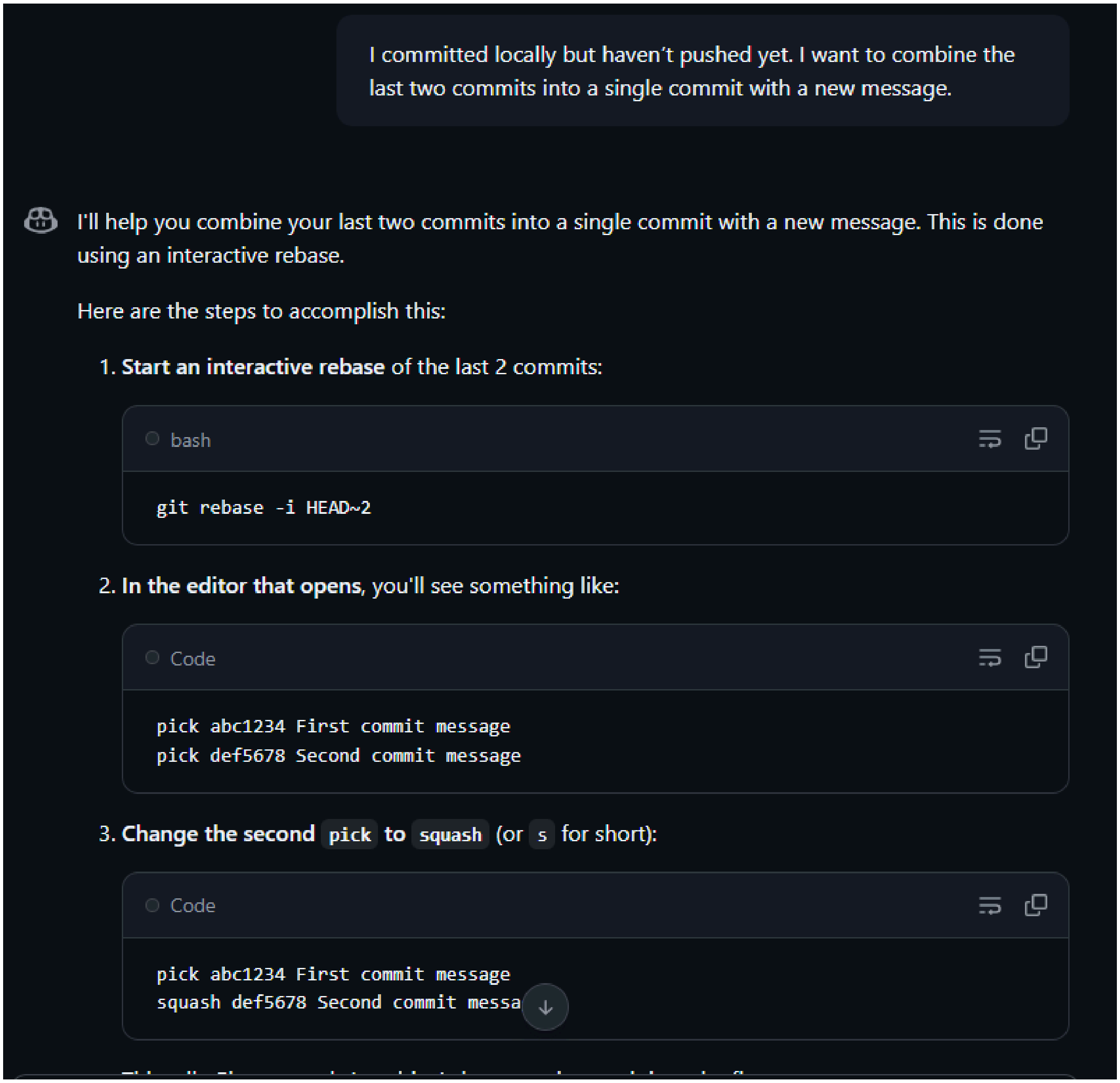


I committed locally but haven't pushed yet. I want to combine the last two commits into a single commit with a new message.

I'll help you combine your last two commits into a single commit with a new message. This is done using an interactive rebase.

Here are the steps to accomplish this:

1. **Start an interactive rebase** of the last 2 commits:

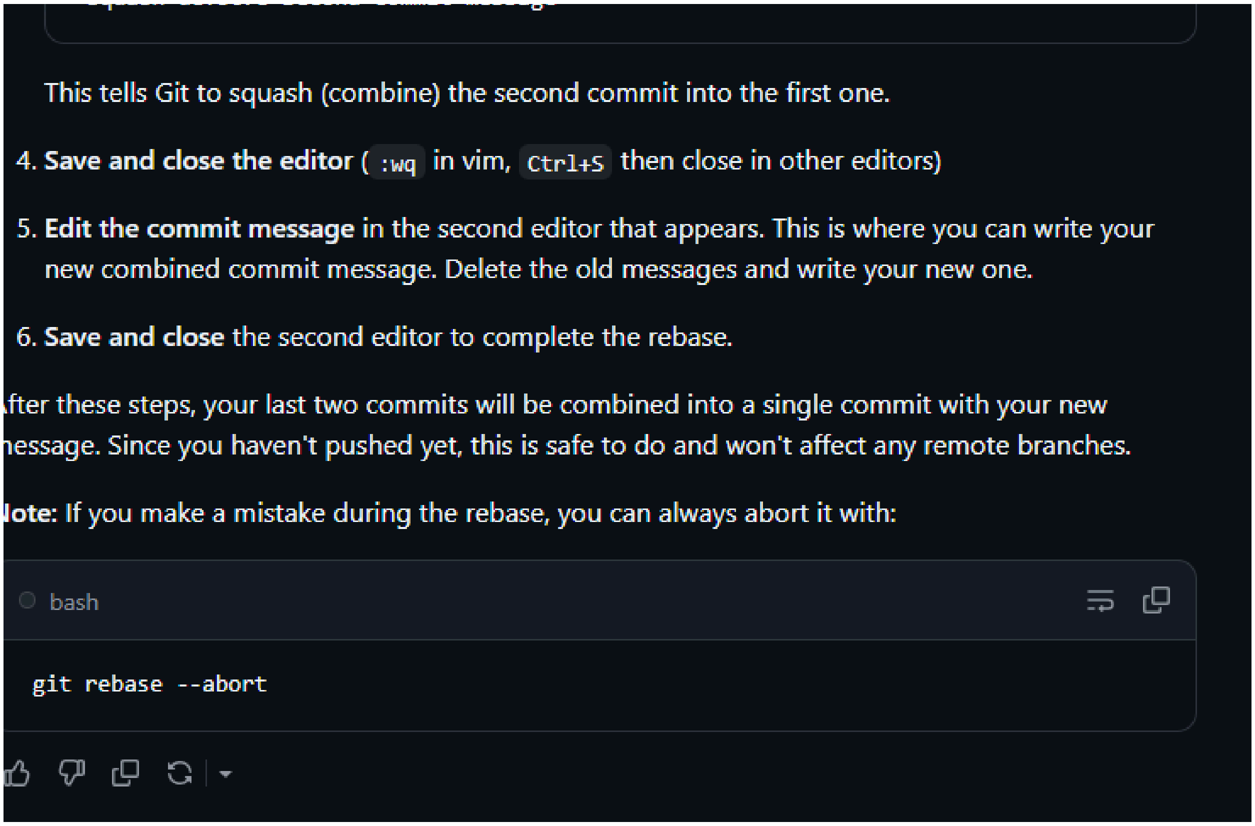
```bash
git rebase -i HEAD~2
```

2. **In the editor that opens**, you'll see something like:

```
pick abc1234 First commit message
pick def5678 Second commit message
```

3. **Change the second `pick` to `squash`** (or `s` for short):

```
pick abc1234 First commit message
squash def5678 Second commit messa
```



This tells Git to squash (combine) the second commit into the first one.

4. **Save and close the editor** (`:wq` in vim, `Ctrl+S` then close in other editors)
5. **Edit the commit message** in the second editor that appears. This is where you can write your new combined commit message. Delete the old messages and write your new one.
6. **Save and close** the second editor to complete the rebase.

fter these steps, your last two commits will be combined into a single commit with your new essage. Since you haven't pushed yet, this is safe to do and won't affect any remote branches.

ote: If you make a mistake during the rebase, you can always abort it with:

```bash
git rebase --abort
```

Overall score

7.5 / 10

The answer works, but it needs clearer editor instructions, conflict handling, and a stronger warning about rebasing pushed commits.