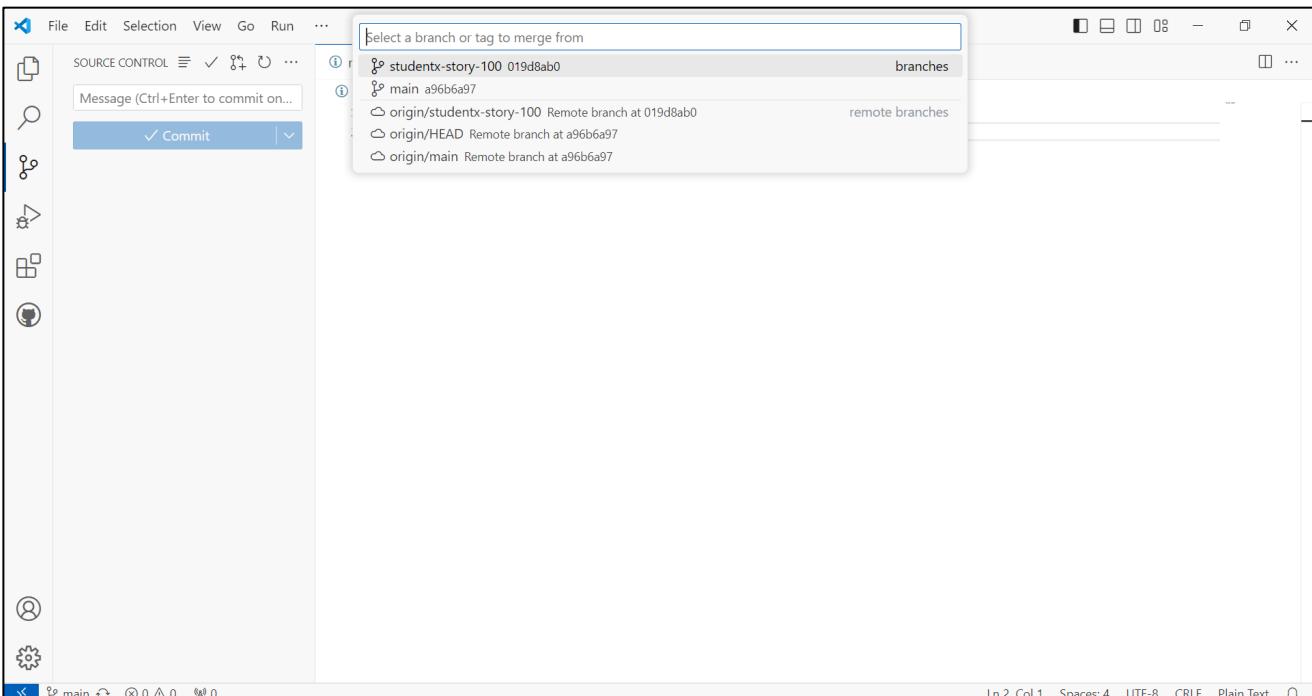
Merge branch

The bottom left corner in Visual Studio Code shows you are on the local repository branch **main**. We now need to merge from the local repository branch **studentx-story-100** to the local repository branch **main**.

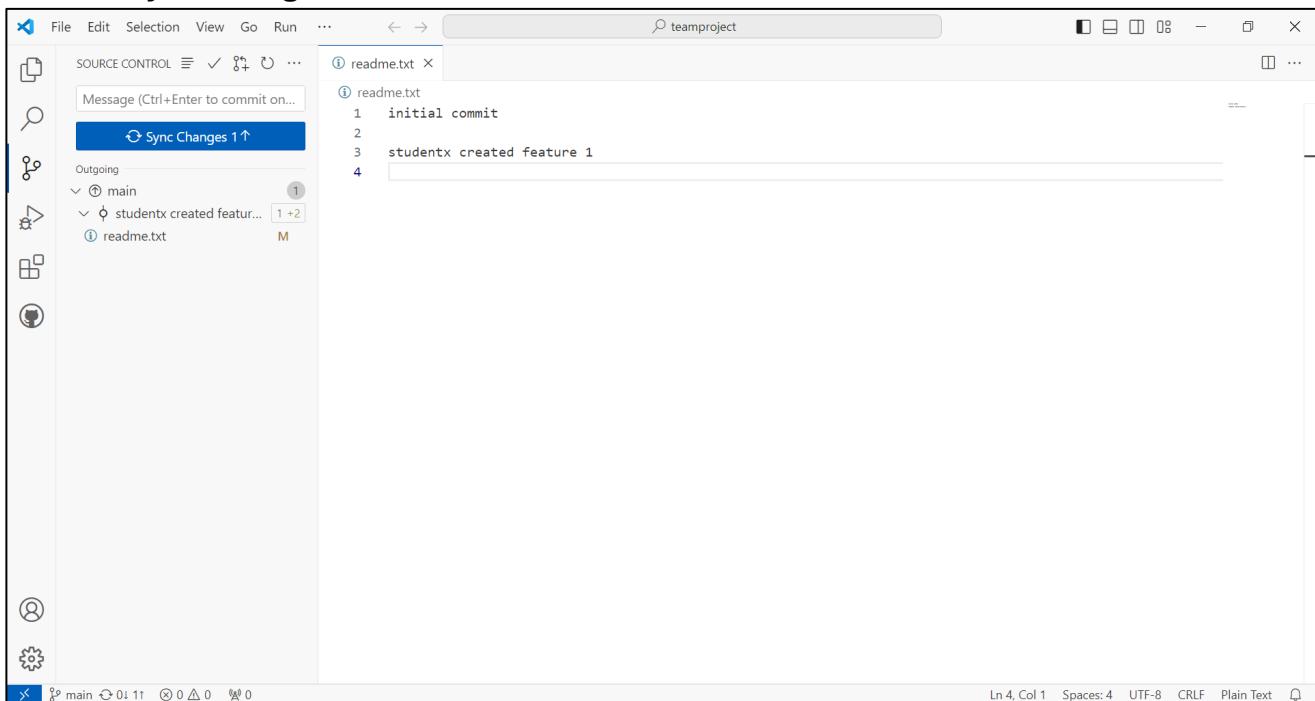
X 28. Click on the ellipsis, select **Branch, Merge..**



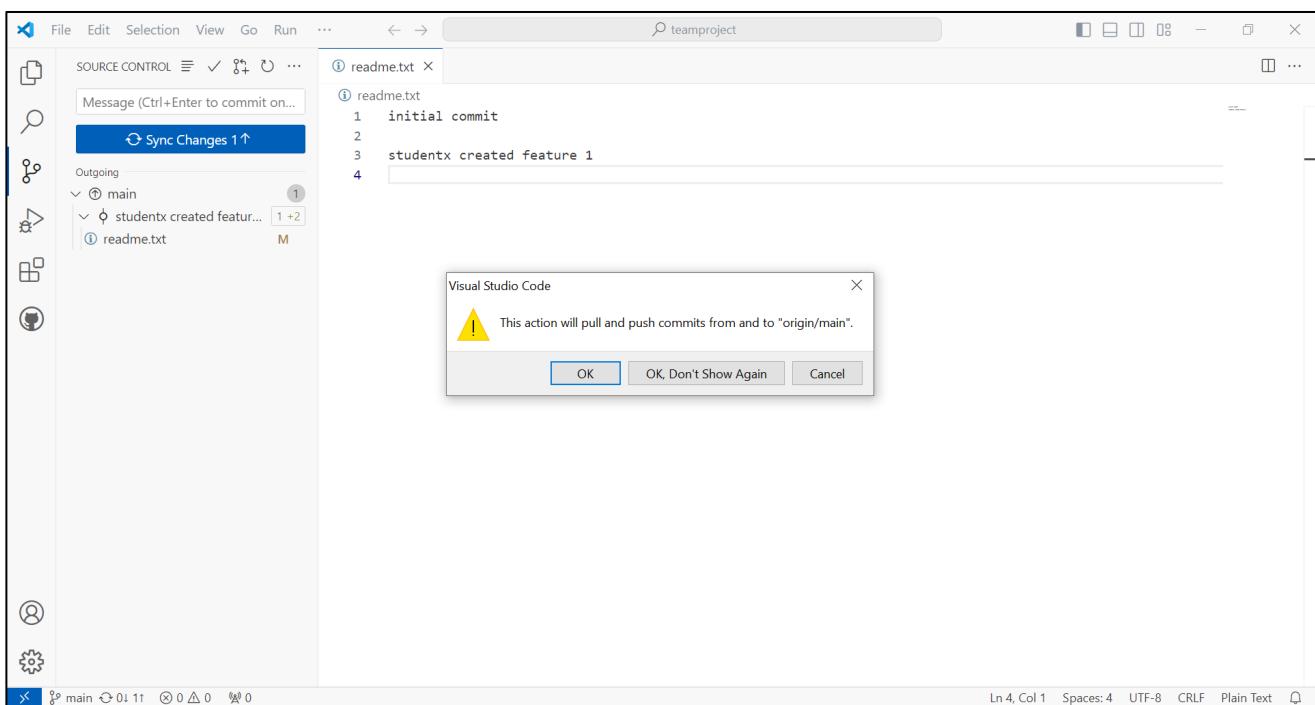
X 29. Select the local repository branch **studentx-story-100**



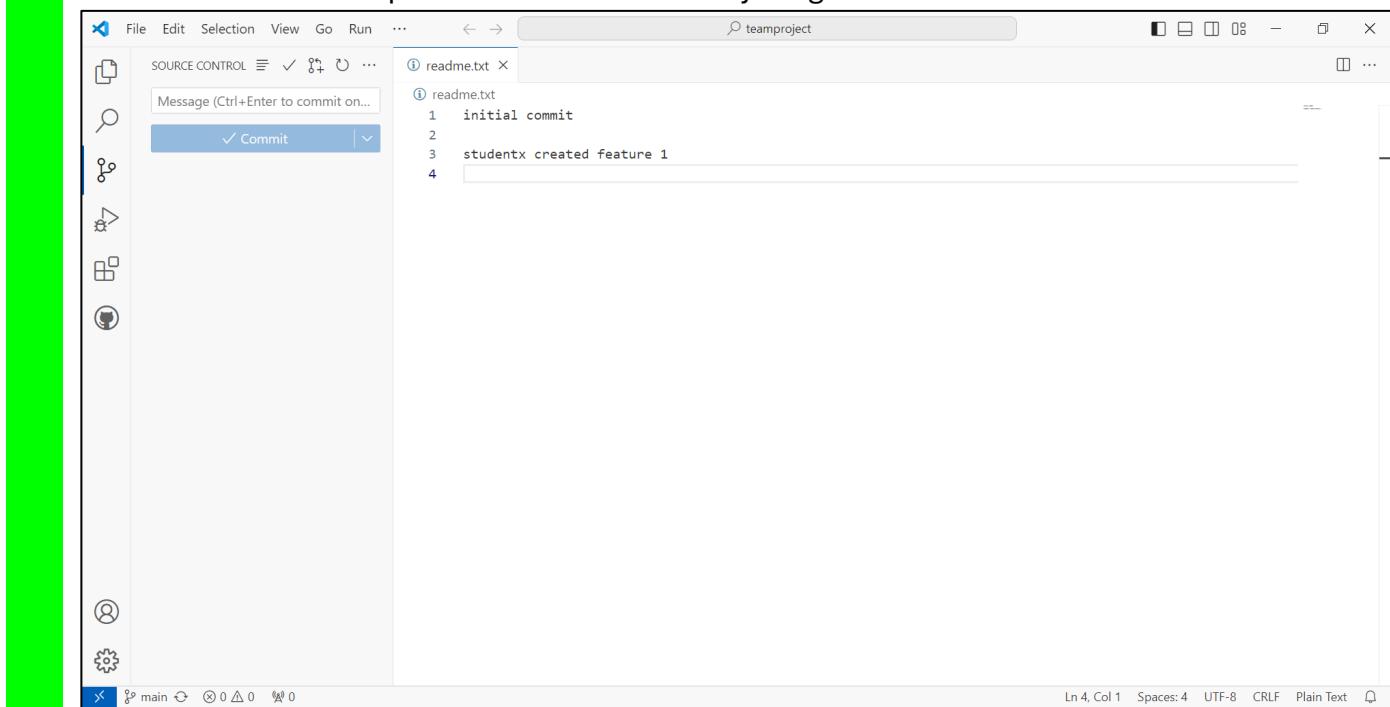
X 30. Click Sync Changes



X 31. The dialog informs you that sync changes will pull and push commits from and to origin/main Click OK



X 32. The source control panel should not show anything of note



The screenshot shows the Visual Studio Code interface with the Source Control panel open. The panel displays a single commit for the file 'readme.txt'. The commit message is:

```
initial commit
studentx created feature 1
```

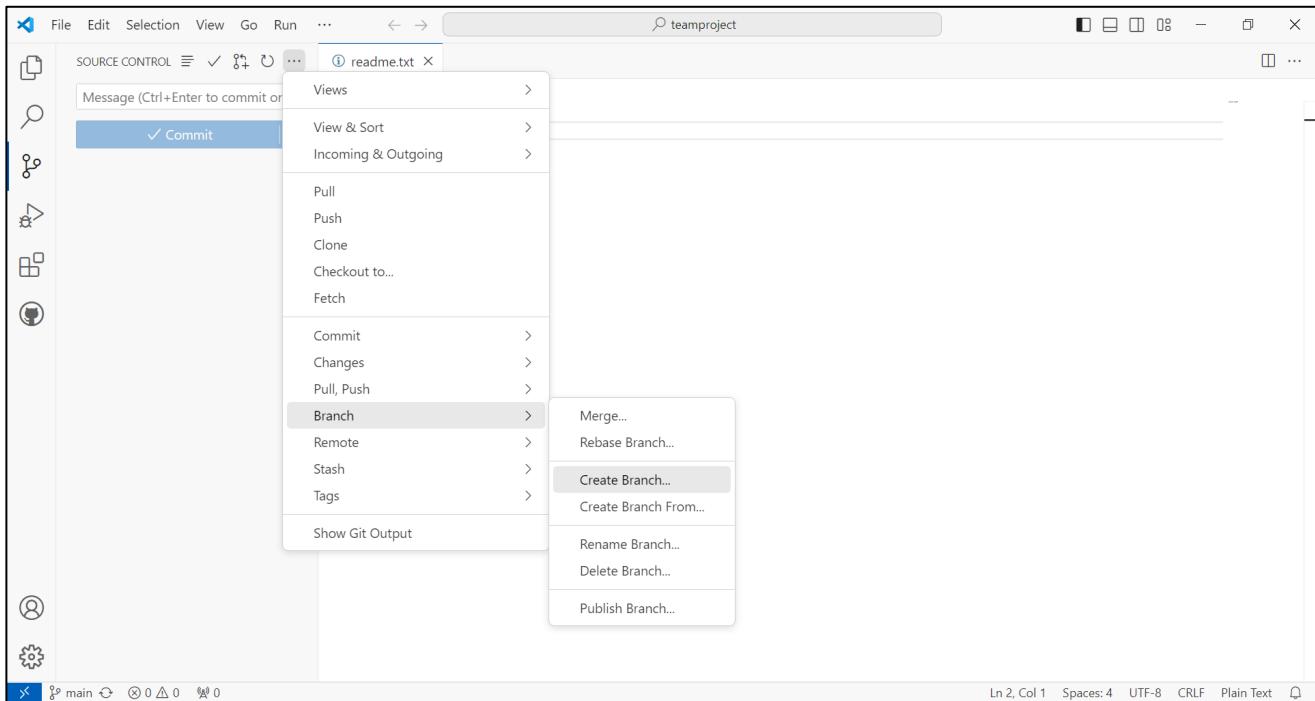
The commit has a status of 'Initial Commit' and a date of '4 days ago'. There is a blue 'Commit' button at the bottom of the commit list. The status bar at the bottom right shows 'Ln 4, Col 1' and other file details.

At this point, StudentX's local repository branch **main** is in sync with the remote repository branch **main**

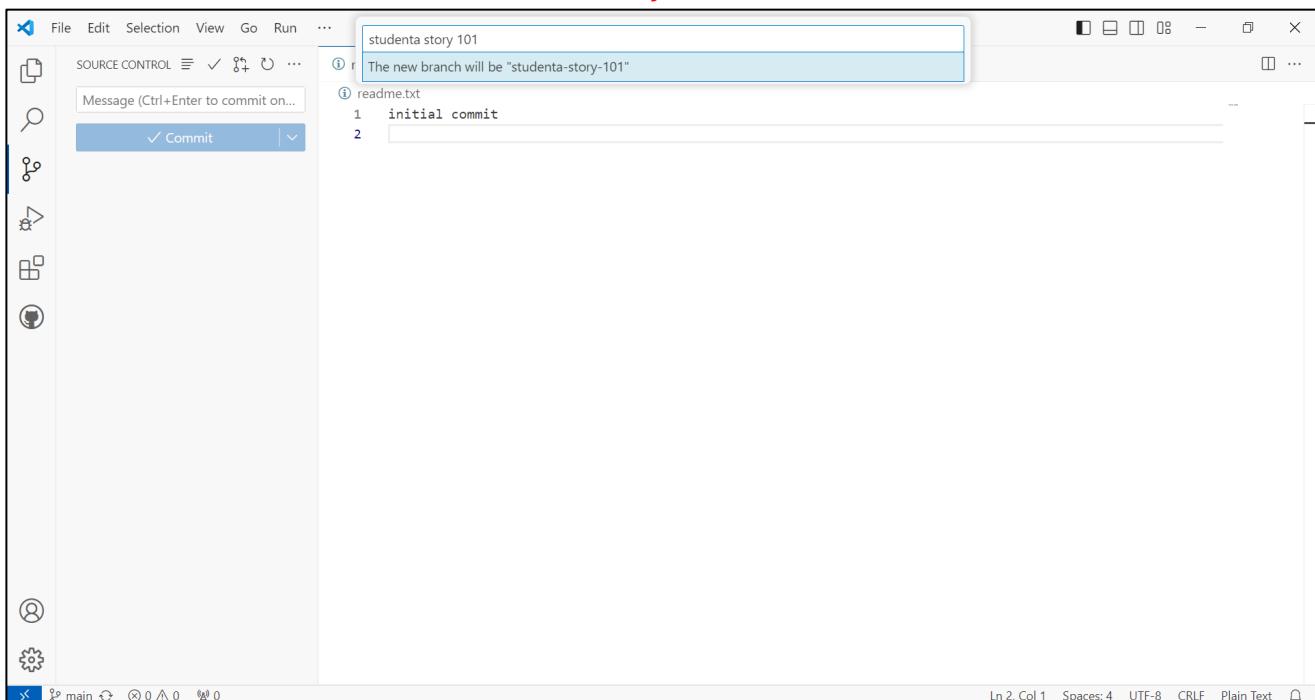
Create branch

The next few steps to create branch, publish branch, make changes, stage changes, commit changes, sync changes are the same as what StudentX did

- A 33. Click on the ellipsis, select **Branch**, **Create Branch...**

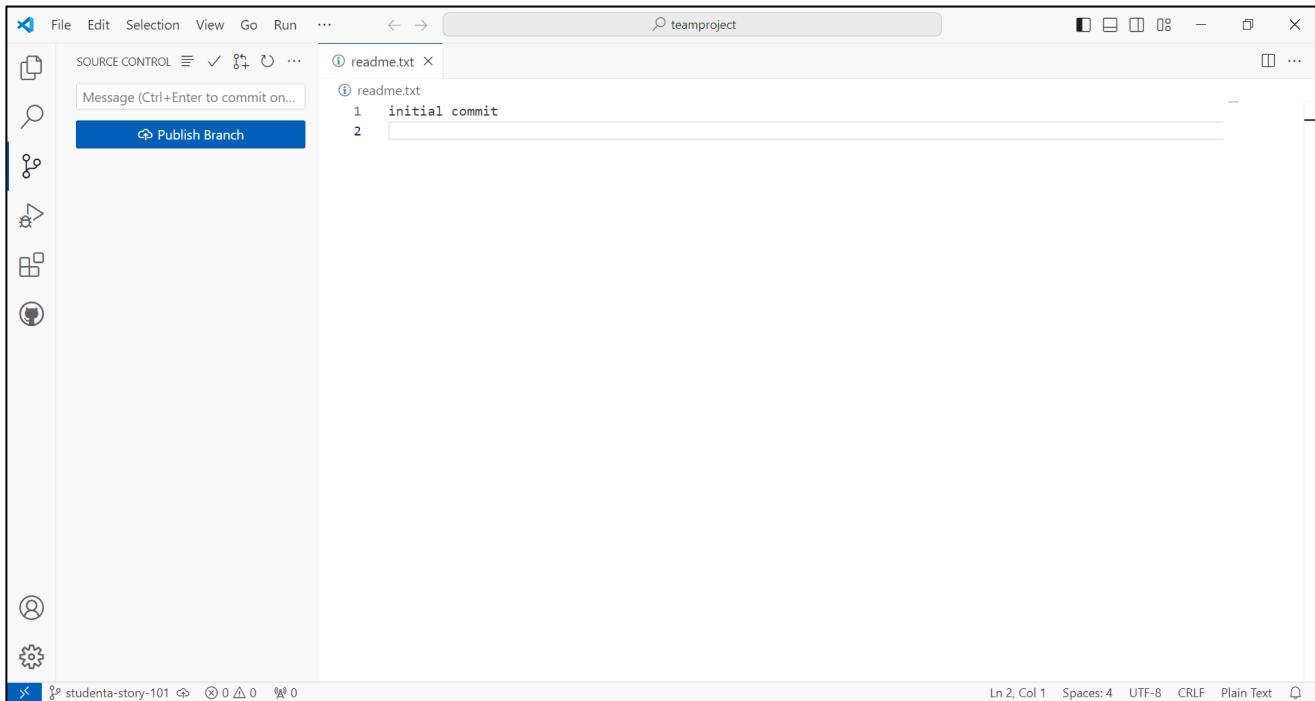


- A 34. Enter the branch name as **studenta_story_101**
The new branch will be called **studenta-story-101**

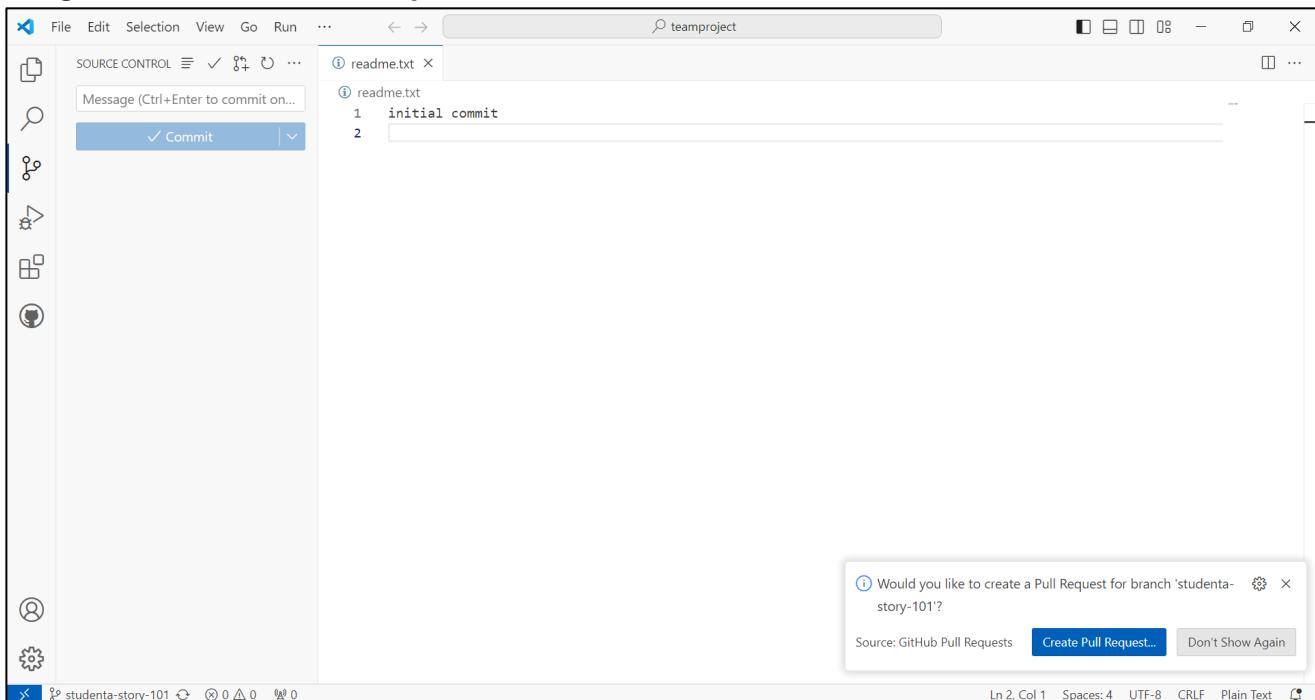


Publish branch

A 35. Click Publish Branch

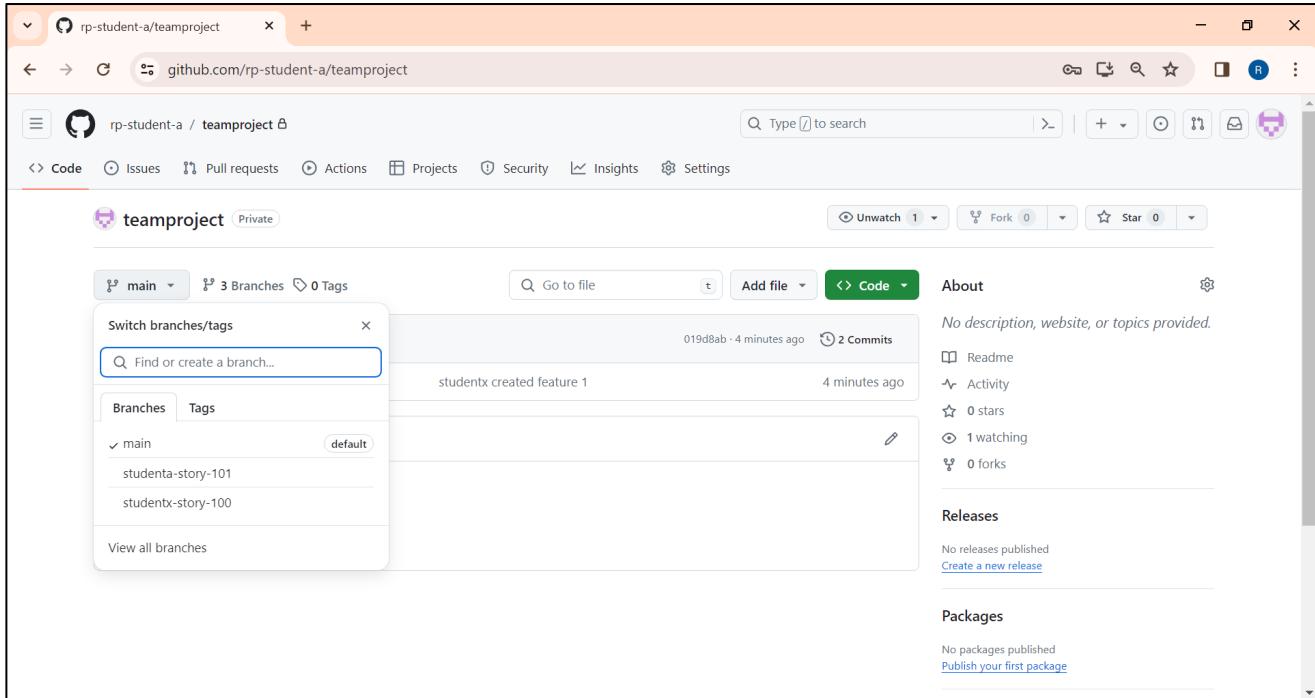


A 36. Ignore the Create Pull Request... button



A 37. View the repository in GitHub in a browser

Click on the dropdown next to **main**, you should see that there are 3 branches - **main**, **studentx-story-100** and **studenta-story-101**



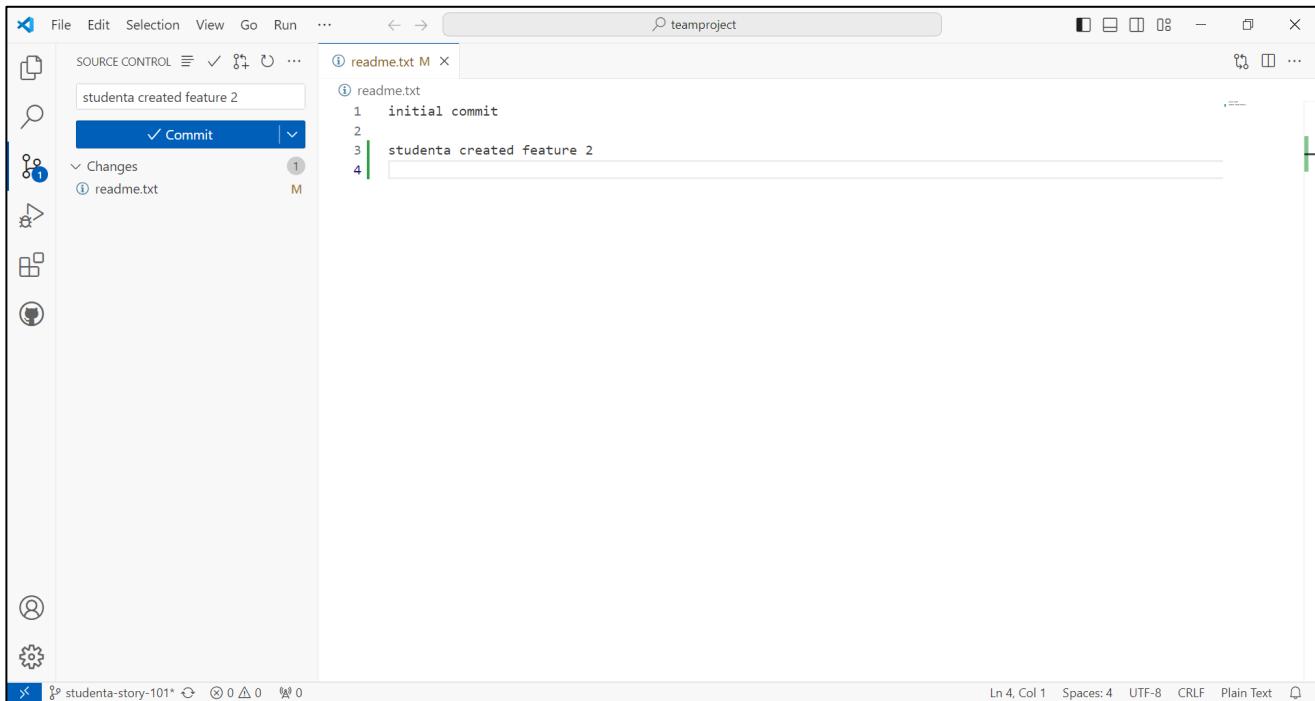
The screenshot shows a GitHub repository page for 'rp-student-a/teamproject'. The 'main' branch is selected in the dropdown menu. The page displays two commits from 'studentx': 'studentx created feature 1' (4 minutes ago) and another commit (4 minutes ago). On the left, a sidebar shows the 'Switch branches/tags' dropdown with three branches listed: 'main' (selected), 'studenta-story-101', and 'studentx-story-100'. The 'About' section indicates no description, website, or topics provided. The 'Releases' and 'Packages' sections are empty.

Make changes, stage changes, commit changes, sync changes

- A 38. Modify `readme.txt` by adding the text `studenta created feature 2` on line 3 and add a new line at line 4

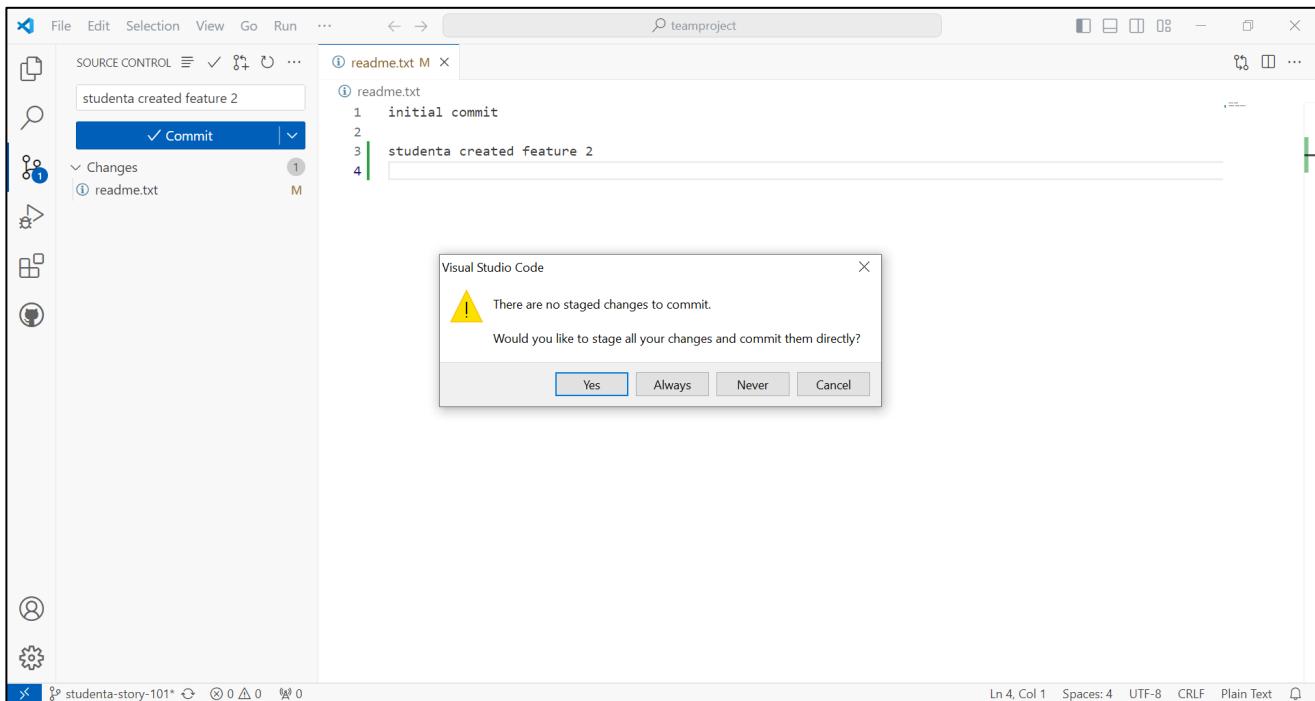
Set the commit message as `studenta created feature 2`

Click **Commit**

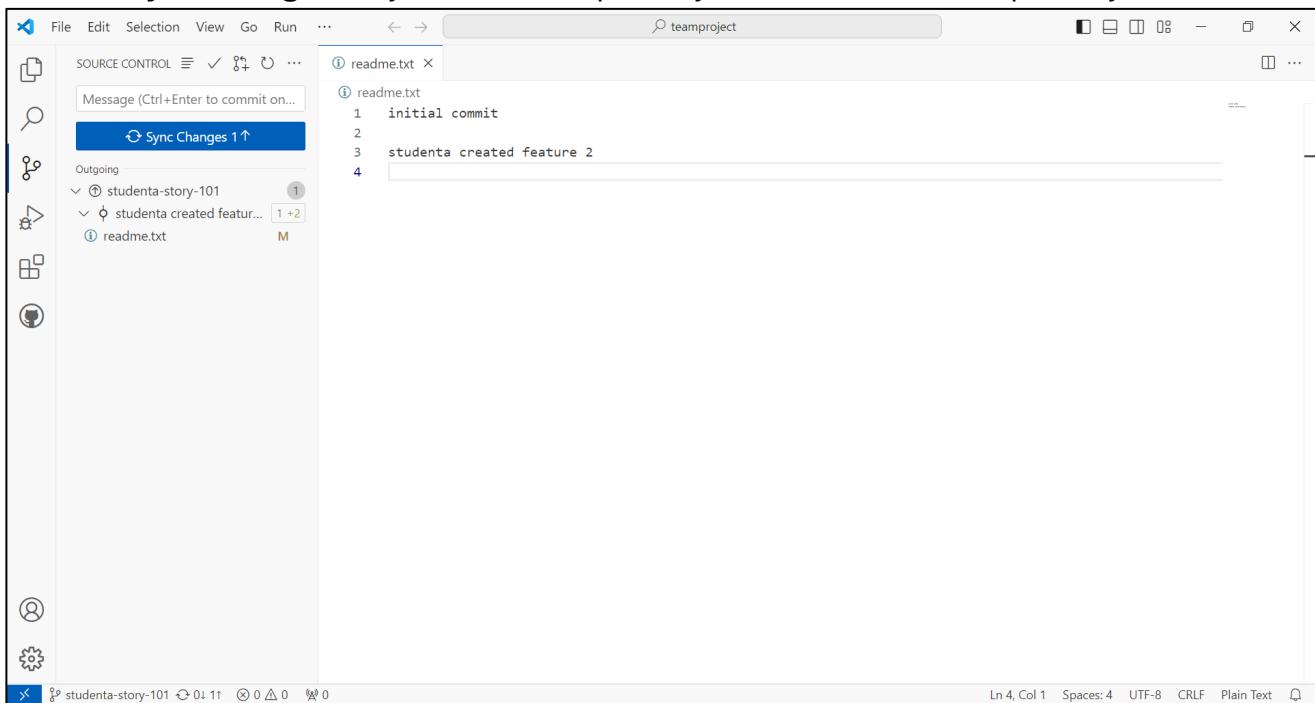


If you try to commit before staging changes, Visual Studio Code will ask if you want to stage **all** changes. Only do so if you are certain that you want to stage all changes

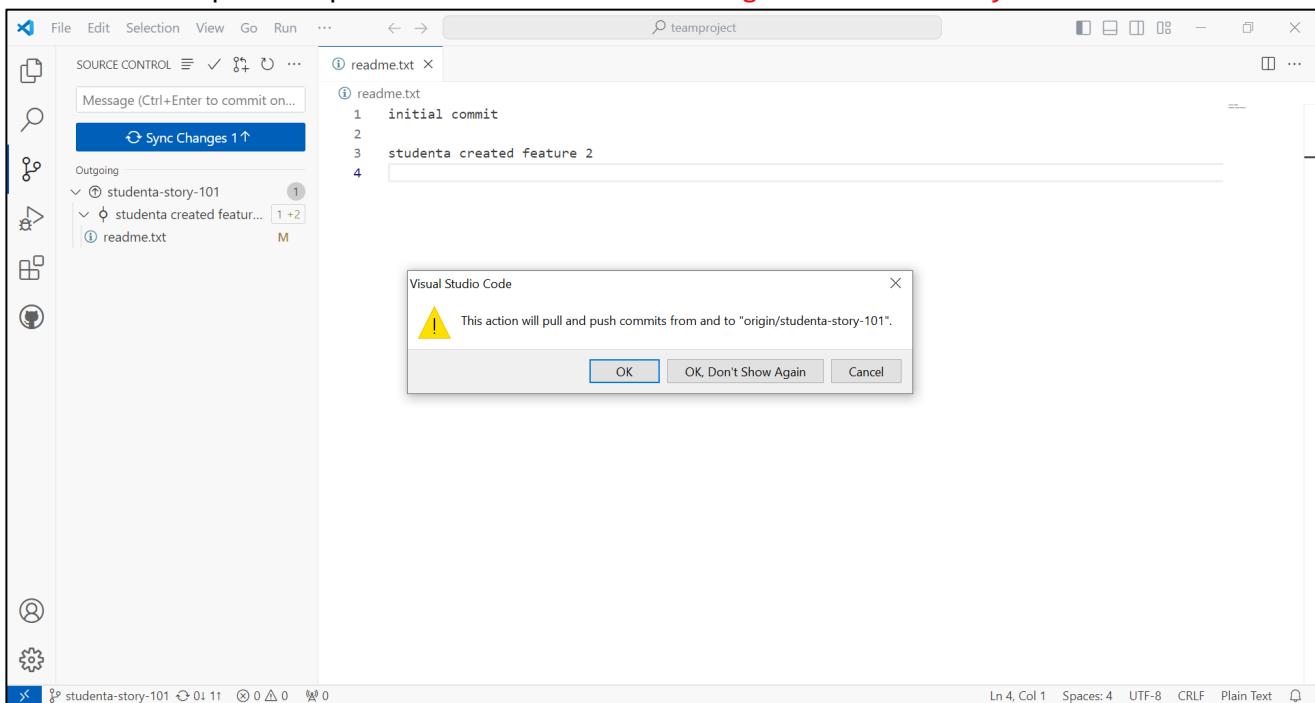
- A 39. Click **Yes** to stage all changes and commit them directly



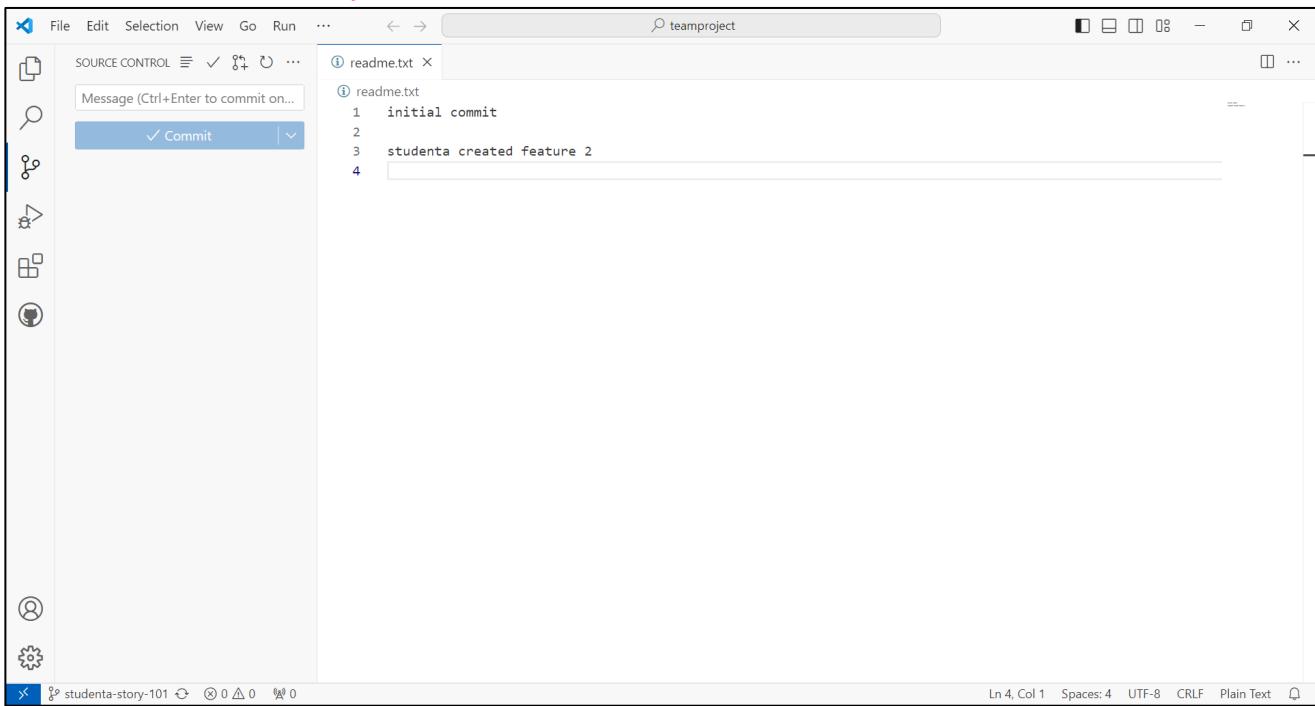
A 40. Click **Sync Changes** to sync the local repository branch and remote repository branch



A 41. Click **OK** to pull and push commits from and to **origin/studenta-story-101**



- A 42. The remote repository branch **studenta-story-101** is now in sync with the local repository branch **studenta-story-101**



The screenshot shows the Visual Studio Code interface with the Source Control tab selected. A commit message is displayed in the center pane:

```
① readme.txt x
① readme.txt
1 initial commit
2
3 studenta created feature 2
4
```

The status bar at the bottom indicates the file is part of the 'studenta-story-101' repository, with 0 changes, 0 additions, and 0 deletions. The status bar also shows the current line (Ln 4), column (Col 1), spaces (Spaces: 4), encoding (UTF-8), and file type (Plain Text).

Note that at this point, the local repository branch **main** is out of sync with the remote repository branch **main**

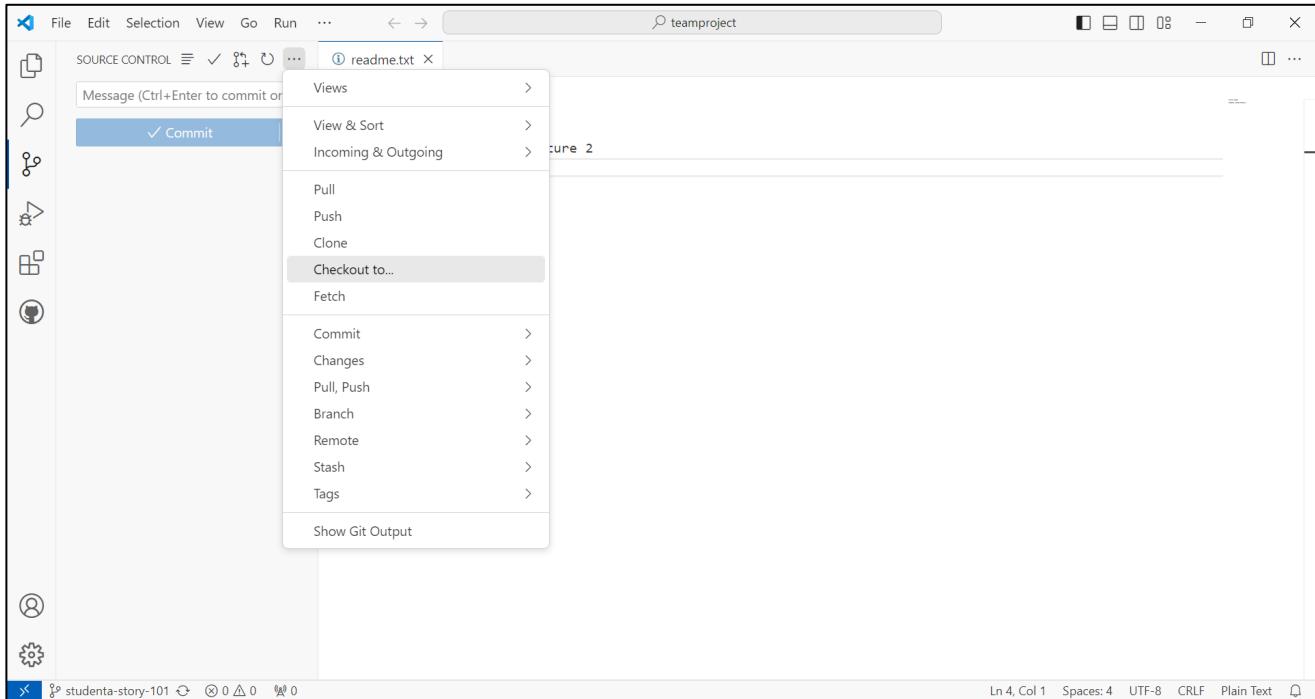
In the local repository branch **main**, line 3 says **studenta created feature 2**

In the remote repository branch **main**, line 3 says **studentx created feature 1**

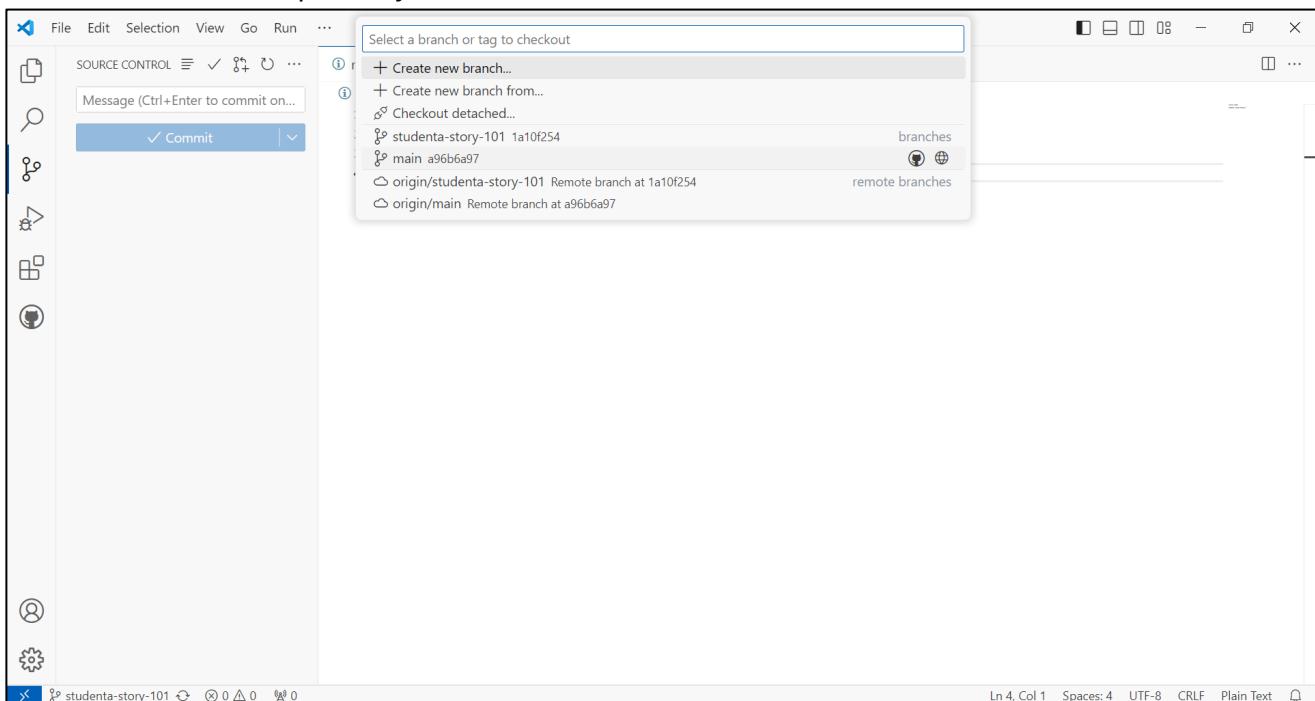
Checkout main

The next few steps to checkout main and merge branch are the same as what StudentX did. However, there will be a conflict we need to resolve when doing the merge branch

- A 43. Click the ellipsis, select **Checkout to..**

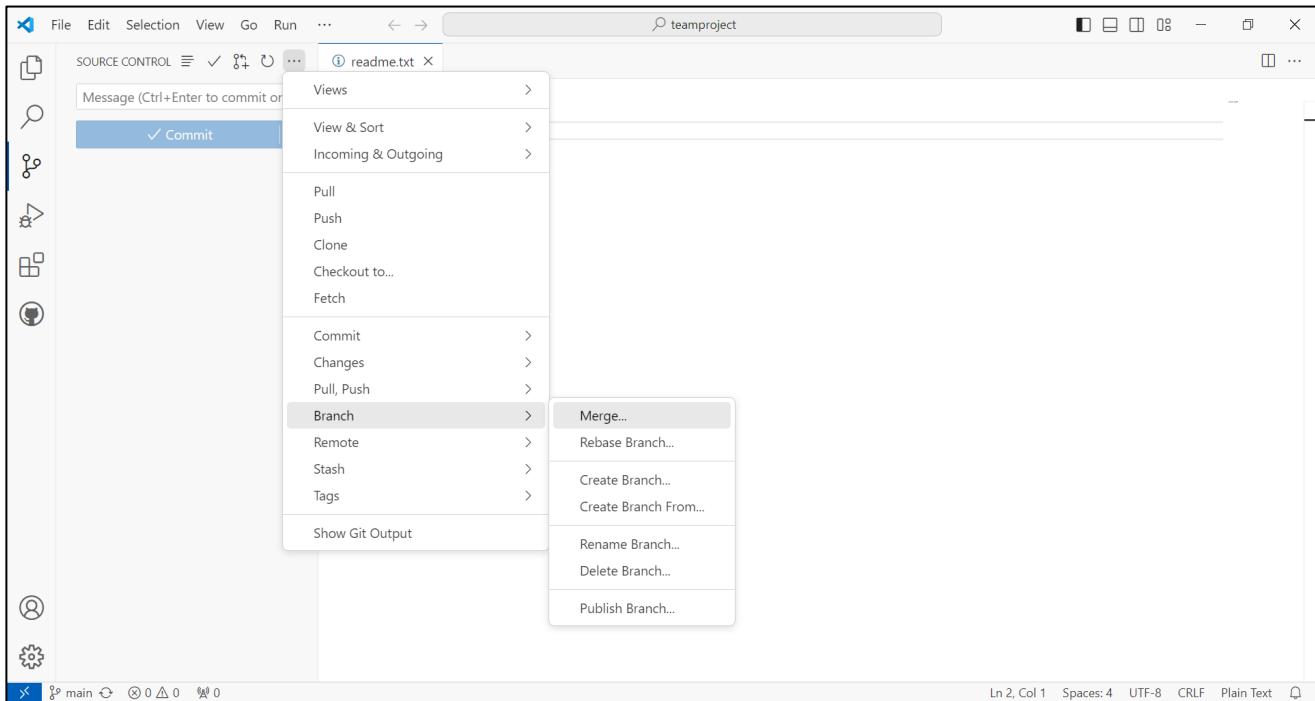


- A 44. Select the local repository branch **main**

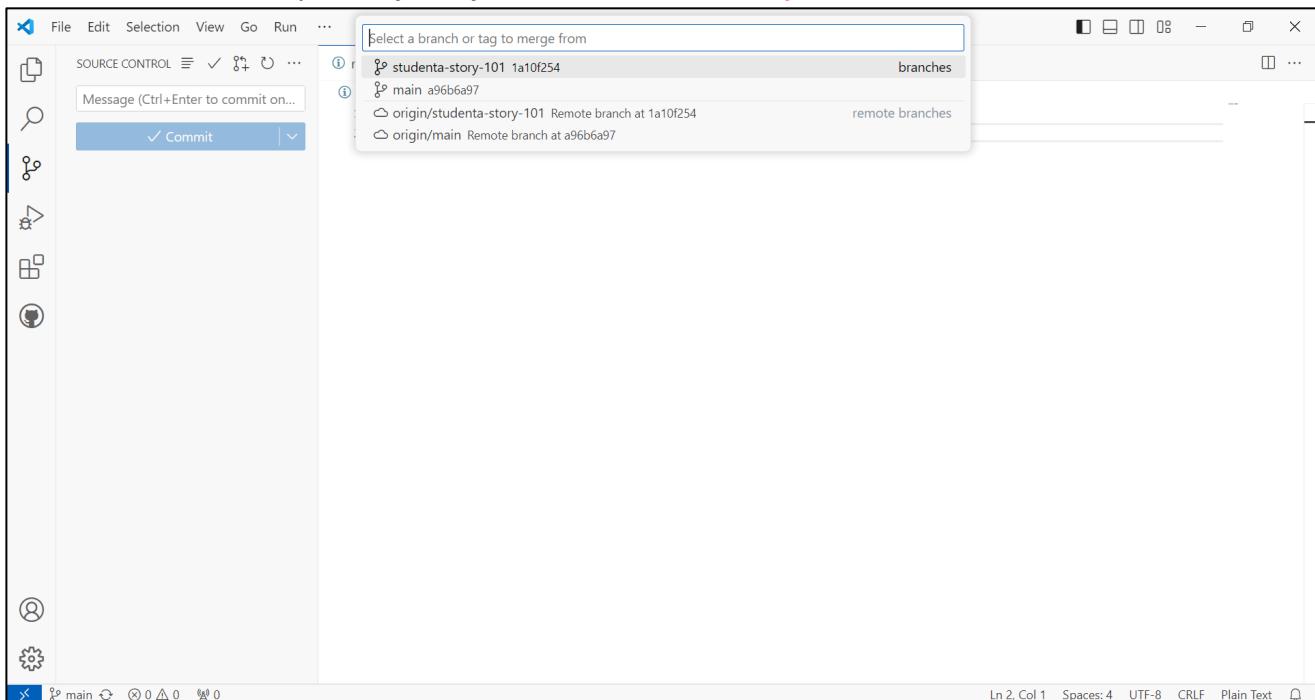


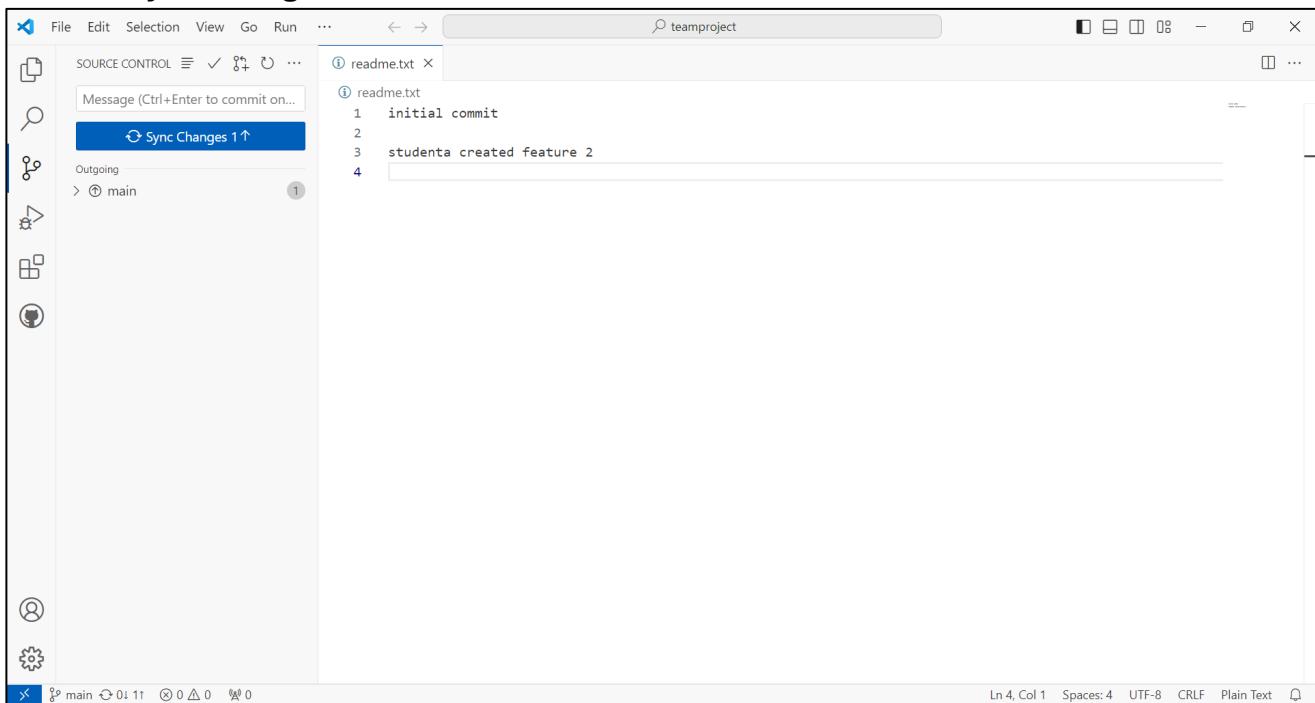
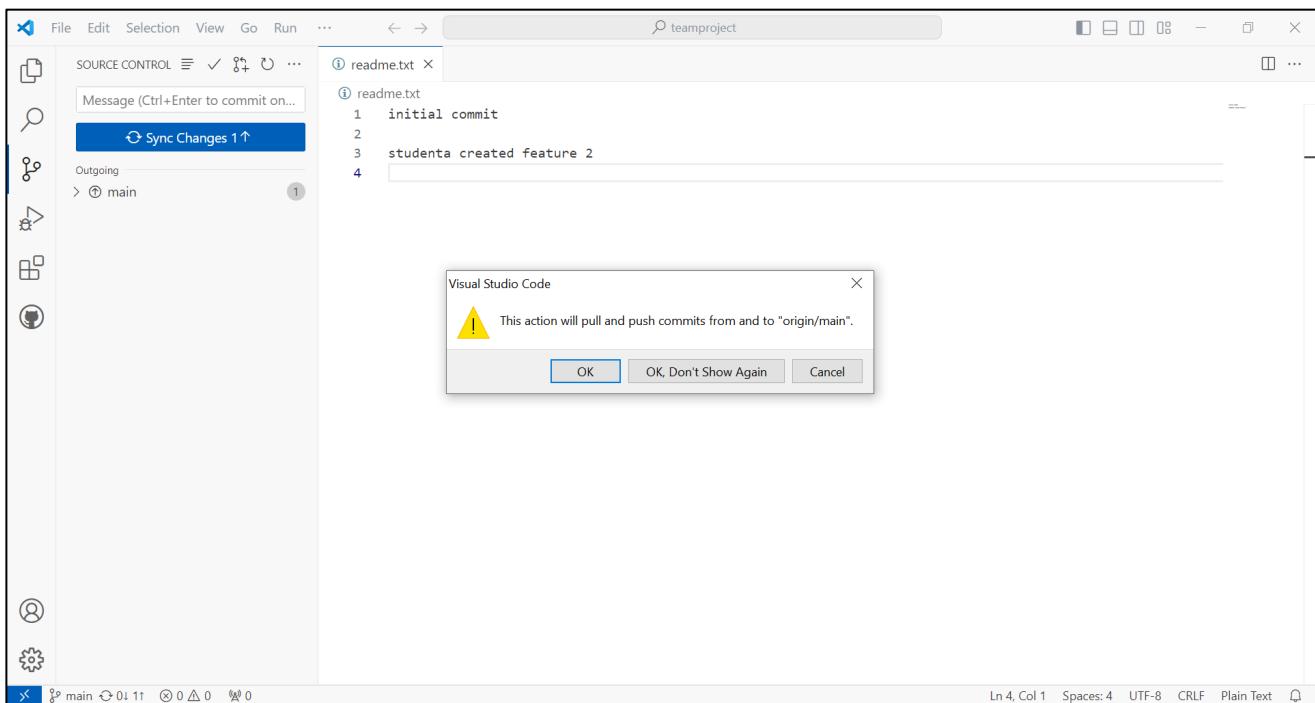
Merge branch

A 45. Click on the ellipsis, select **Branch, Merge..**



A 46. Select the local repository story branch **studenta-story-101**

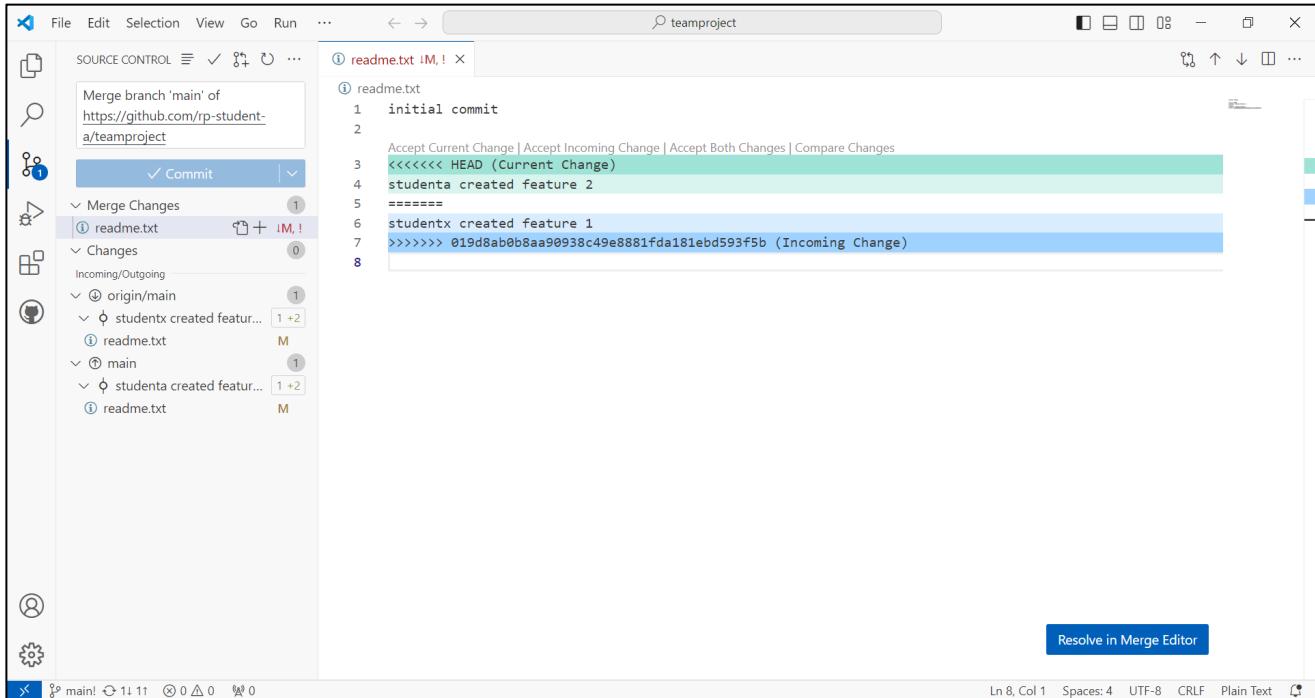


A 47. Click Sync Changes**A 48. Click OK**

Resolve conflicts

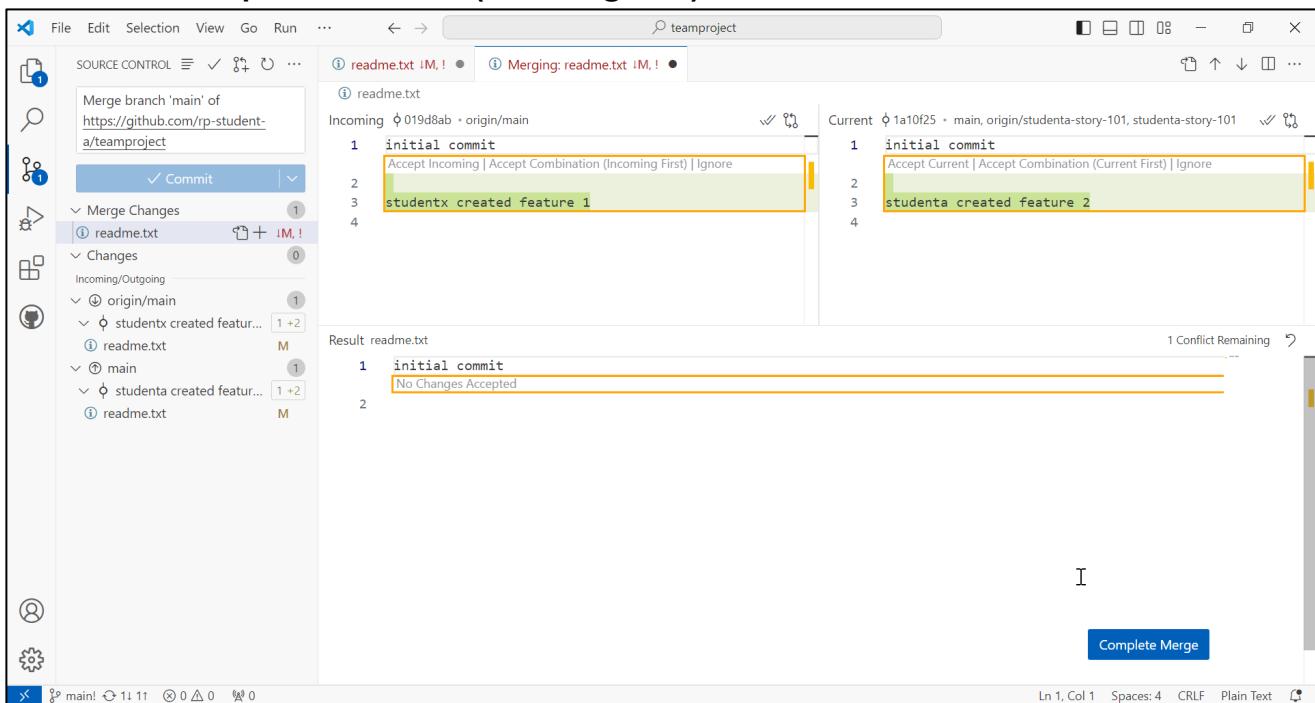
A 49. There is a conflict when trying to merge `readme.txt`

Click **Resolve in Merge Editor**



The screenshot shows the Visual Studio Code interface with the Source Control sidebar open. A merge conflict for the file `readme.txt` is displayed. The left pane shows the merge history: an initial commit from studenta, followed by studentx creating feature 2, and then studenta creating feature 1. The right pane shows the current state with studentx's changes (green highlights) and studenta's changes (blue highlights). A 'Resolve in Merge Editor' button is located at the bottom right of the editor area.

A 50. Click on **Accept Combination (Incoming First)**

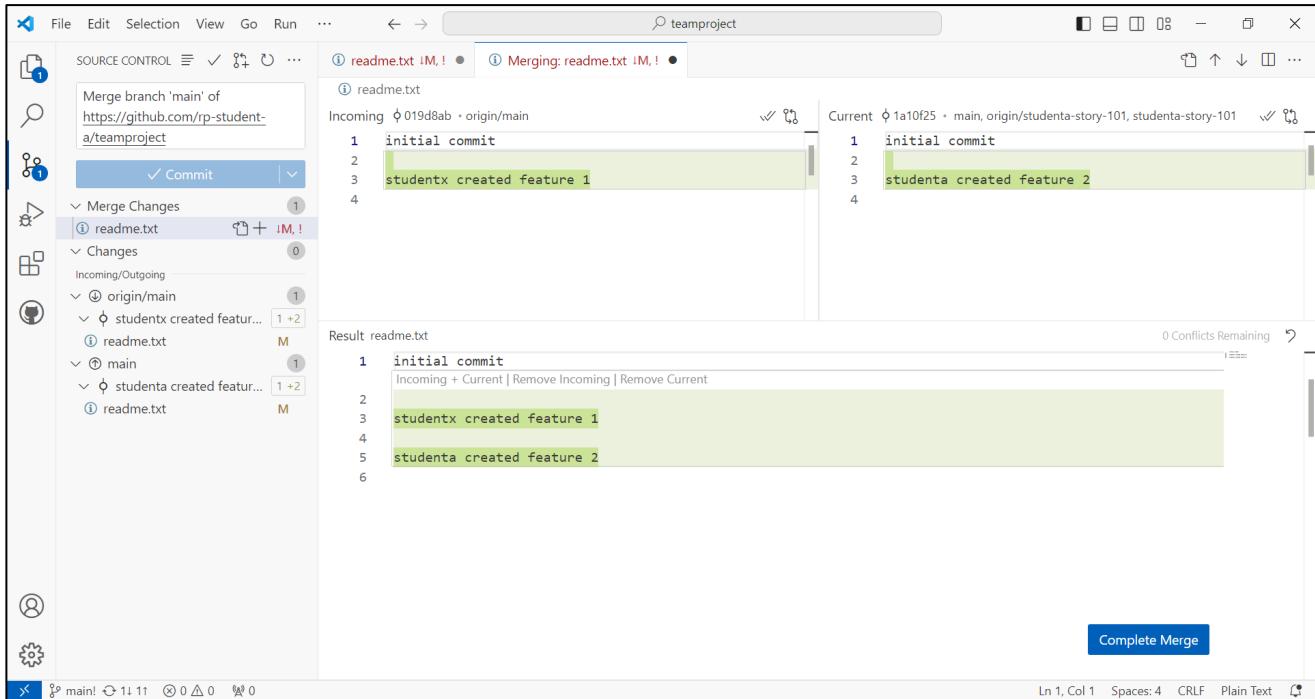


The screenshot shows the Visual Studio Code interface with the Source Control sidebar open. A merge conflict for the file `readme.txt` is displayed in the 'Merging' dialog. Both the 'Incoming' (studentx) and 'Current' (studenta) branches have their first 'initial commit' selected. In the 'Result' pane below, it shows 'No Changes Accepted'. A 'Complete Merge' button is located at the bottom right.

There are several options for how to resolve the merge - accept incoming, accept current, accept combination, ignore, etc, or you can manually edit the contents of the file in the bottom panel

When merging files with conflicts, talk to the person who did the previous merge for that file to understand the changes and what the final merged version should contain

A 51. Check that the resultant `readme.txt` looks as shown below then click **Complete Merge**

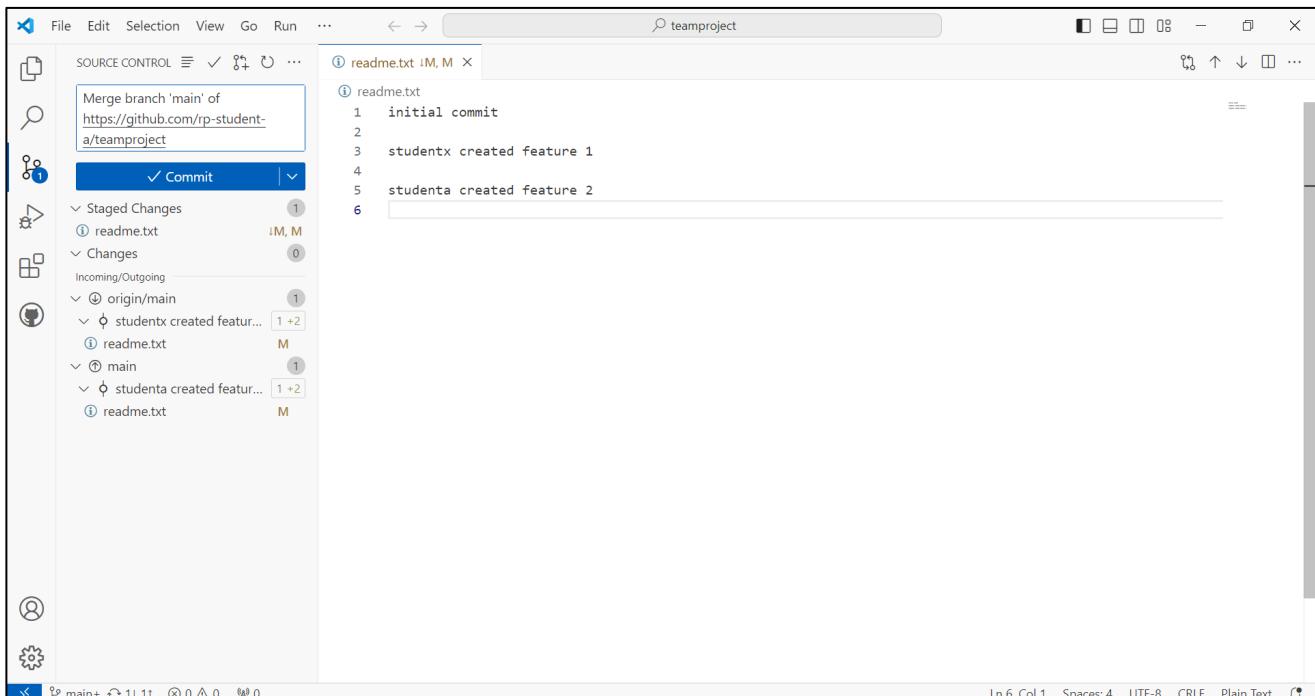


```

readme.txt IM, M
Initial commit
1 studentx created feature 1
2 studenta created feature 2
3
4
5
6

```

A 52. Click **Commit**

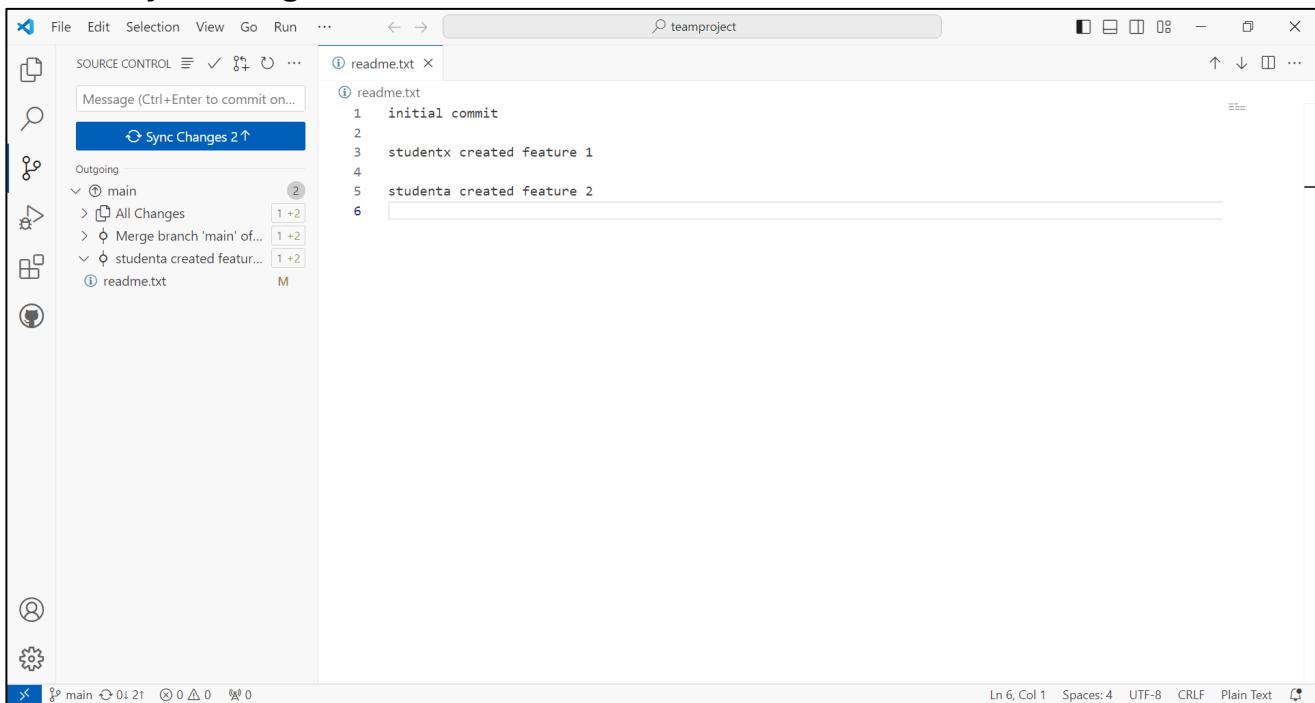


```

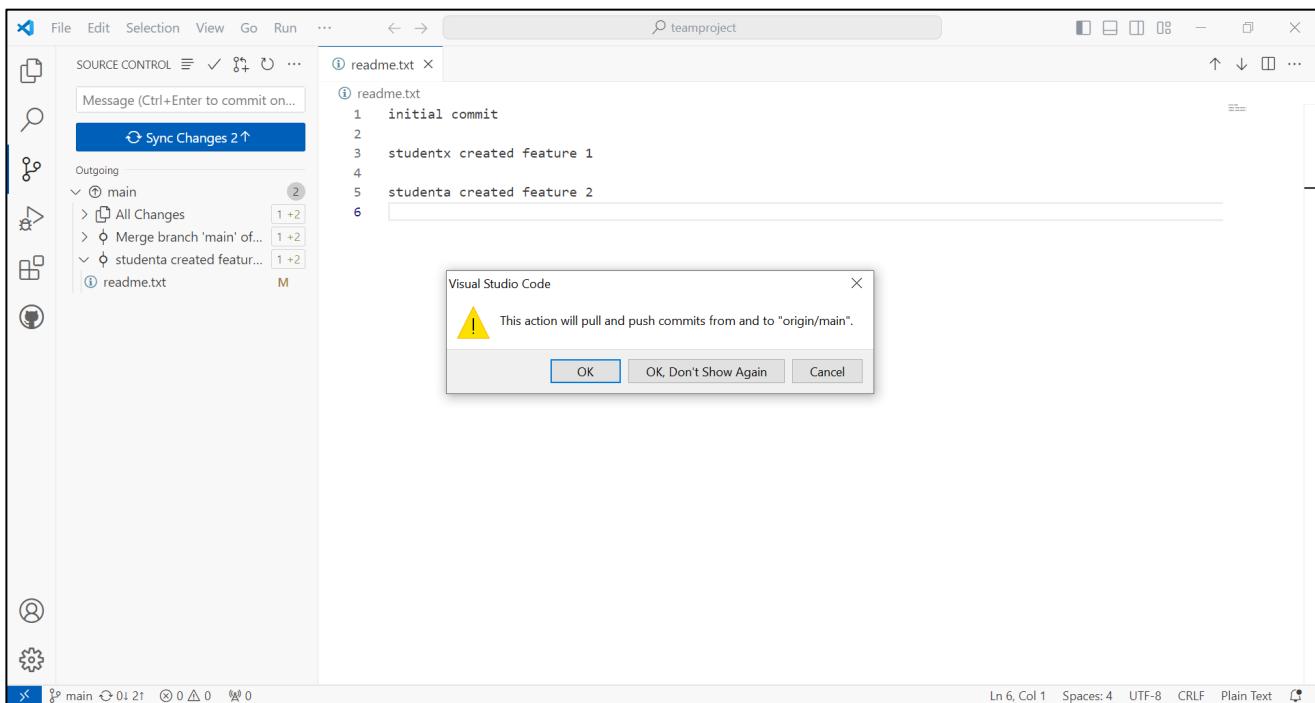
readme.txt IM, M
Initial commit
1 studentx created feature 1
2 studenta created feature 2
3
4
5
6

```

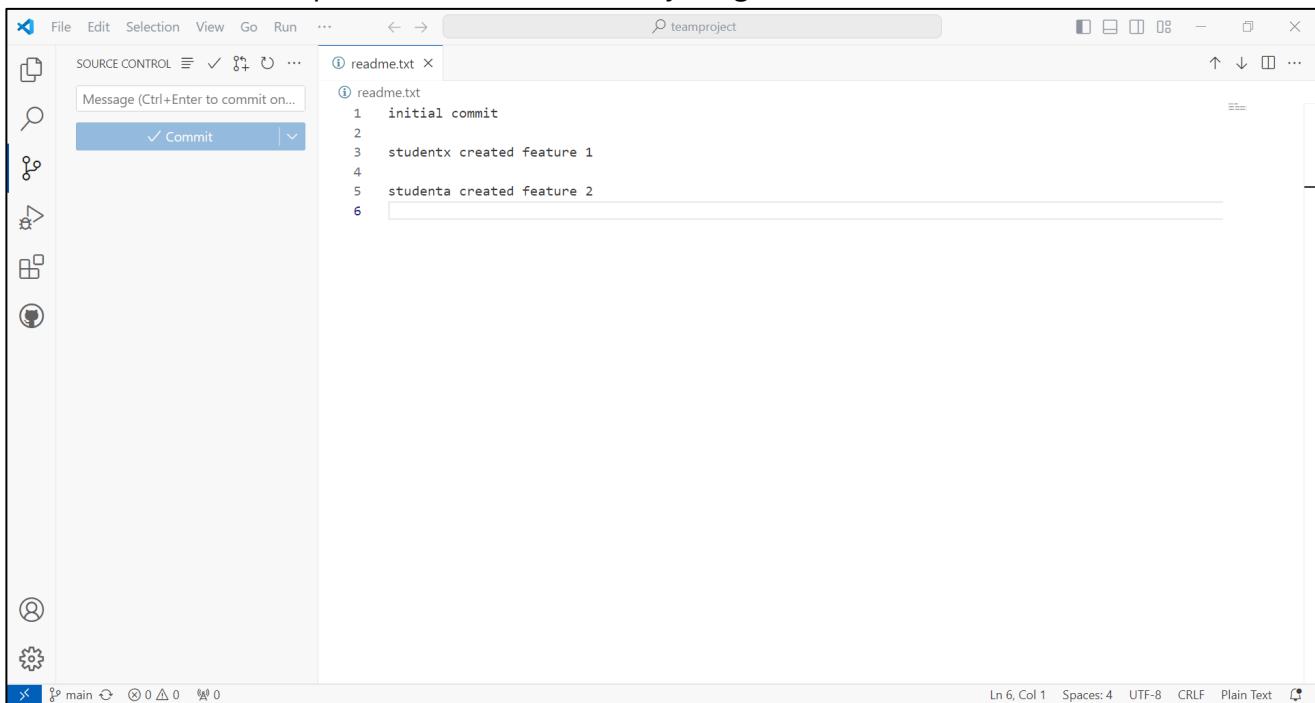
A 53. Click Sync Changes



A 54. Click OK



A 55. The source control panel should not show anything of note



The screenshot shows the Visual Studio Code interface with the Source Control panel open. The commit history for the file 'readme.txt' is displayed:

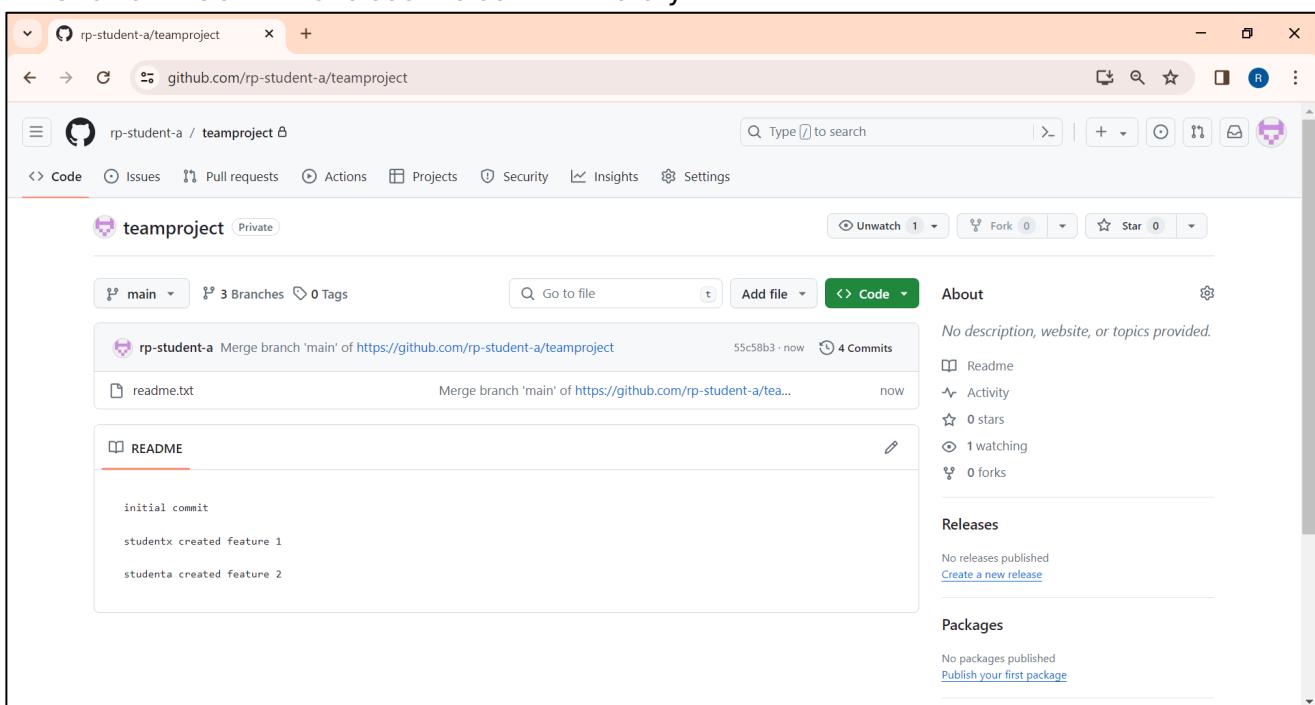
```

① readme.txt x
① readme.txt
1 initial commit
2
3 studentx created feature 1
4
5 studenta created feature 2
6

```

The commit message for the first commit is "initial commit". The second commit was made by "studentx" and the third by "studenta", both adding features. The commit history is currently at line 6.

A 56. Check the project in GitHub in a web browser
Click on **4 Commits** to see the commit history



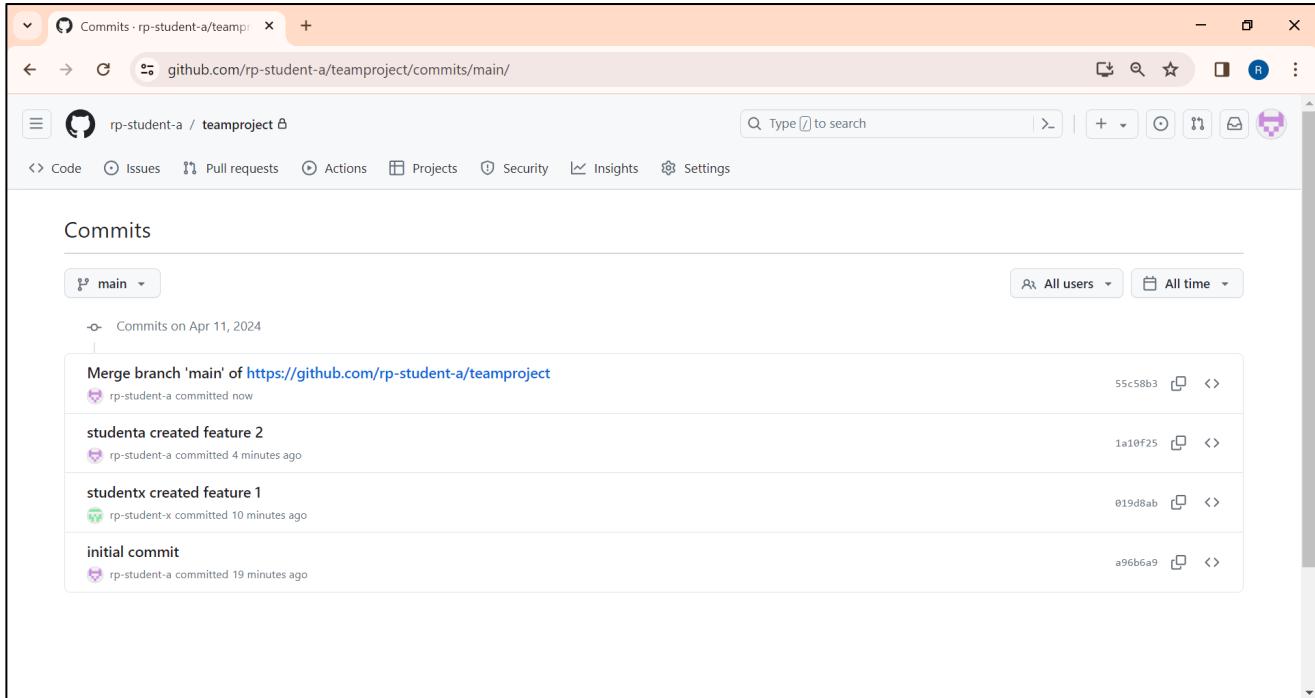
The screenshot shows a GitHub repository page for 'rp-student-a / teamproject'. The repository is private. The commit history for the 'main' branch is shown, with 4 commits listed:

- rp-student-a Merge branch 'main' of https://github.com/rp-student-a/teamproject 55c58b3 · now 4 Commits
- readme.txt Merge branch 'main' of https://github.com/rp-student-a/tea... now
- README
- initial commit
- studentx created feature 1
- studenta created feature 2

The commit history section includes a link to 'Create a new release' under the 'Releases' heading.

A 57. Check the project in GitHub in a web browser

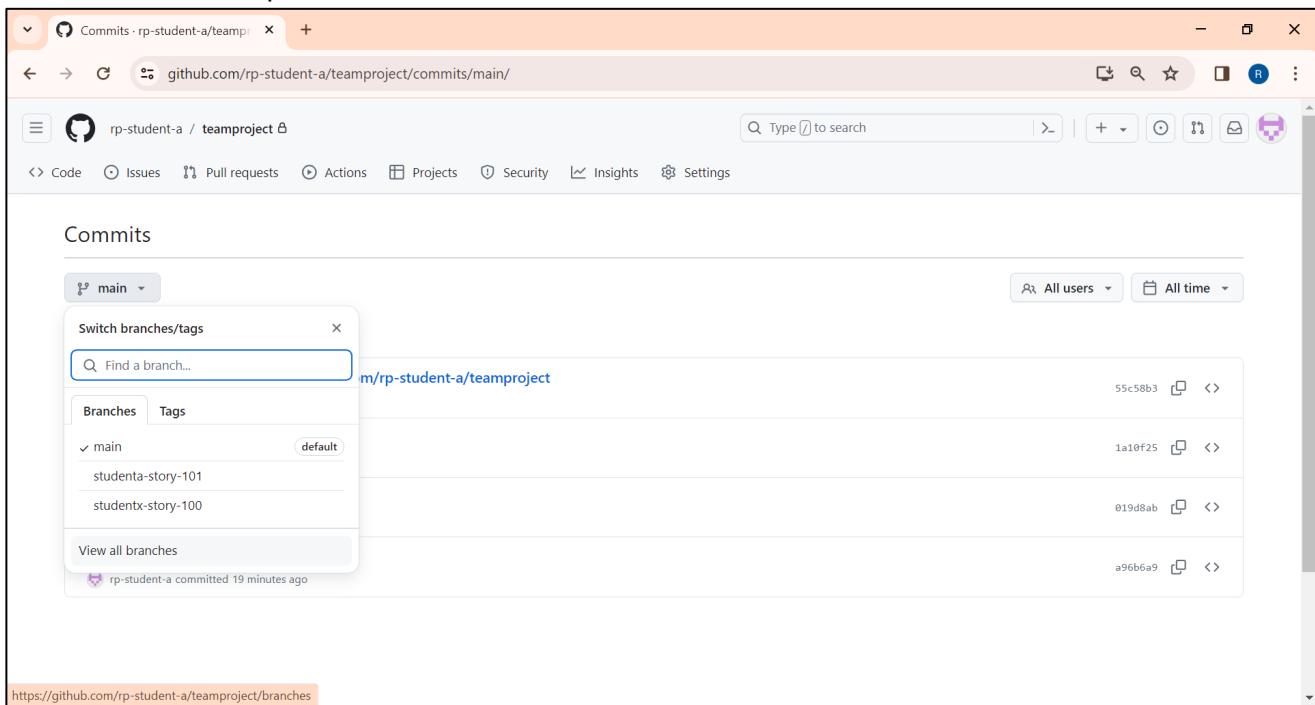
The commit messages are shown here and describe the reason for the commit, hence the need for meaningful commit message



The screenshot shows the GitHub 'Commits' page for the repository 'rp-student-a/teamproject'. The main branch is 'main'. The commits listed are:

- Merge branch 'main' of <https://github.com/rp-student-a/teamproject> by rp-student-a committed now. Commit hash: 55c58b3
- studenta created feature 2 by rp-student-a committed 4 minutes ago. Commit hash: 1a10f25
- studentx created feature 1 by rp-student-x committed 10 minutes ago. Commit hash: 019d8ab
- initial commit by rp-student-a committed 19 minutes ago. Commit hash: a96b6a9

A 58. Click on the dropdown next to main to see all the branches. Click **View all branches**

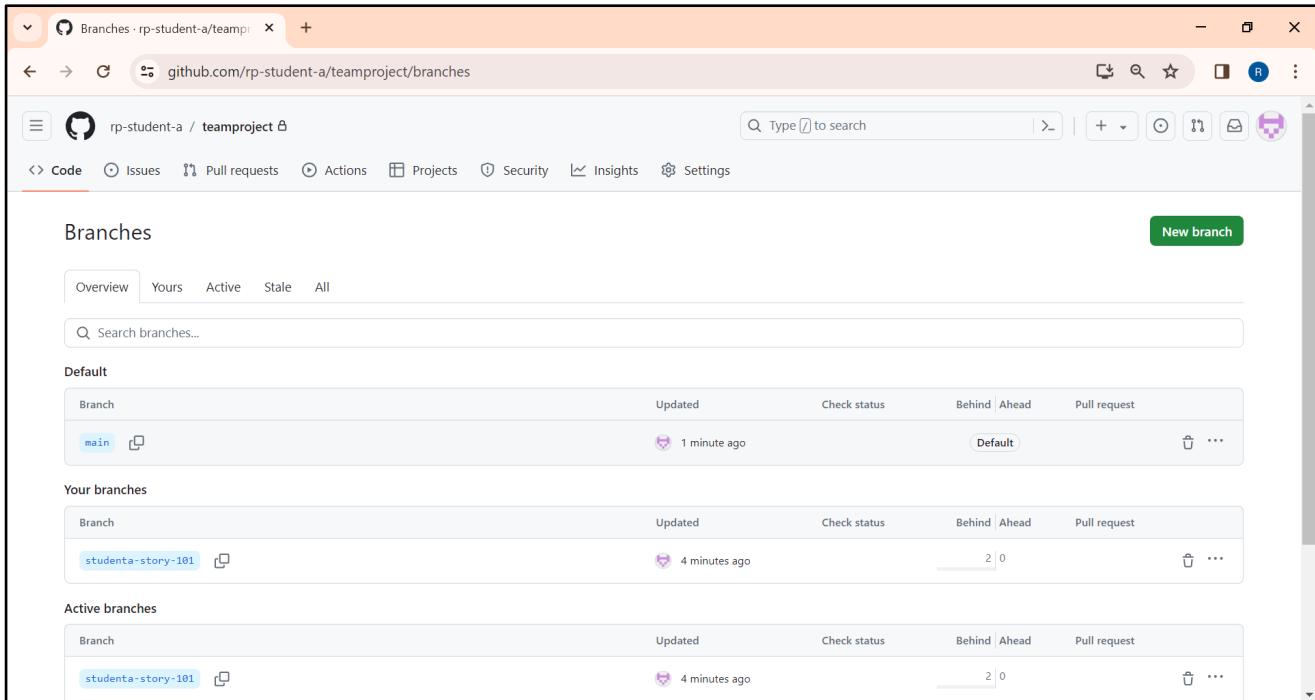


The screenshot shows the GitHub 'Commits' page for the repository 'rp-student-a/teamproject'. The dropdown menu next to 'main' is open, showing the following branches and tags:

- main (selected)
- studenta-story-101
- studentx-story-100
- View all branches

Below the dropdown, there is a link: <https://github.com/rp-student-a/teamproject/branches>

- A 59. On this page, you can see the branches for this repository. A best practice once the code from a branch has been committed to main is to delete the branch. The commit history for that branch would still exists in the main branch.



Branch	Updated	Check status	Behind Ahead	Pull request
main	1 minute ago	Default		⋮

Branch	Updated	Check status	Behind Ahead	Pull request
studenta-story-101	4 minutes ago		2 0	⋮

Branch	Updated	Check status	Behind Ahead	Pull request
studenta-story-101	4 minutes ago		2 0	⋮

End of exercise