

Built for developers

GitHub is a development platform inspired by the way you work. From [open source](#) to [business](#), you can host and review code, manage projects, and build software alongside 31 million developers.

What is GitHub?


GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

This tutorial teaches you GitHub essentials like *repositories*, *branches*, *commits*, and *Pull Requests*. You'll create your own [RETAIL_PROJECTS] repository and learn GitHub's Pull Request workflow, a popular way to create and review code.

Plan:

Choose your plan

With tools developers love and the world's largest open source community, there's no wrong choice.



Free

The basics of GitHub for every developer


\$0

per month

Includes:

- ∞ Unlimited public and private repositories
- ✓ 3 collaborators for private repositories
- ✓ Issues and bug tracking
- ✓ Project management

Are you a [student](#)? Get access to the best developer tools for free with the [GitHub Student Developer Pack](#).



Pro

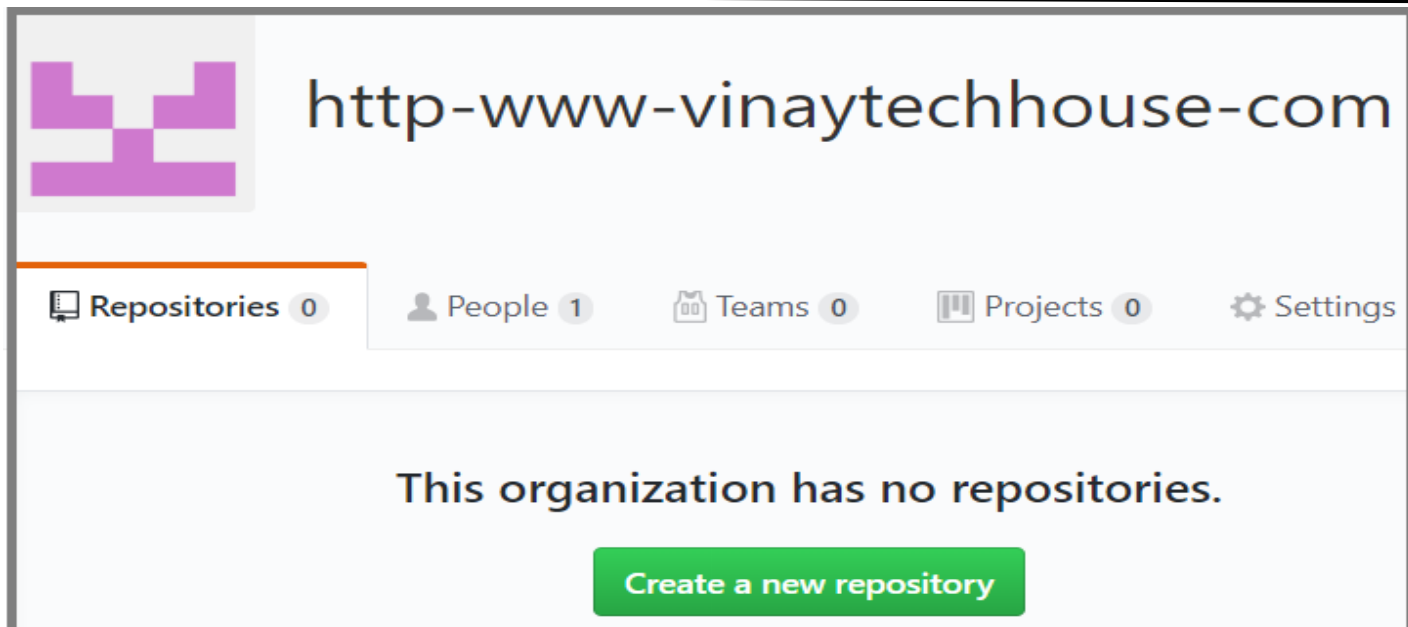
Pro tools for developers with advanced requirements

₹497.25

per month
([view in USD](#))

Includes:

- ∞ Unlimited public and private repositories
- ∞ Unlimited collaborators
- ✓ Issues and bug tracking
- ✓ Project management
- ✓ [Advanced tools and insights](#)



No coding necessary

To complete this tutorial, you need a [GitHub.com account](#) and Internet access. You don't need to know how to code, use the command line, or install Git (the version control software GitHub is built on).

Tip: Open this guide in a separate browser window (or tab) so you can see it while you complete the steps in the tutorial.

Step 1. Create a Repository

A **repository** is usually used to organize a single project. Repositories can contain folders and files, images, videos, spreadsheets, and data sets – anything your project needs. We recommend including a *README*, or a file with information about your project. GitHub makes it easy to add one at the same time you create your new repository. *It also offers other common options such as a license file.*

Your Vinaytech repository can be a place where you store ideas, resources, or even share and discuss things with others.

To create a new repository

1. In the upper right corner, next to your avatar or identicon, click and then select **New repository**.
2. Name your repository RETAIL_PROJECTS.
3. Write a short description.
4. Select **Initialize this repository with a README**.

Click **Create repository**.

Create a new repository

A repository contains all project files, including the revision history.

Owner

vinaytechhouse

Repository name *

RETAIL_PROJECTS

Great repository names are short and memorable. Need inspiration? How about [super-duper-guide?](#)

Description (optional)

Distribution and retail projects available

☒ Public

Anyone can see this repository. You choose who can commit.

☐ Private

You choose who can see and commit to this repository.

☒ Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None

Add a license: None

Create repository

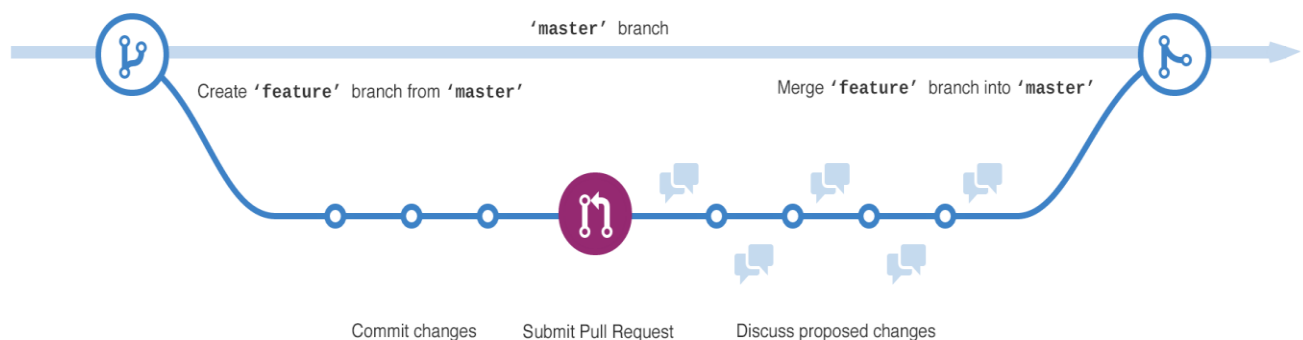
Step 2. Create a Branch

Branching is the way to work on different versions of a repository at one time.

By default your repository has one branch named `master` which is considered to be the definitive branch. We use branches to experiment and make edits before committing them to `master`. When you create a branch off the `master` branch, you're making a copy, or snapshot, of `master` as it was at that point in time. If someone else made changes to the `master` branch while you were working on your branch, you could pull in those updates.

This diagram shows:

- The `master` branch
- A new branch called `feature` (because we're doing 'feature work' on this branch)
- The journey that `feature` takes before it's merged into `master`



Have you ever saved different versions of a file? Something like:

- `story.txt`
- `story-joe-edit.txt`
- `story-joe-edit-reviewed.txt`

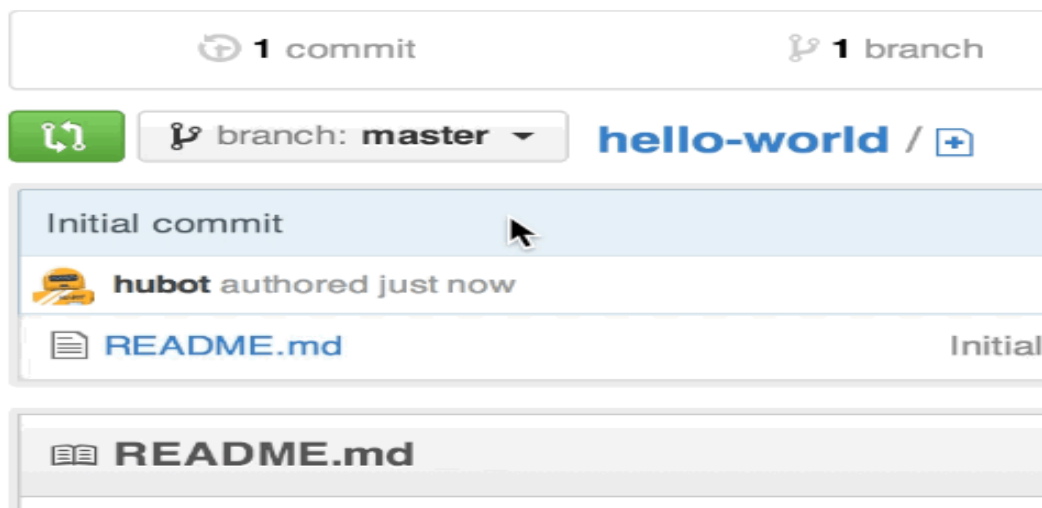
Branches accomplish similar goals in GitHub repositories.

Here at GitHub, our developers, writers, and designers use branches for keeping bug fixes and feature work separate from our `master` (production) branch. When a change is ready, they merge their branch into `master`.

To create a new branch

1. Go to your new repository `hello-world`.
2. Click the drop down at the top of the file list that says **branch: master**.
3. Type a branch name, `readme-edits`, into the new branch text box.
4. Select the blue **Create branch** box or hit “Enter” on your keyboard.

Just another repository — Edit



Now you have two branches, `master` and `readme-edits`. They look exactly the same, but not for long! Next we'll add our changes to the new branch.

Step 3. Make and commit changes

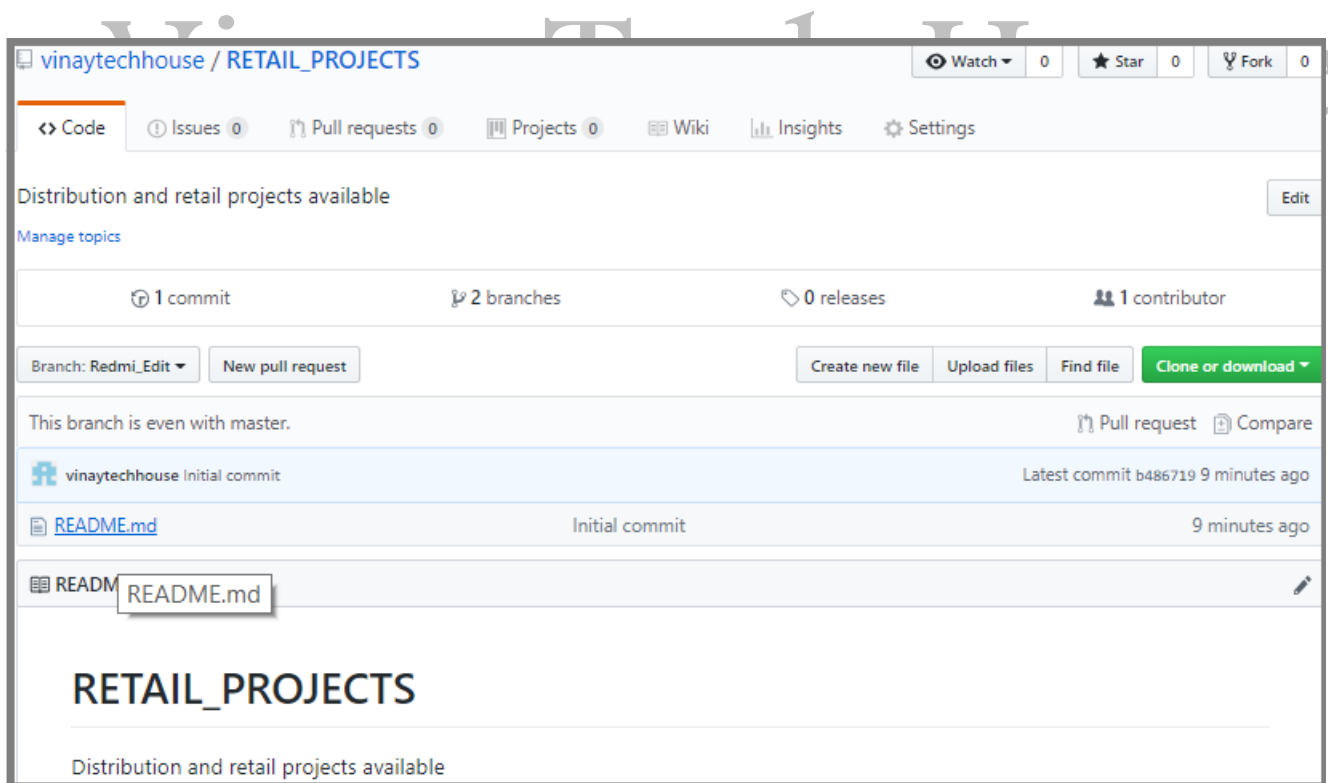
Bravo! Now, you're on the code view for your `readme-edits` branch, which is a copy of `master`. Let's make some edits.

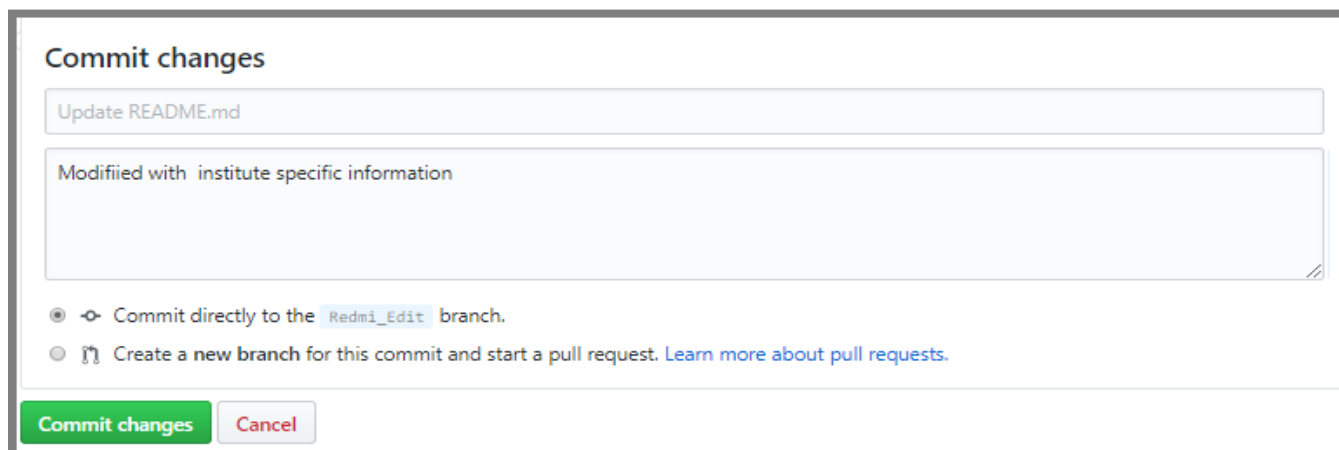
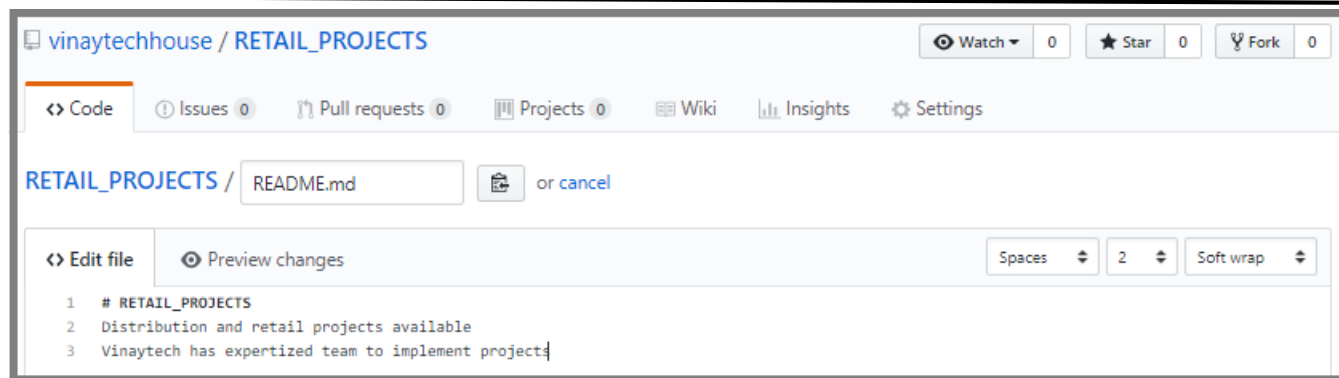
On GitHub, saved changes are called *commits*. Each commit has an associated *commit message*, which is a description explaining why a particular change was made. Commit messages capture the history of your changes, so other contributors can understand what you've done and why.

Make and commit changes

1. Click the `README.md` file.
2. Click the pencil icon in the upper right corner of the file view to edit.
3. In the editor, write a bit about yourself.
4. Write a commit message that describes your changes.
5. Click **Commit changes** button.

These changes will be made to just the README file on your `readme-edits` branch, so now this branch contains content that's different from `master`.





Step 4. Open a Pull Request

Nice edits! Now that you have changes in a branch off of `master`, you can open a *pull request*.

Pull Requests are the heart of collaboration on GitHub. When you open a *pull request*, you're proposing your changes and requesting that someone review and pull in your contribution and merge them into their branch. Pull requests show *diffs*, or differences, of the content from both branches. The changes, additions, and subtractions are shown in green and red.

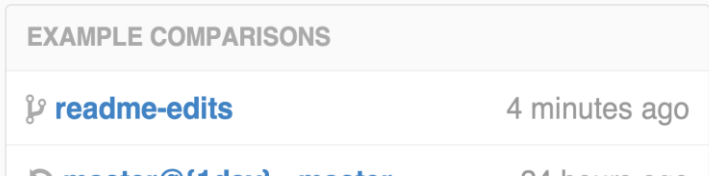
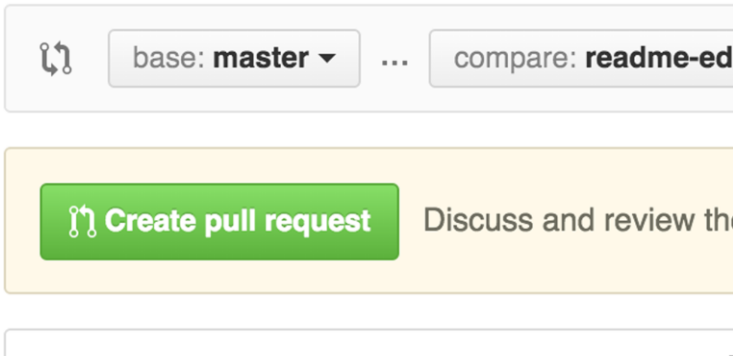
As soon as you make a commit, you can open a pull request and start a discussion, even before the code is finished.

By using GitHub's [@mention system](#) in your pull request message, you can ask for feedback from specific people or teams, whether they're down the hall or 10 time zones away.

You can even open pull requests in your own repository and merge them yourself. It's a great way to learn the GitHub flow before working on larger projects.

Open a Pull Request for changes to the README

Click on the image for a larger version

Step	Screenshot
Click the Pull Request tab, then from the Pull Request page, click the green New pull request button.	
In the Example Comparisons box, select the branch you made, <code>readme-edits</code> , to compare with <code>master</code> (the original).	
Look over your changes in the diffs on the Compare page, make sure they're what you want to submit.	
When you're satisfied that these are the changes you want to submit, click the big green Create Pull Request button.	

Step	Screenshot
Give your pull request a title and write a brief description of your changes.	

When you're done with your message, click **Create pull request!**

Tip: You can use [emoji](#) and [drag and drop images and gifs](#) onto comments and Pull Requests.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master
compare: Redmi_Edit
✓ Able to merge. These branches can be automatically merged.

Update README.md

Write

Preview

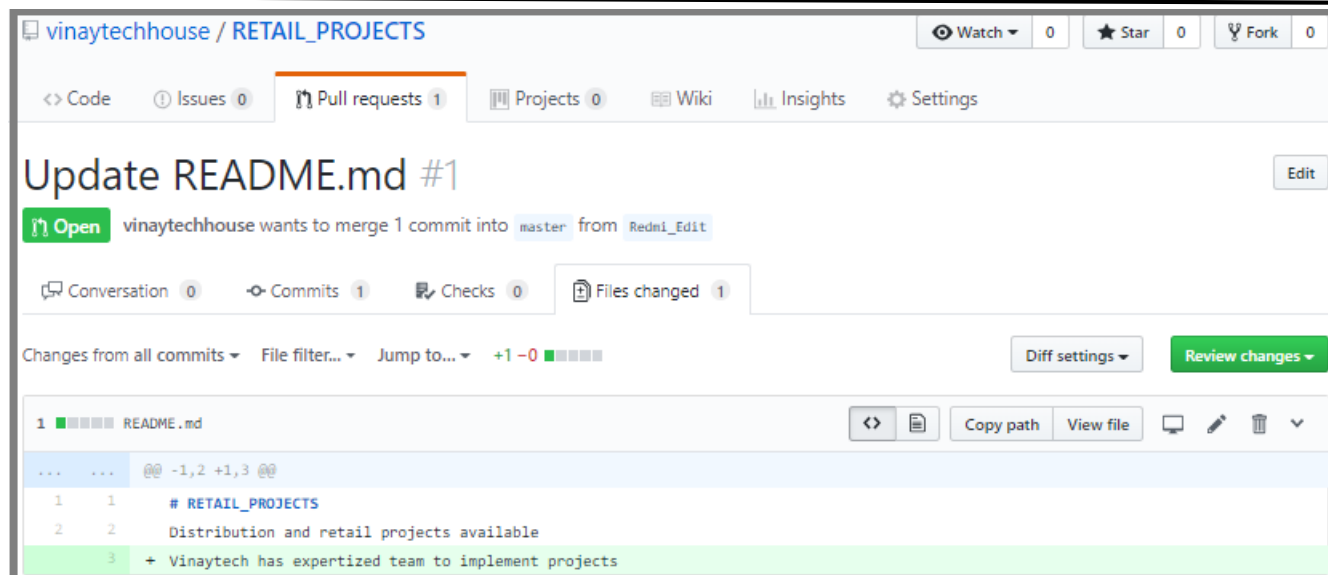
AA B i
 “ < >
 ⌨
 ⋮ ⋮ ⋮
 @ 📌 ↩

Modified with institute specific information

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

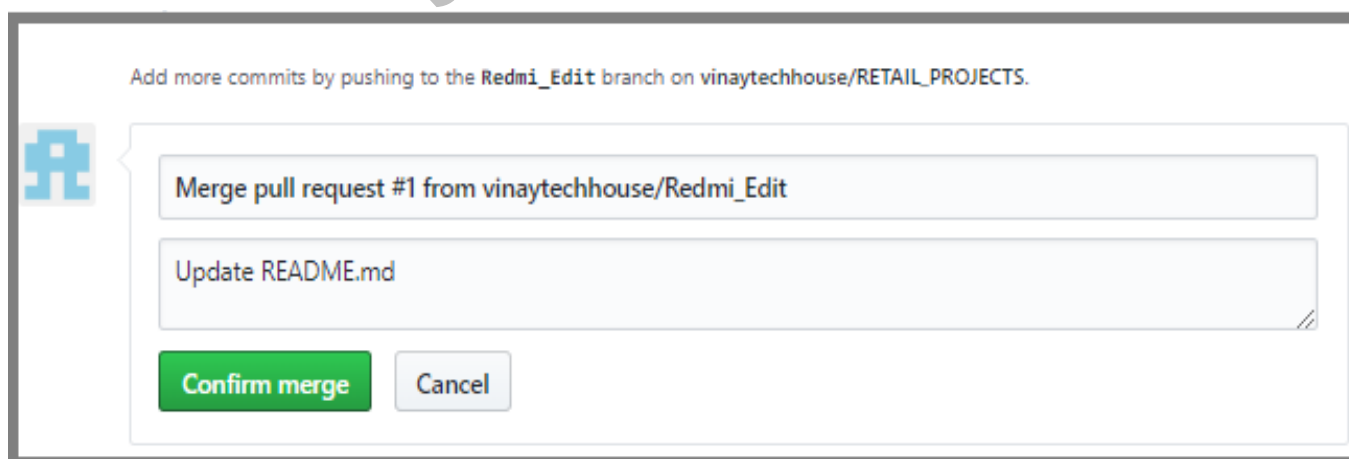
Create Pull Request

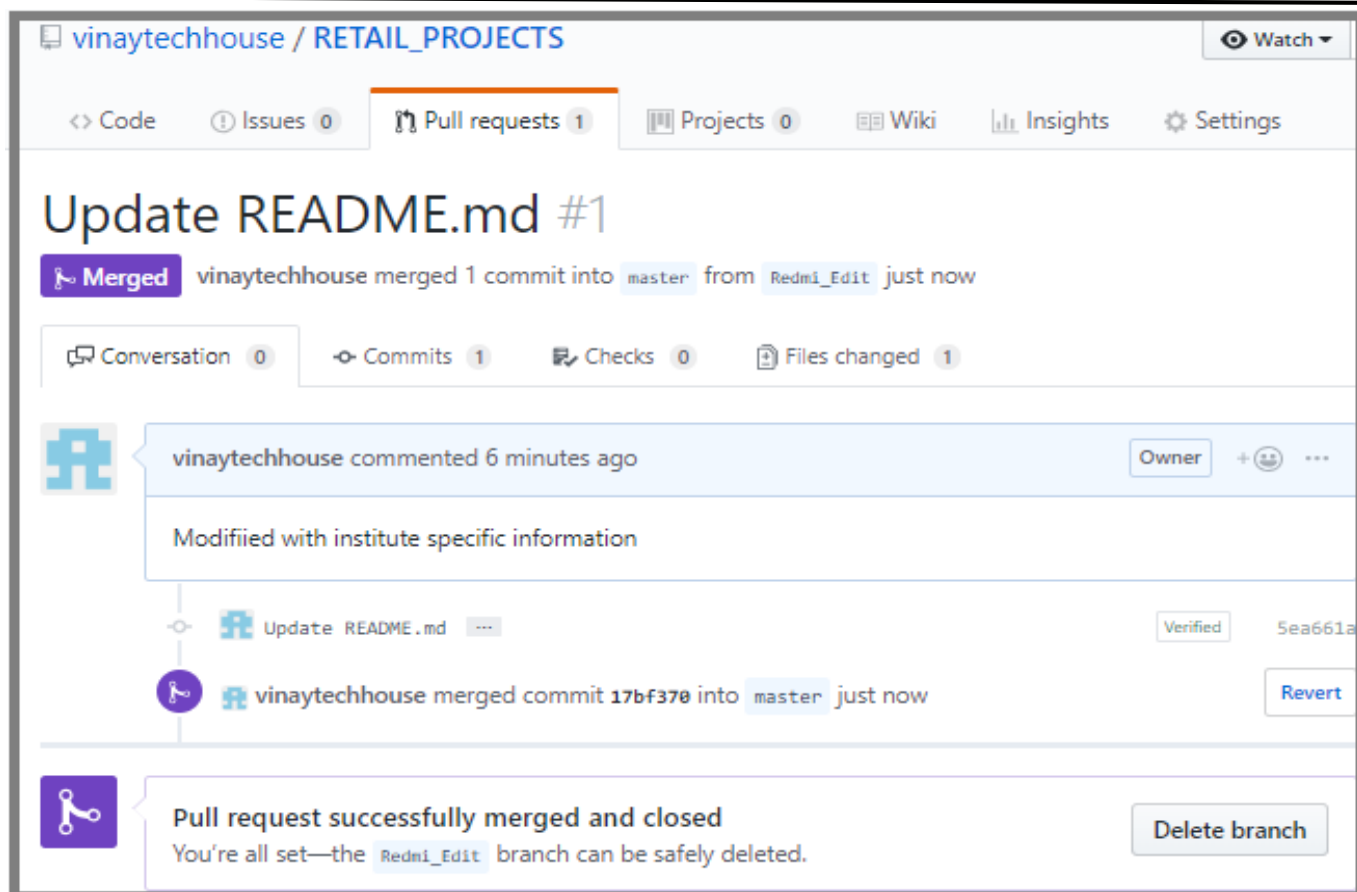


Step 5. Merge your Pull Request

In this final step, it's time to bring your changes together – merging your `readme-edits` branch into the `master` branch.

1. Click the green **Merge pull request** button to merge the changes into `master`.
2. Click **Confirm merge**.
3. Go ahead and delete the branch, since its changes have been incorporated, with the **Delete branch** button in the purple box.





This screenshot shows the GitHub interface for a pull request titled "Update README.md #1". The repository is "vinaytechhouse / RETAIL_PROJECTS". The pull request is in the "Merged" state, having been merged into the "master" branch from the "Redmi_Edit" branch just now. The interface includes tabs for Code, Issues (0), Pull requests (1), Projects (0), Wiki, Insights, and Settings. Below the title, there are statistics for Conversation (0), Commits (1), Checks (0), and Files changed (1). A comment from "vinaytechhouse" states "Modified with institute specific information". The commit history shows the commit "17bf370" being merged into "master" just now. A message at the bottom indicates the pull request was successfully merged and closed, suggesting the "Redmi_Edit" branch can be safely deleted. A "Delete branch" button is visible.

vinaytechhouse / RETAIL_PROJECTS

<> Code Issues 0 Pull requests 1 Projects 0 Wiki Insights Settings

Update README.md #1

Merged vinaytechhouse merged 1 commit into master from Redmi_Edit just now

Conversation 0 Commits 1 Checks 0 Files changed 1

vinaytechhouse commented 6 minutes ago

Owner + 👤 ...

Modified with institute specific information

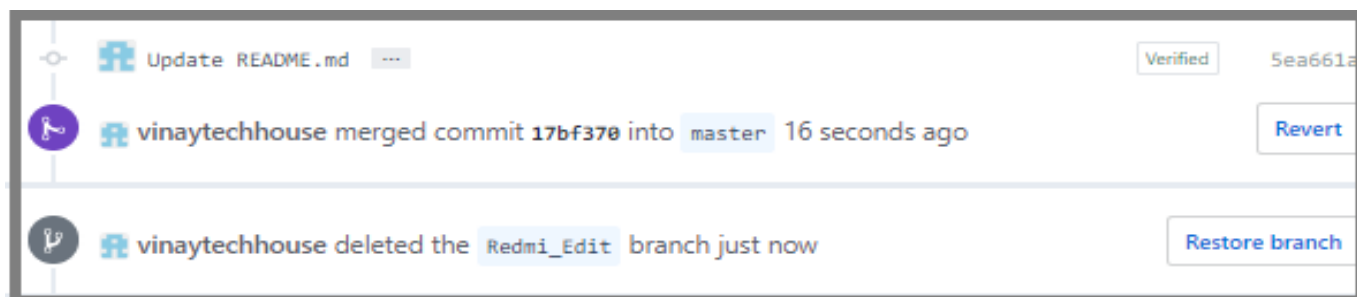
Update README.md ... Verified 5ea661a

vinaytechhouse merged commit 17bf370 into master just now Revert

Pull request successfully merged and closed

You're all set—the Redmi_Edit branch can be safely deleted.

Delete branch



This screenshot continues the history of the pull request. It shows the commit "17bf370" being merged into "master" 16 seconds ago. Below this, it shows the "Redmi_Edit" branch being deleted just now. A "Restore branch" button is visible.

Update README.md ... Verified 5ea661a

vinaytechhouse merged commit 17bf370 into master 16 seconds ago Revert

vinaytechhouse deleted the Redmi_Edit branch just now Restore branch