MY project had few commits that were pushed to github, subsequently i made changes in local and added few more files to my project. After a while i used command git reset --hard to first commit.  
Now there are 5 commits in github repo and one commit in my local. I want github repo to have my current local commit, deleting previous commit history on github.

Of course! I can help with that. You've correctly identified the situation: your local repository's history has diverged from the remote (GitHub) repository's history because you used git reset --hard.

To make your GitHub repository match your local repository (i.e., have only the one commit), you need to perform a "force push".

**The Solution: Force Push**

A regular git push will fail because Git sees that you are trying to push changes that would remove commits from the remote repository. This is a safety feature to prevent accidental data loss. To override this, you must use the --force flag.

**Here is the command you need to run:**

Generated bash

git push origin main --force

Or, a slightly safer version:

Generated bash

git push origin main --force-with-lease

IGNORE\_WHEN\_COPYING\_START

content\_copy download

Use code [with caution](https://support.google.com/legal/answer/13505487). Bash

IGNORE\_WHEN\_COPYING\_END

**Step-by-Step Instructions**

1. **Open your Git Bash terminal** in your project directory (e:/Devops/github practice).
2. **Verify your local history** one last time to be absolutely sure it's what you want on GitHub.

Generated bash

git log --oneline

IGNORE\_WHEN\_COPYING\_START

content\_copy download

Use code [with caution](https://support.google.com/legal/answer/13505487). Bash

IGNORE\_WHEN\_COPYING\_END

This should show you only the one commit, like in your screenshot:  
680elab (HEAD -> main) first commit

1. **Execute the force push command.** I recommend using --force-with-lease as a best practice, but for a personal project like this, --force is perfectly fine.

Generated bash

# Recommended safer option

git push origin main --force-with-lease

# Or the more common, direct option

git push origin main --force

IGNORE\_WHEN\_COPYING\_START

content\_copy download

Use code [with caution](https://support.google.com/legal/answer/13505487). Bash

IGNORE\_WHEN\_COPYING\_END

1. **Check your GitHub Repository.** After the command completes, refresh your GitHub repository page in the browser. You will see that the commit history now matches your local repository, showing only 1 commit.

**⚠️ Important Warning: The Dangers of Force Pushing**

What you are doing is **rewriting the public history** of your repository.

* **Destructive Action:** git push --force is a destructive command. It overwrites the branch on the remote server with whatever you have locally, discarding any commits that were on the remote but not in your local history.
* **Collaboration Issues:** If anyone else were working on this project with you, a force push would cause major problems for them. They would need to manually fix their own local repositories.
* **When is it OK?** It is generally considered acceptable to force push **only on your own personal branches or on a repository that nobody else is using**, like this GitPractice project. **Never force push to shared branches like main or develop in a collaborative project.**

For future reference, if you want to "undo" a commit in a shared repository without rewriting history, the correct command is git revert <commit-hash>. This creates a *new* commit that undoes the changes of a previous one, keeping the history intact.