

# AI: Mastering the Shell with GitHub Copilot CLI

## Screenshot of Interaction

The screenshot shows a terminal window with the following text:

```
combine the last two local commits into one commit with a new message without affecting the remote branch.
Describe a task to get started.

Pick a model with /model. Send this session to GitHub with /delegate, and Copilot will create a PR. Enter ? to see all commands. Copilot uses AI, so always check for mistakes.

No copilot instructions found. Run /init to generate a copilot-instructions.md file for this project.

combine the last two local commits into one commit with a new message without affecting the remote branch
The user wants to combine the last two local commits into one commit with a new message without affecting the remote branch. This is a git operation that involves:
1. Identifying the last two commits
2. Using git rebase or git reset to squash them
3. Writing a new commit message
4. Not affecting the remote branch (so only local changes)

First, I need to check:
1. Is this a git repository?
2. What are the last two commits?
3. Has the remote been pushed?

Let me start by checking if we're in a git repository and what the current state is.

Check git status and current branch
$ git status
error: tool 'PowerShell-Gv' not found. Please install it from https://aka.ms/powershell. Error: Error: Command failed: push.exe --version
'push.exe' is not recognized as an internal or external command,
operable program or batch file.
?

Let me use the Windows PowerShell instead.

Check git status
$ powershell -Command "git status"
The user rejected this tool call. User feedback: Do not run any commands. Just give me the Git command to squash the last two local commits into one with a new commit message, without pushing.

To combine the last two local commits into one with a new message, use:
git rebase -i HEAD~2
This opens your editor. Replace pick with squash (or s) on the second commit line, save and exit. Then you'll be prompted to edit the new combined commit message.

Alternatively, use this one-liner:
git reset --soft HEAD~2 && git commit -m "Your new commit message"
This unstages the last 2 commits but keeps changes staged, then creates a new commit with your message.

Type @ to mention files or / for commands
shift+tab switch mode
Remaining reqs.: 0%
```

The screenshot shows a terminal window with the following text:

```
First, I need to check:
1. Is this a git repository?
2. What are the last two commits?
3. Has the remote been pushed?

Let me start by checking if we're in a git repository and what the current state is.

Check git status and current branch
$ git status
error: tool 'PowerShell-Gv' not found. Please install it from https://aka.ms/powershell. Error: Error: Command failed: push.exe --version
'push.exe' is not recognized as an internal or external command,
operable program or batch file.

Let me use the Windows PowerShell instead.

Check git status
$ powershell -Command "git status"
The user rejected this tool call. User feedback: Do not run any commands. Just give me the Git command to squash the last two local commits into one with a new commit message, without pushing.

To combine the last two local commits into one with a new message, use:
git rebase -i HEAD~2
This opens your editor. Replace pick with squash (or s) on the second commit line, save and exit. Then you'll be prompted to edit the new combined commit message.

Alternatively, use this one-liner:
git reset --soft HEAD~2 && git commit -m "Your new commit message"
This unstages the last 2 commits but keeps changes staged, then creates a new commit with your message.

Type @ to mention files or / for commands
shift+tab switch mode
```

## Accuracy Ranking and Justification:

The AI provided the correct Git strategy using interactive rebase or soft reset to combine unpushed commits and rewrite the commit message. The solution preserves the remote branch state and demonstrates a proper understanding of Git history rewriting.