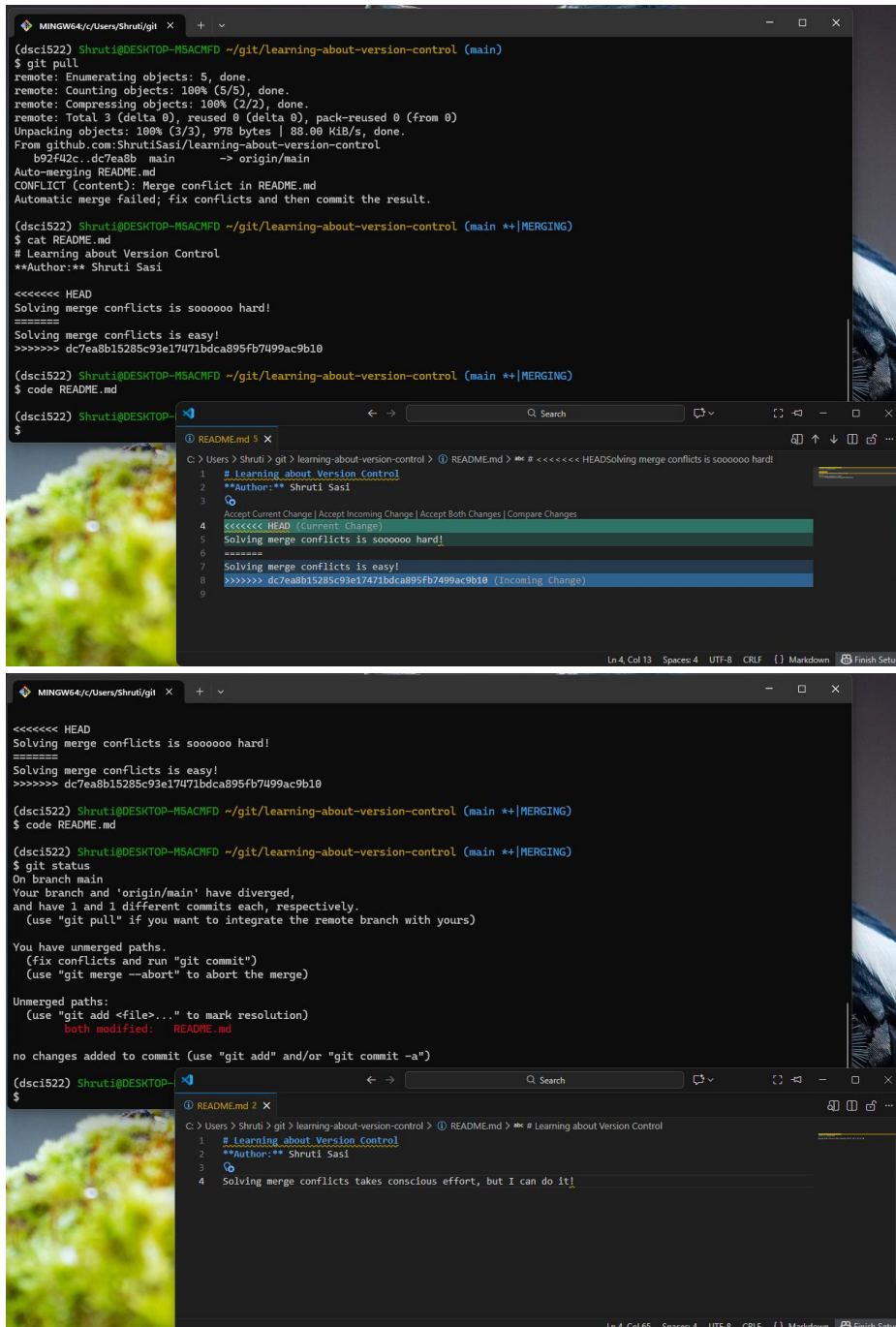


1. The URL of the commit history of GitHub.com repository

<https://github.com/ShrutiSasi/learning-about-version-control/commits/main/>

2. The screenshot of resolved merge conflict

<https://github.com/ShrutiSasi/learning-about-version-control/tree/main/img>



The image consists of two screenshots of a terminal window, likely a Windows Command Prompt, showing the process of resolving a merge conflict in a GitHub repository.

Top Screenshot:

```
(dsc1522) Shruti@DESKTOP-MSACMFD ~/git/learning-about-version-control (main)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 978 bytes | 88.00 KiB/s, done.
From github.com:ShrutiSasi/learning-about-version-control
   b92f42c..dc7ea8b  main       -> origin/main
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

(dsc1522) Shruti@DESKTOP-MSACMFD ~/git/learning-about-version-control (main ++|MERGING)
$ cat README.md
# Learning about Version Control
**Author:** Shruti Sasi

<<<<<< HEAD
Solving merge conflicts is soooooo hard!
=====
Solving merge conflicts is easy!
>>>>>> dc7ea8b15285c93e17471bdca895fb7499ac9b10

(dsc1522) Shruti@DESKTOP-MSACMFD ~/git/learning-about-version-control (main ++|MERGING)
$ code README.md
```

The terminal output shows the result of a `git pull` command. It indicates a merge conflict in `README.md`. The conflict is resolved by manually editing the file. The `code` command opens the file in a code editor, showing the conflict resolution process. The editor displays the current state of the file, with the conflicting lines highlighted. The user has resolved the conflict by keeping the content from the `HEAD` (current change) and adding the content from the `incoming change`.

Bottom Screenshot:

```
(dsc1522) Shruti@DESKTOP-MSACMFD ~/git/learning-about-version-control (main ++|MERGING)
$ code README.md

(dsc1522) Shruti@DESKTOP-MSACMFD ~/git/learning-about-version-control (main ++|MERGING)
$ git status
On branch main
Your branch and 'origin/main' have diverged,
and have 1 and 1 different commits each, respectively.
(use "git pull" if you want to integrate the remote branch with 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:   README.md

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

(dsc1522) Shruti@DESKTOP-MSACMFD ~/git/learning-about-version-control (main ++|MERGING)
$
```

The terminal output shows the result of a `git status` command. It indicates that the branch `main` has diverged from `origin/main`. The user has resolved the conflict and committed the changes. The `git status` command shows that the branch `main` is now up-to-date with `origin/main`.

3. Working in a branch–pull-request workflow allows us to experiment and make changes safely without affecting the main branch, which should always stay stable. It also makes it easier to review, test, and discard changes if needed as opposed to working by directly committing each change to the main branch. This workflow is especially useful when working collaboratively in a team environment, as it reduces merge conflicts and ensures that every change is reviewed before being integrated.