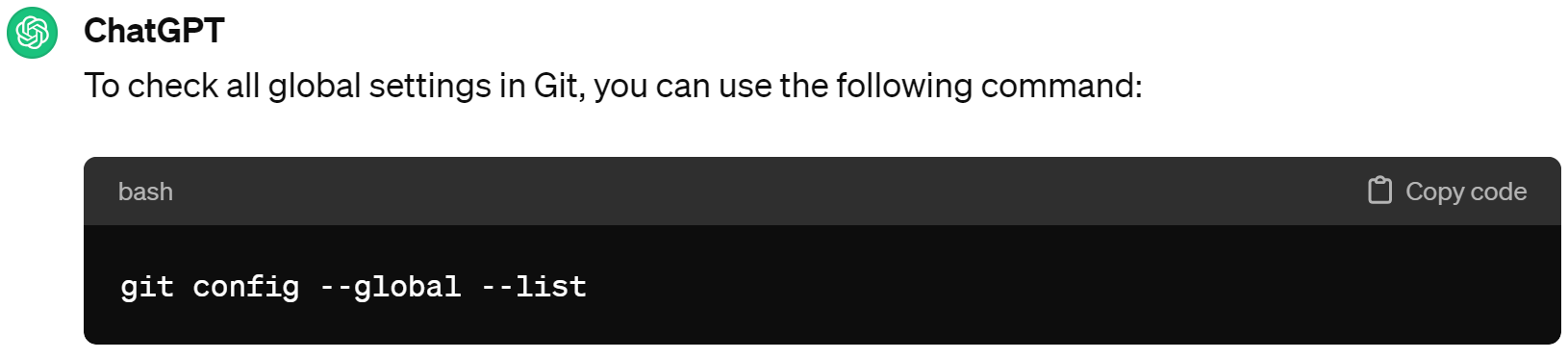
GIT:

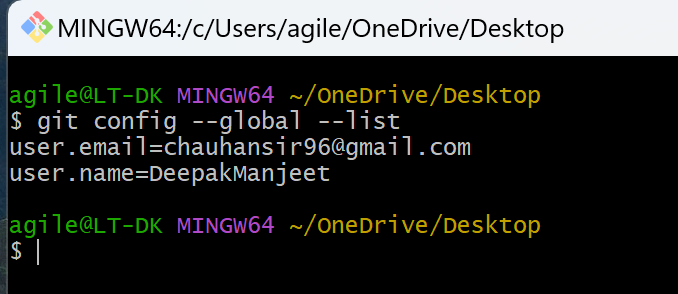
Parallel development, track every change

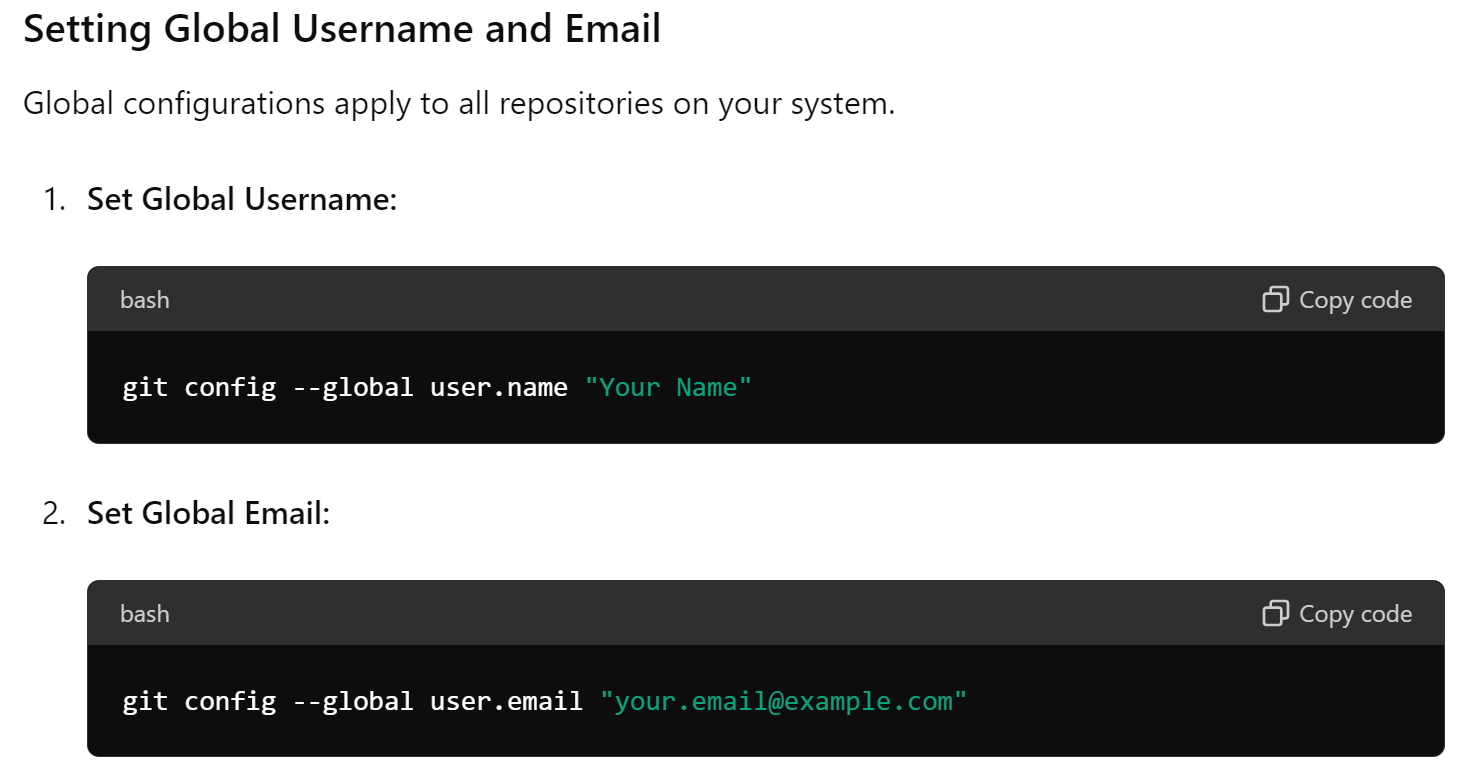
Distributed version control system

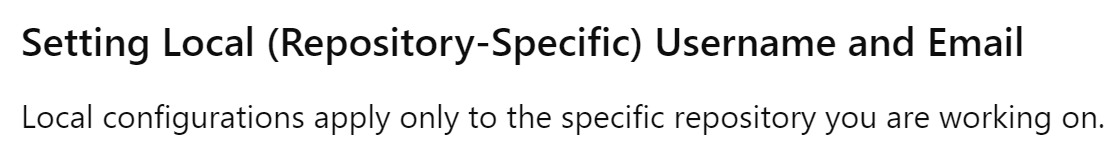
Working directory – staging area – local repo – remote repo



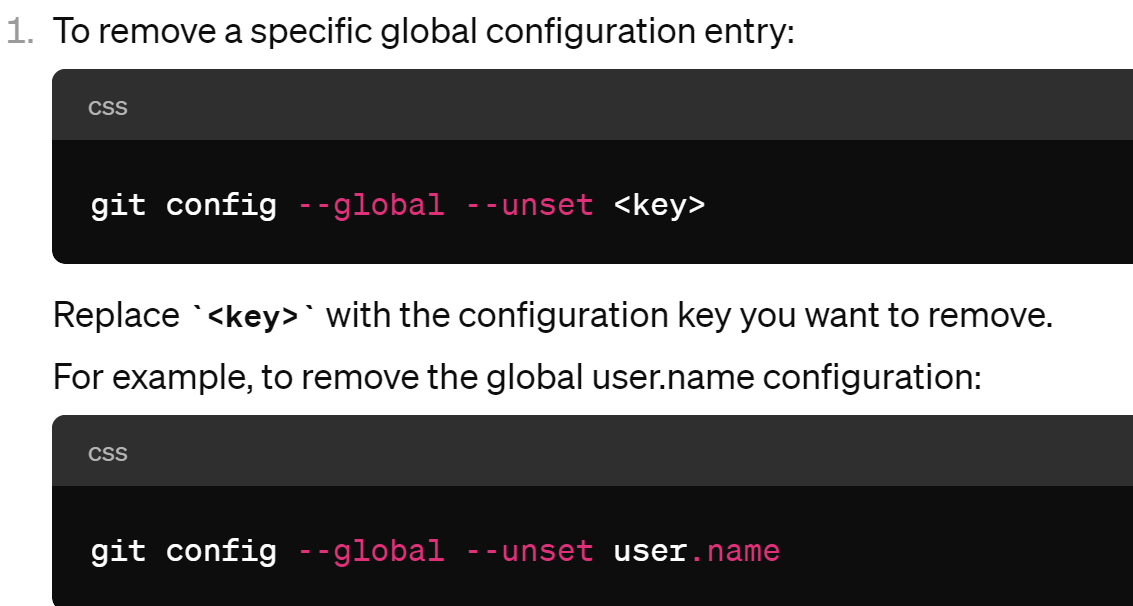
Ex











**Git steps to use in FIS environment:**

1. Apply proxy

git config --global https.proxy <https://e#:pswd@proxy.fnfis.com:8080>

git config --global http.proxy http://e#:pswd@proxy.fnfis.com:8080

* In fisdev.local environment like TA8 above lines are OK, but for client.local environment like WEBSTER, CFC:

**Proxy.fnfis.com** should be replaced by **proxy.prod.local**

|  |  |
| --- | --- |
| **In fisdev.local environment like TA8** | **in client.local environment like WEBSTER, CFC** |
| git config --global https.proxy <https://e#:pswd@proxy.fnfis.com:8080>  git config --global http.proxy <http://e#:pswd@proxy.fnfis.com:8080>  in above lines:  e# = e5721153  pswd = P@ssw0rd@9992284742 (means pswd of system)  P@ssw0rdSy$tem#68  P%40ssw0rdSy%24tem%2368 | git config --global https.proxy <https://e#:pswd@proxy.prod.local:8080>  git config --global http.proxy <http://e#:pswd@proxy.prod.local:8080>  in above lines:  e# = e5721153  pswd = Vista#24Automation (means admin pswd of server) |

Proxy can not be resolved if Password contains special character, so you have to replace those special characters with corresponding % symbol e.g.

1. **Space**: %20
2. **Exclamation mark (!)**: %21
3. **Double quote (")**: %22
4. **Hash or pound sign (#)**: %23
5. **Dollar sign ($)**: %24
6. **Percent sign (%)**: %25
7. **Ampersand (&)**: %26
8. **Single quote (')**: %27
9. **Left parenthesis (()**: %28
10. **Right parenthesis ())**: %29
11. **Asterisk (\*)**: %2A
12. **Plus sign (+)**: %2B
13. **Comma (,)**: %2C
14. **Hyphen (-)**: %2D
15. **Period or full stop (.)**: %2E
16. **Forward slash (/)**: %2F
17. **Colon (:)**: %3A
18. **Semicolon (;)**: %3B
19. **Less than sign (<)**: %3C
20. **Equals sign (=)**: %3D
21. **Greater than sign (>)**: %3E
22. **Question mark (?)**: %3F
23. **Commercial at (@)**: %40
24. **Left square bracket ([)**: %5B
25. **Backslash (\)**: %5C
26. **Right square bracket (])**: %5D
27. **Caret (^)**: %5E
28. **Underscore (\_)**: %5F
29. **Grave accent (`)**: %60
30. **Left curly brace ({)**: %7B
31. **Vertical bar or pipe (|)**: %7C
32. **Right curly brace (})**: %7D
33. **Tilde (~)**: %7E

Thus,

git config --global https.proxy [https://e5721153:P**%40**ssw0rd**%40**9992284742@proxy.fnfis.com:8080](https://e5721153:P%40ssw0rd%409992284742@proxy.fnfis.com:8080)

git config --global http.proxy [http://e5721153:P**%40**ssw0rd**%40**9992284742@proxy.fnfis.com:8080](http://e5721153:P%40ssw0rd%409992284742@proxy.fnfis.com:8080)

1. After setting these proxy, if you clone the project, a window dialog will pop up asking for your bitbucket credentials:

Username = e5721152

Pswd = P@ssw0rd@9992284742

1. You can check already set proxy details:

git config --global --get http.proxy

git config --global --get https.proxy

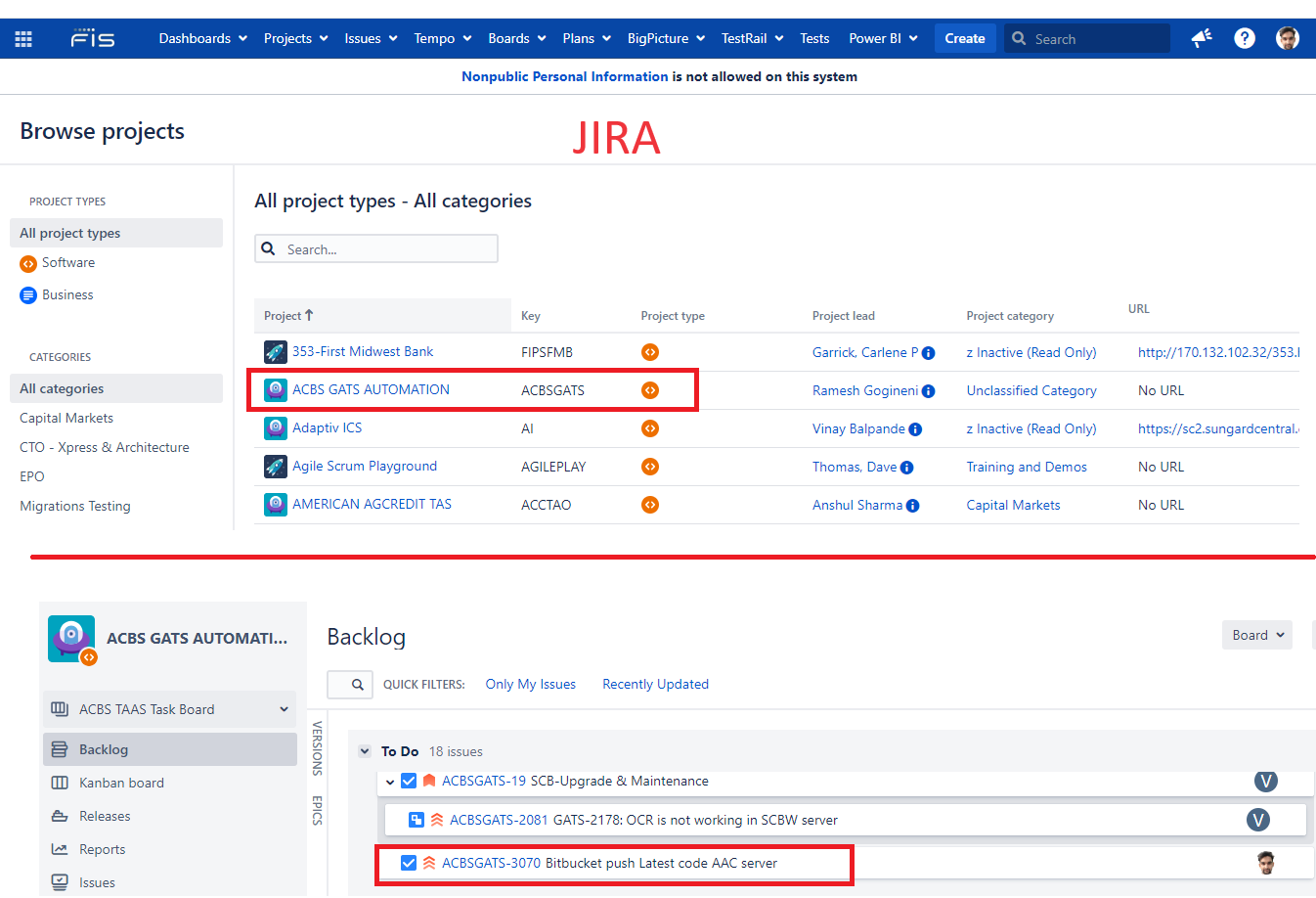
1. You can remove already set proxy settings:

git config --global --unset http.proxy

git config --global --unset https.proxy

**Steps to push code to Bitbucket and merge to master:**

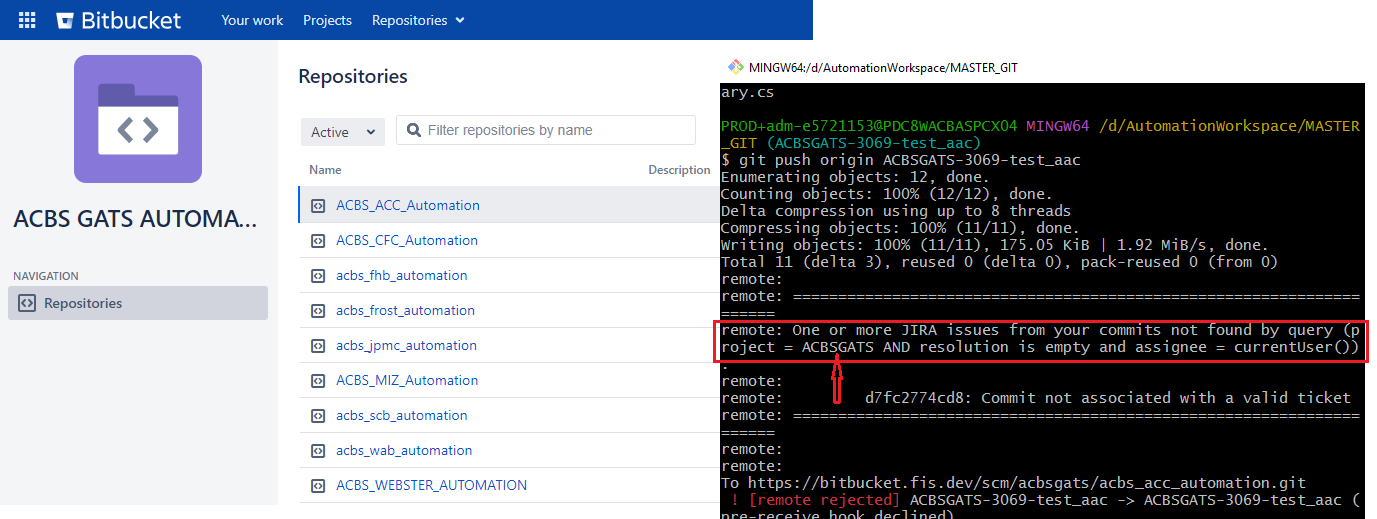
1. Create a Jira ticket under ACBS GATS AUTOMATION project:



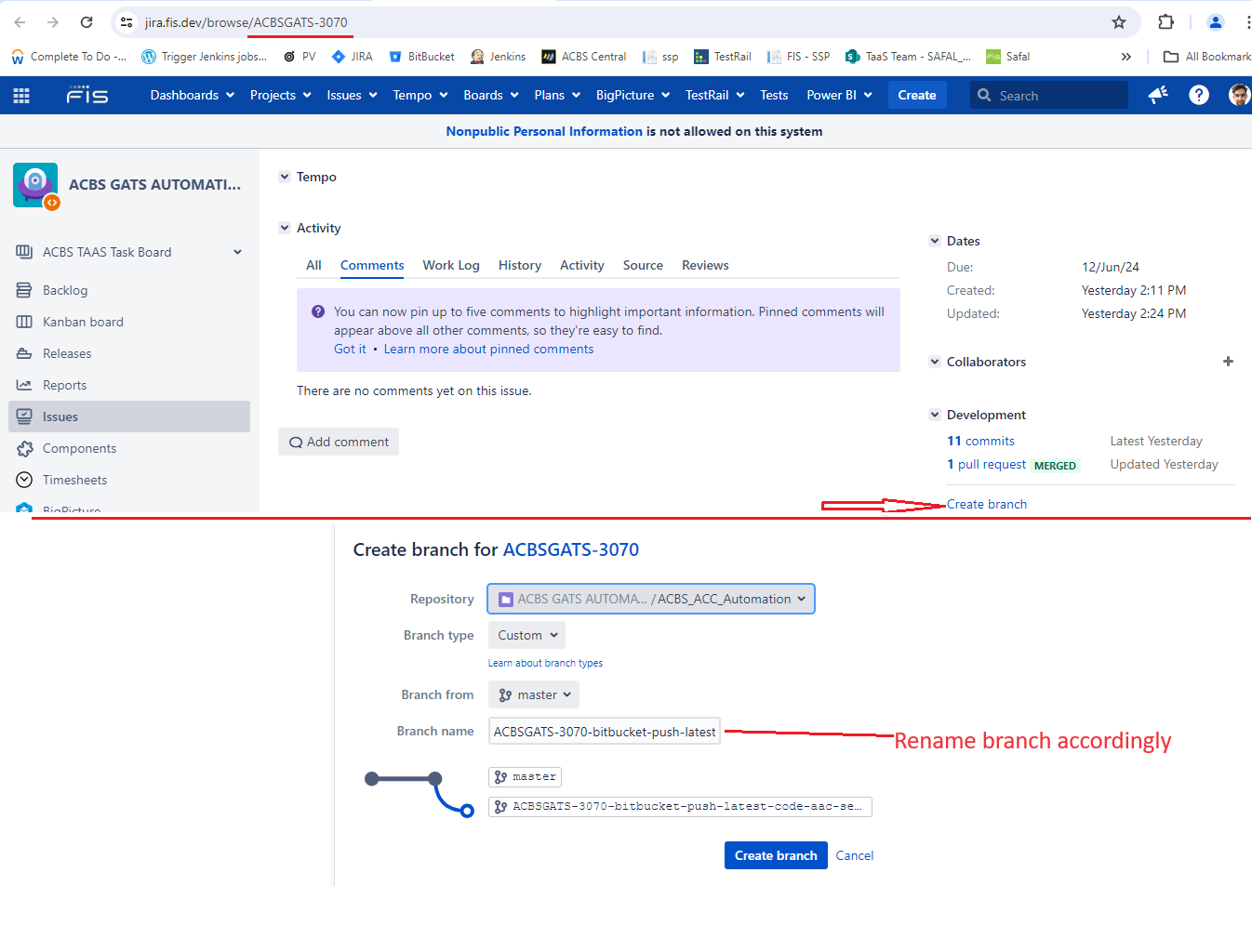
Under this ticket, we will create a branch

We have created this ticket under ACBS GATS AUTOMATION because all our repos are under this project and while pushing code to bitbucket, it executes a query to check for this project name else it will throw error as shown below in right image.

You may change this query by contacting repo admin of bitbucket.

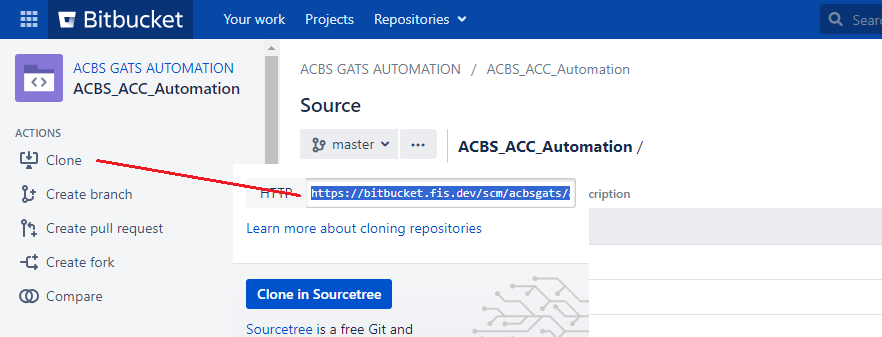


1. Create a branch under this ticket and switch to that branch using git bash on asp server:



1. Now, on the server, in git bash:

* Open git bash in your working directory and clone using master repo link from bitbucket



**git clone above\_link**

* fetch all refs in git on server:

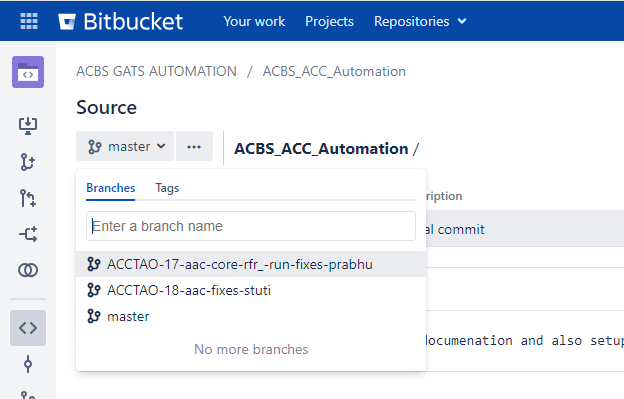
**git fetch --all**

* switch to branch created in jira ticket.

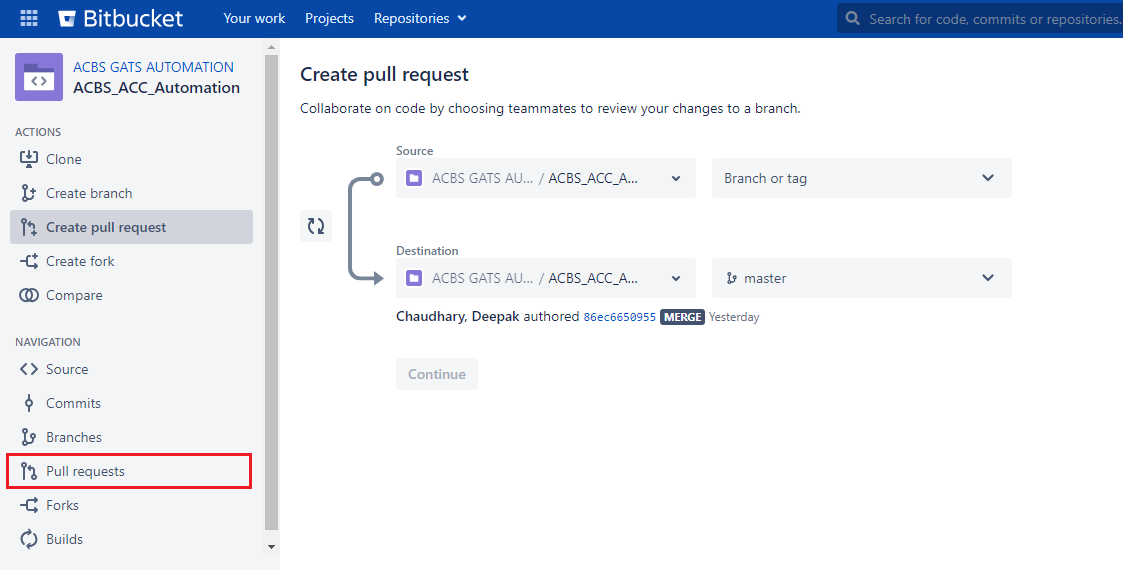
**git checkout branch\_name**

* Add your files/folder to your working directory where you have opened git bash
* git add .
* git commit -m "ticket\_name:description about commit "
* git push
* (After pushing one user WS, you may copy other user WS and replace files in that git working directory and follow these above 5 steps again)

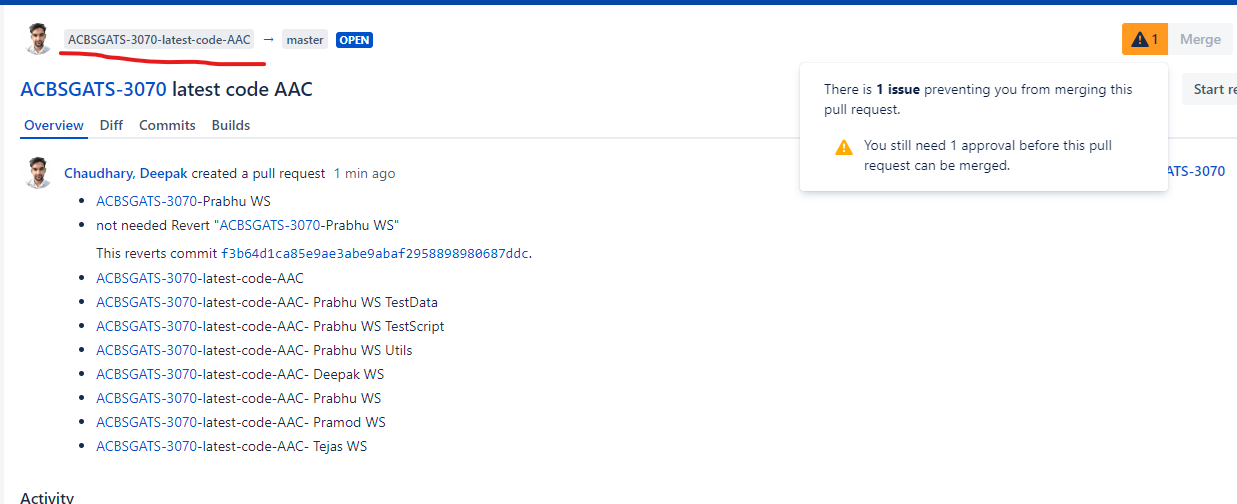
Now, your complete code is available in bitbucket, attached to branch created.



You have to create a pull request to merge this branch to master



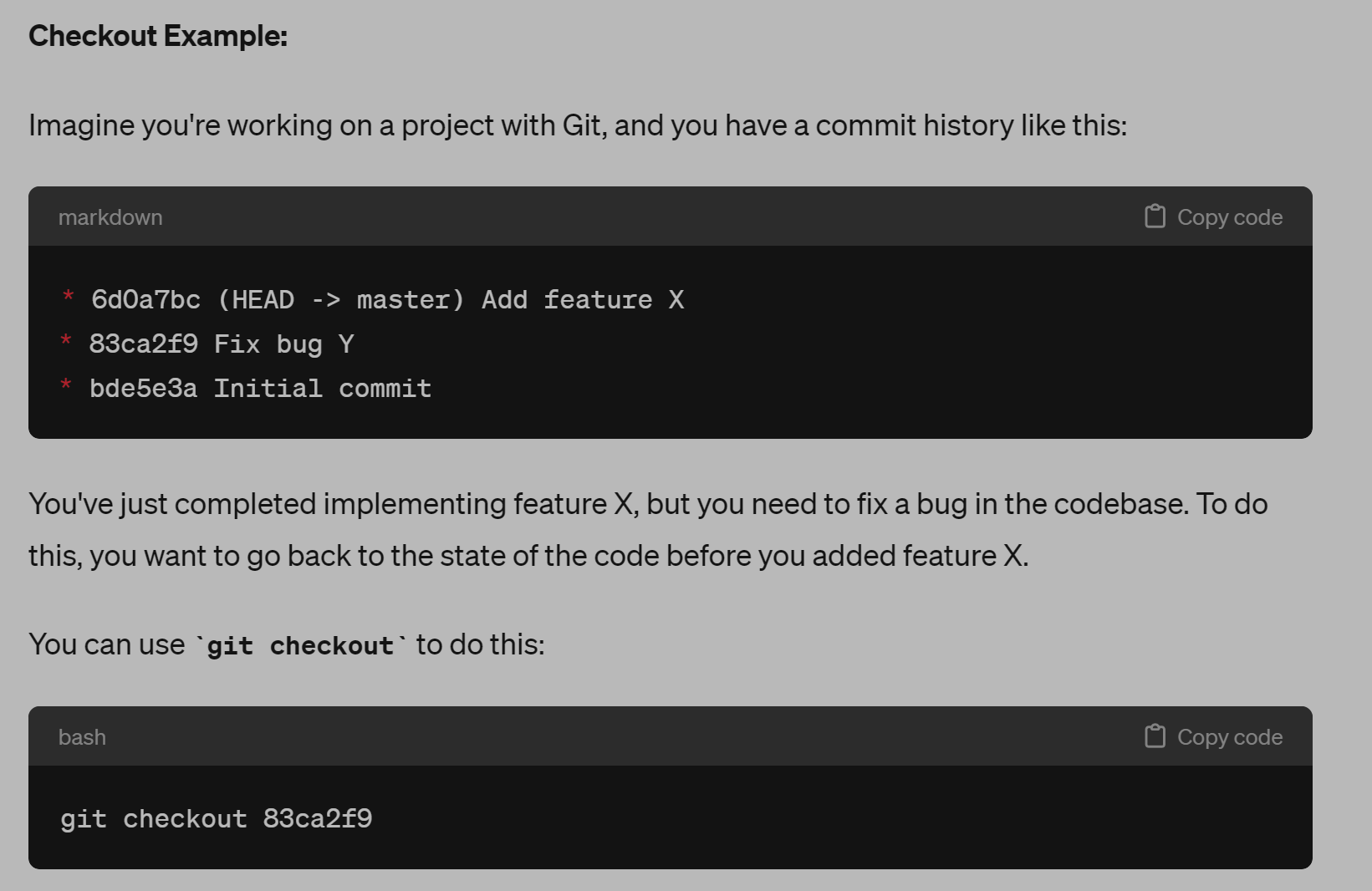
Once any team member approves that pull request, you can merge the branch to master.

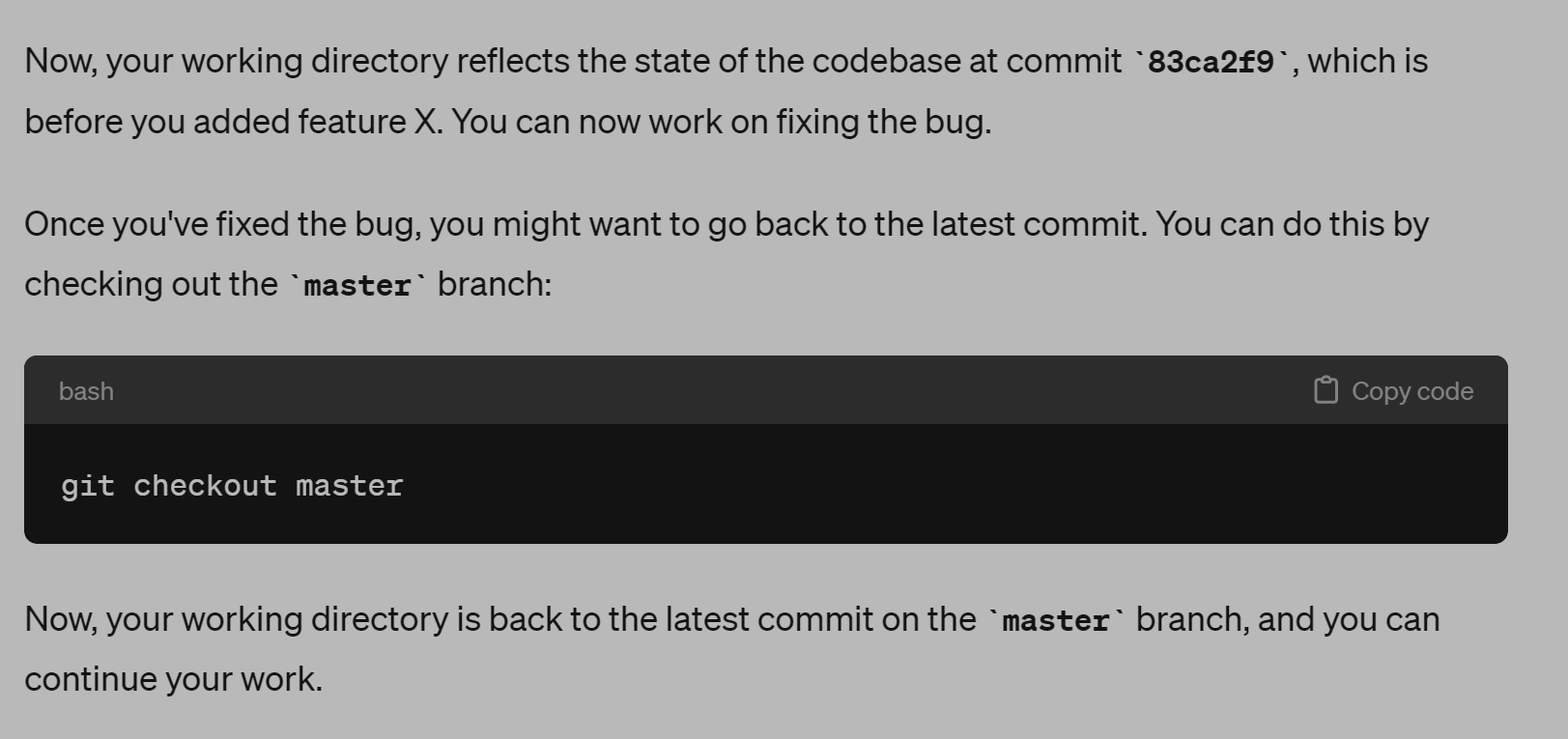


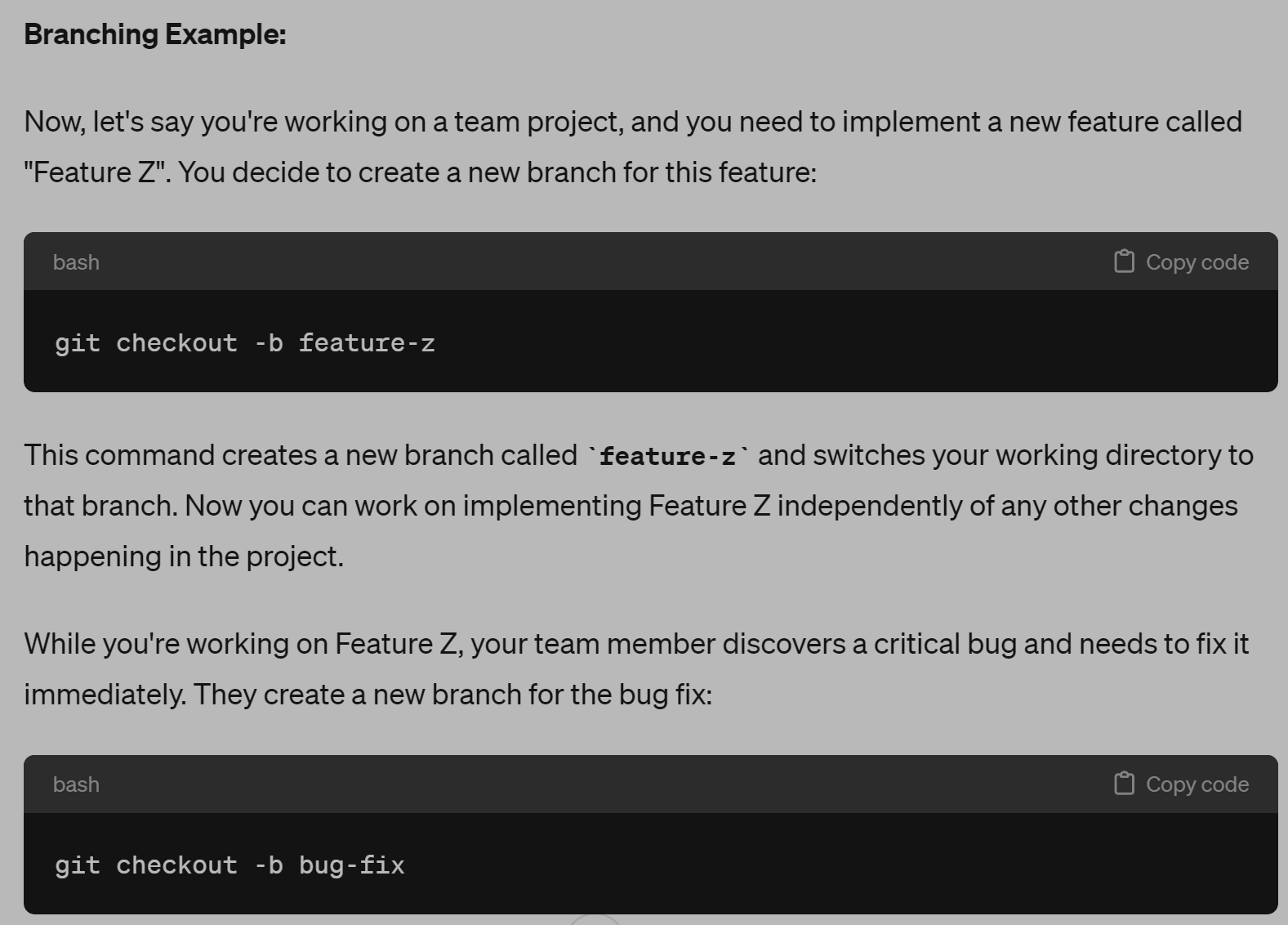
**Difference between Checkout and Branching**

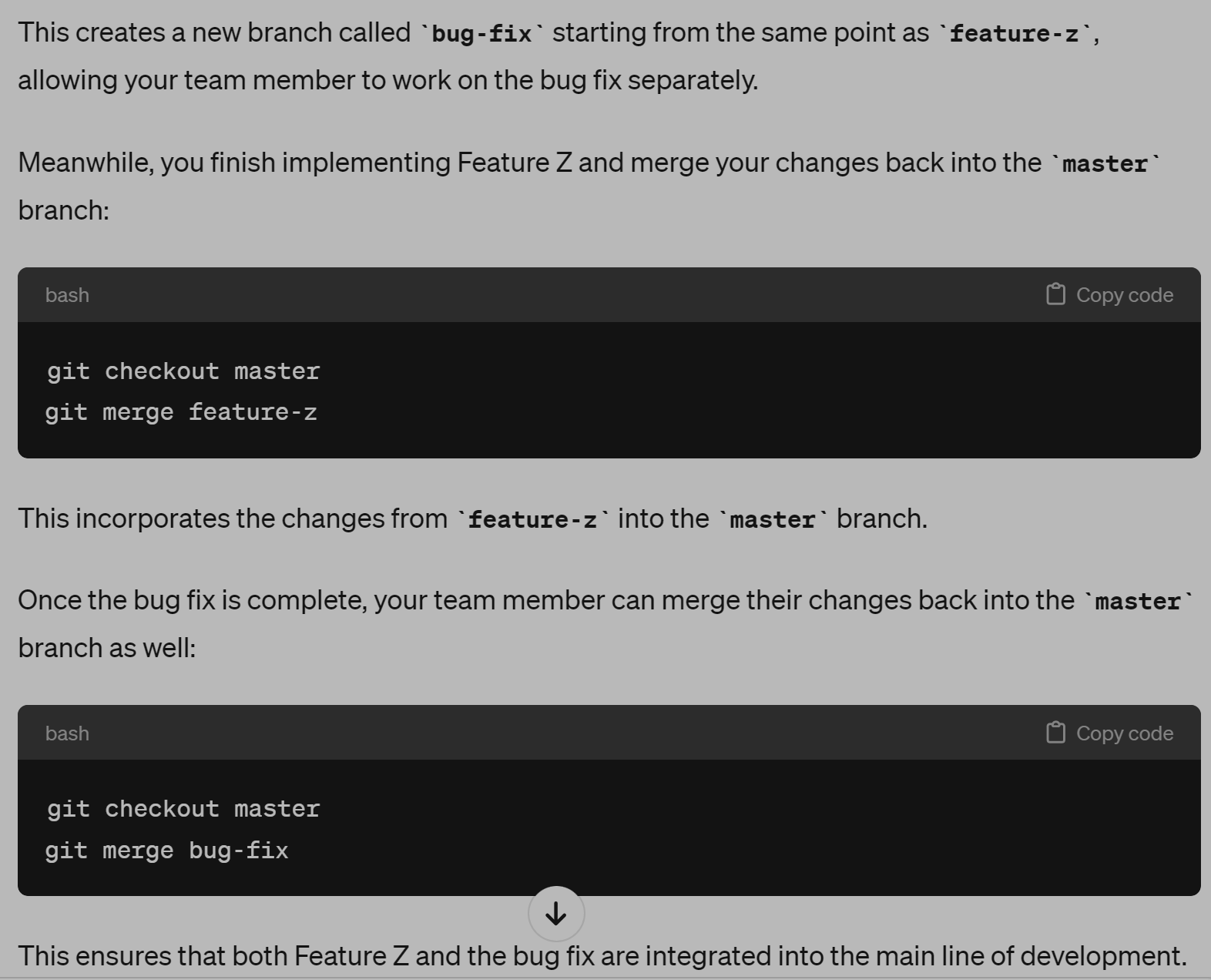
1. **Checkout**:
   * Checkout refers to the act of switching between different versions of a codebase.
   * It allows you to move your working directory to a specific state of the codebase, typically a specific commit or branch.
   * When you checkout a commit, you're essentially switching your working directory to reflect the state of the project at that particular commit. This can be useful for inspecting past states of the project, debugging, or reverting changes.
   * You can also checkout branches to switch between different lines of development in your project.
2. **Branching**:
   * Branching involves creating a divergent line of development from a specific point in the version history.
   * It allows multiple developers to work on different features or fixes simultaneously without interfering with each other's work.
   * When you create a branch, you're essentially creating a separate "timeline" of changes that can be merged back into the main line of development (usually the **master** branch) at a later stage.
   * Branches are lightweight and efficient in Git, making them a preferred method for managing parallel workstreams.

In summary, while both checkout and branching involve manipulating the state of a codebase, they serve different purposes: checkout is about navigating between different states of the codebase, while branching is about creating separate lines of development for parallel work.

****

****

****

****

**clear --- in git bash**

**cls ---in vs code**

**when accrual methodology is: include last day (in Addendum tab)**

Certainly! Here's the complete step-by-step process and the formula used to calculate the Flat Fee of 2358.90 SGD, including both the start date (08/08/2024) and the end date (29/09/2027).

* + End date comes from mail end period

Step 1: Determine the Loan Duration (in Days)

* Start Date: 08/08/2024
* End Date: 29/09/2027

To calculate the total loan duration including both the start and end dates, we first calculate the number of days between the two dates and then add 1 day to account for both days being included.

1. Initial Loan Duration (excluding end date): 1147 days.
2. Including Both Dates: Loan Duration=1147+1=1148 days.\text{Loan Duration} = 1147 + 1 = 1148 \, \text{days}.Loan Duration=1147+1=1148days.

Step 2: Loan Amount and All-In Rate

* Loan Amount: 25,000 SGD.
* All-In Rate: 3% per annum (Base Rate + Spread Rate + Reserve Rate).

The all-in rate is expressed as a percentage, which needs to be converted to a decimal for the formula:

All-In Rate=3100=0.03.\text{All-In Rate} = \frac{3}{100} = 0.03.All-In Rate=1003​=0.03.

Step 3: Formula to Calculate the Flat Fee (for Loan Duration in Days)

The formula to calculate the flat fee based on the loan duration in days is:

Flat Fee=Loan Amount×(All-In Rate100)×Loan Duration (in days)365\text{Flat Fee} = \text{Loan Amount} \times \left( \frac{\text{All-In Rate}}{100} \right) \times \frac{\text{Loan Duration (in days)}}{365}Flat Fee=Loan Amount×(100All-In Rate​)×365Loan Duration (in days)​

Step 4: Apply the Formula

Now, applying the values we have:

Flat Fee=25,000×0.03×1148365\text{Flat Fee} = 25,000 \times 0.03 \times \frac{1148}{365}Flat Fee=25,000×0.03×3651148​

Breaking it down:

Flat Fee=25,000×0.03×3.145=25,000×0.09435=2358.90 SGD.\text{Flat Fee} = 25,000 \times 0.03 \times 3.145 = 25,000 \times 0.09435 = 2358.90 \, \text{SGD}.Flat Fee=25,000×0.03×3.145=25,000×0.09435=2358.90SGD.

Final Result:

The Flat Fee is calculated as 2358.90 SGD for a loan amount of 25,000 SGD over a loan duration of 1148 days, with an all-in rate of 3%.

This is the complete step-by-step breakdown to reach the flat fee amount you were expecting.

4o

You said:

calculate flat fee if Start Date: 28/08/2024 End Date: 14/10/2027

ChatGPT said:

ChatGPT

For the new loan duration from 28/08/2024 to 14/10/2027 (including both dates), the total loan duration is 1143 days.

Using this duration, the calculated Flat Fee is 2348.63 SGD.