



# Git/GitHub Demo

CHANDRIKA MUKHERJEE  
COMPUTER SCIENCE

# Topics

1. What is GitHub
2. How to Use GitHub
3. Create a Repo
4. Commit and Adding Changes to Remote
5. Branch
6. Pull Request
7. Merge
8. Resolve Conflicts
9. Some Useful Commands

# What is GitHub?

- ▶ A product that helps developer to store code and collaborate with others.
- ▶ Use of GitHub?
  - ▶ Version or Source Control
    - ▶ To keep track and manage every change made to the code.
      - ▶ Protects source code from catastrophe and the casual degradation of human error and unintended consequences.

▶ Sources -

▶ <https://www.atlassian.com/git/tutorials/what-is-version-control>

▶ <https://docs.github.com/en>

# How to Use GitHub?

- ▶ Store GitHub Credentials and connect to GitHub over HTTPS

- ▶ GitHub CLI

- ▶ [Install](#) GitHub CLI

- ▶ In the command line, enter `gh auth login`, then follow the prompts.

- ▶ [GitHub CLI Manual](#)

- ▶ Git Credential Manager

- ▶ Mac :

- ▶ Install git using Homebrew (package installer for Mac, Linux)

- ▶ Install GCM using Homebrew

- ▶ Windows :

- ▶ Install Git for Windows, which includes GCM. For more information, see "[Git for Windows releases](#)" from its [releases page](#)

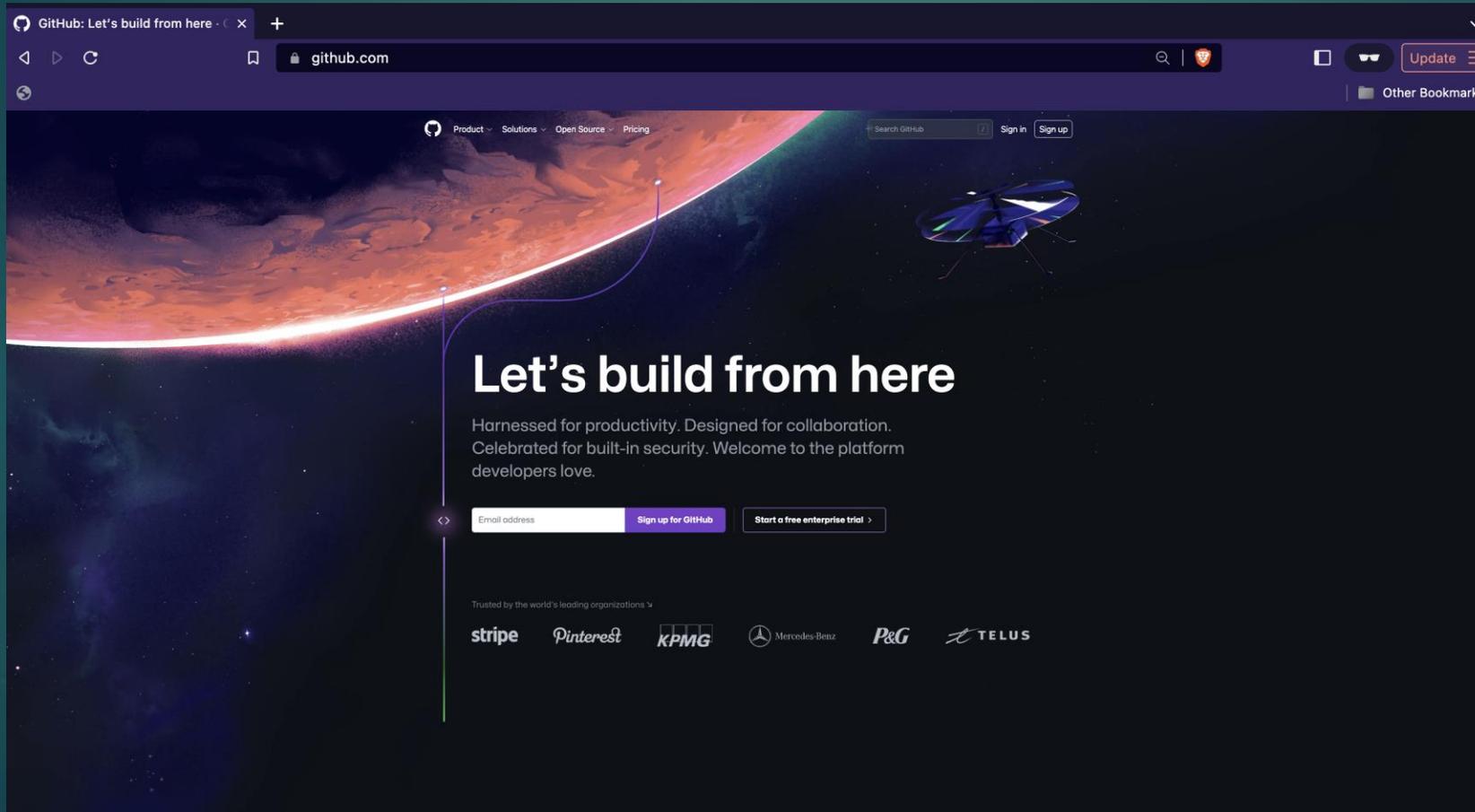
- ▶ Linux :

- ▶ Install Git from your distro's packaging system > Install GCM > Configure Git to use GCM

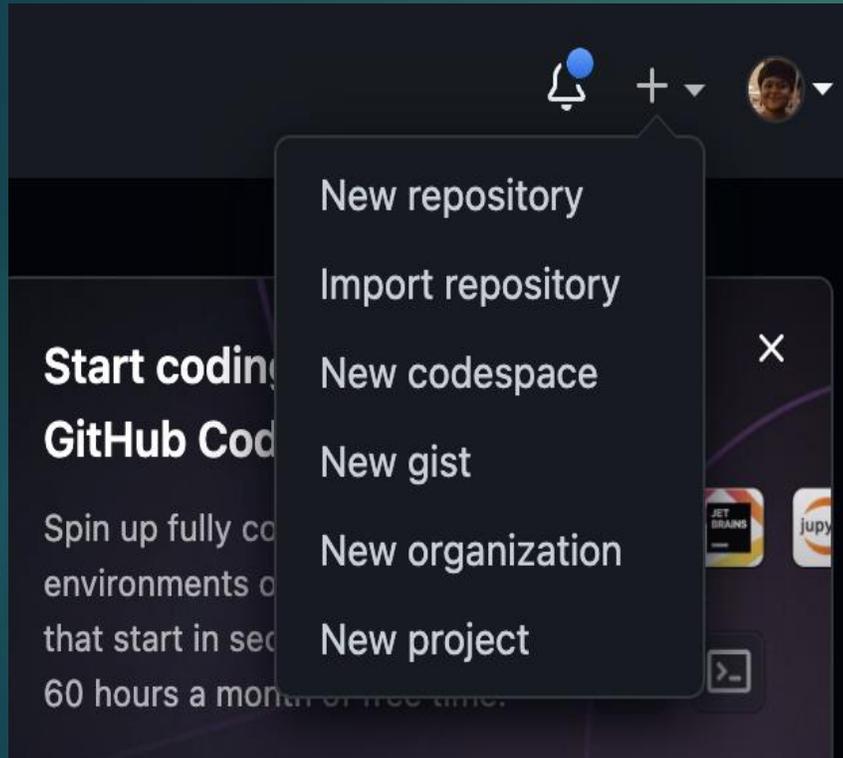
Source – <https://docs.github.com/en/get-started/getting-started-with-git/caching-your-github-credentials-in-git>

# Basics

Need to have a GitHub Account –



# Create a GitHub Repository



## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

---

**Owner \*** **Repository name \***

 chandms /

Great repository names are short and memorable. Need inspiration? How about [congenial-computing-machine?](#)

**Description (optional)**

---

**Public**  
Anyone on the internet can see this repository. You choose who can commit.

**Private**  
You choose who can see and commit to this repository.

---

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

**Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

**Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

**Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

---

ⓘ You are creating a public repository in your personal account.

---

# Creating the Repository Locally and Linking with Remote Repository

- ▶ Clone the GitHub Repo
  - ▶ git clone [https://github.com/<username>/<repo\\_name>](https://github.com/<username>/<repo_name>)
- ▶ Go to the repository folder
- ▶ Command to make the empty/existing repo as a git repository
  - ▶ git init
- ▶ Link the local repository with the remote repository
  - ▶ git remote add origin <url of the repo>

# Adding Changes to GitHub Repository

- ▶ Add/Change anything in the repository
- ▶ Check the status of the repository
  - ▶ `git status`
- ▶ Add the new changes to next commit
  - ▶ `git add .`
    - ▶ or
  - ▶ `git add <file name>` (to add any specific file change to next commit)
- ▶ Commit staged content as a new commit snapshot
  - ▶ `git commit -m "<specific comment to recognize the commit>"`
- ▶ Push the commit to the corresponding remote branch
  - ▶ `git push origin main`

# Making and committing changes

- ▶ Instead of doing locally, we can directly make changes GitHub
  - ▶ Using Add/Edit a file
  - ▶ Adding a commit using the form below

### Commit new file

  
  
 @gmail.com  
Choose which email address to associate with this commit  
 Commit directly to the `main` branch.  
 Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commits on Jan 1, 2023

**adding blog in navbar**  
chandms committed 4 days ago

 731fa26  

Commits on Dec 24, 2022

**changing routes**  
chandms committed 2 weeks ago

 088ed0e  

**adding routes**  
chandms committed 2 weeks ago

 93bbc54  

**changing responsiveness**  
chandms committed 2 weeks ago

 d97bc73  

Commits on Dec 21, 2022

**responsiveness change**  
chandms committed 2 weeks ago

 0827917  

**responsiveness change**  
chandms committed 2 weeks ago

 1dba6e9  

**changing icon and arranging pictures, pdfs**  
chandms committed 2 weeks ago

 22e7744  

Commits on Dec 20, 2022

**changing the contact form**  
chandms committed 2 weeks ago

 a0a6f5d  

**changing the contact form**  
chandms committed 2 weeks ago

 4b3bc18  

Commits on Dec 19, 2022

**changing slides**  
chandms committed 2 weeks ago

 8366927  

**changing slides**  
chandms committed 2 weeks ago

 6c331b6  

**first commit**  
chandms committed 2 weeks ago

 9b048b0  

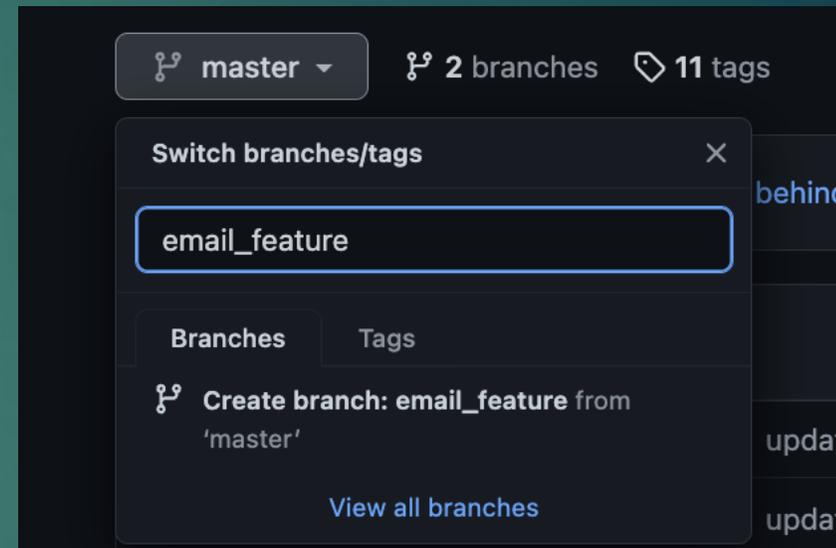
# Branch

Branching lets you have different versions of a repository at one time.

By default, each repository has one branch named “main”

## In the Command Line

1. List all branches  
`git branch`
2. Create a branch  
`git branch <branch name>`
3. To switch to a different branch  
`git checkout <branch name>`



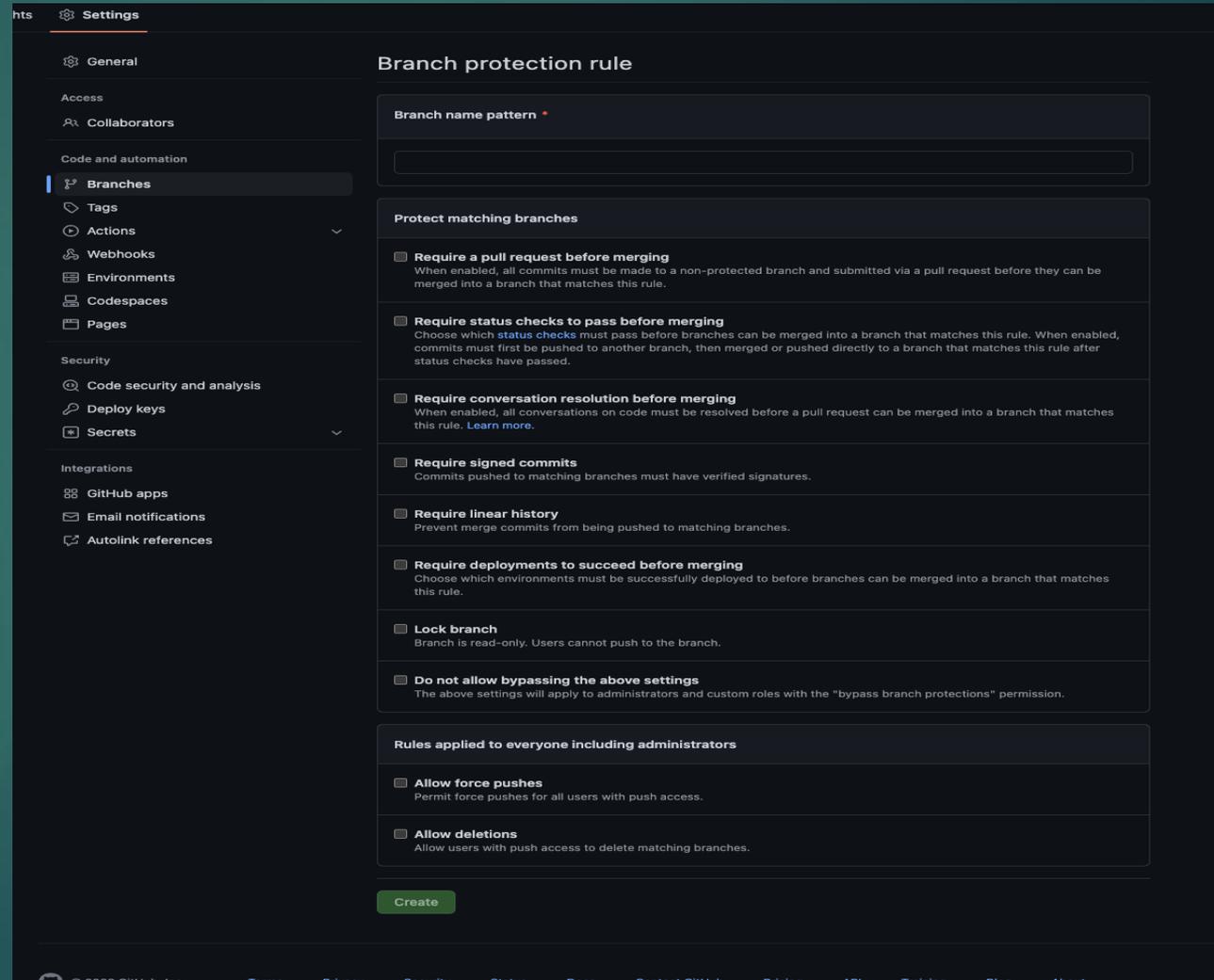
For example, here in the example -

We are creating new branch  
“email\_feature” from “master” branch

# Protect Branch

Need to add rules

- How a modification can be done on that branch.
- Who are the approvers.
- If the rule applies to the administrator as well.



The screenshot shows the GitHub Settings page for a repository, specifically the 'Branches' section. The left sidebar contains navigation options: General, Access (Collaborators), Code and automation (Branches, Tags, Actions, Webhooks, Environments, Codespaces, Pages), Security (Code security and analysis, Deploy keys, Secrets), and Integrations (GitHub apps, Email notifications, Autolink references). The main content area is titled 'Branch protection rule' and includes a 'Branch name pattern' input field. Below this, there are several sections of settings, each with a checkbox and a description:

- Protect matching branches**
  - Require a pull request before merging**  
When enabled, all commits must be made to a non-protected branch and submitted via a pull request before they can be merged into a branch that matches this rule.
  - Require status checks to pass before merging**  
Choose which **status checks** must pass before branches can be merged into a branch that matches this rule. When enabled, commits must first be pushed to another branch, then merged or pushed directly to a branch that matches this rule after status checks have passed.
  - Require conversation resolution before merging**  
When enabled, all conversations on code must be resolved before a pull request can be merged into a branch that matches this rule. [Learn more.](#)
  - Require signed commits**  
Commits pushed to matching branches must have verified signatures.
  - Require linear history**  
Prevent merge commits from being pushed to matching branches.
  - Require deployments to succeed before merging**  
Choose which environments must be successfully deployed to before branches can be merged into a branch that matches this rule.
  - Lock branch**  
Branch is read-only. Users cannot push to the branch.
  - Do not allow bypassing the above settings**  
The above settings will apply to administrators and custom roles with the "bypass branch protections" permission.
- Rules applied to everyone including administrators**
  - Allow force pushes**  
Permit force pushes for all users with push access.
  - Allow deletions**  
Allow users with push access to delete matching branches.

A 'Create' button is located at the bottom of the configuration area. The footer of the page includes copyright information and links to Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

# Pull Request

- ▶ Scenario –
  - ▶ To add all the changes of a feature branch to main branch -> Need to create a pull request.
- ▶ New Pull Request > Select Branch (Base -> Target branch, Compare -> The branch with desired changes )

# 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: main ← compare: test\_feature ✓ **Able to merge.** These branches can be automatically merged.



Test feature

Write Preview

H B I

Leave a comment

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

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Reviewers

No reviews—at least 1 approving review is required.

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Use [Closing keywords](#) in the description to automatically close issues

Helpful resources

[GitHub Community Guidelines](#)

2 commits

2 files changed

1 contributor

Commits on Jan 5, 2023

"Demo file added"

chandms committed 2 minutes ago

Verified



79463ab



Tip

Sample Readme Change for DEMO

chandms committed now

Verified



19a5e7a



Tip

# Demo Pull Request #1

Edit <> Code

**Open** chandms wants to merge 2 commits into `main` from `test_feature`

Conversation 0 Commits 2 Checks 0 Files changed 2

+2 -0



chandms commented now

Tip

No description provided.

chandms added 2 commits 3 minutes ago

"Demo file added"

Verified 79463ab

Sample Readme Change for DEMO

Verified 19a5e7a

Add more commits by pushing to the `test_feature` branch on `chandms/blogs`.



**This branch has not been deployed**

No deployments



**Review required**

At least 1 approving review is required by reviewers with write access. [Learn more.](#)



**Merging is blocked**

Merging can be performed automatically with 1 approving review.

Merge without waiting for requirements to be met (bypass branch protections)

Merge pull request

You can also open this in GitHub Desktop or view command line instructions.



Write

Preview

H B I  $\equiv$  <> @

Leave a comment

Reviewers



No reviews—at least 1 approving review is required.

Assignees



No one—assign yourself

Labels



None yet

Projects



None yet

Milestone



No milestone

Development



Successfully merging this pull request may close these issues.

None yet

Notifications

Customize



Unsubscribe

You're receiving notifications because you're watching this repository.

1 participant



Lock conversation

# Git Merge

- ▶ We can directly merge from GitHub by clicking on Merge button.
  - ▶ (provided – there is no merge conflict).
- ▶ Say, we need to merge email\_feature branch to main branch
  - ▶ `git checkout main` (To stay in main branch)
  - ▶ `git merge email_feature`

# Resolve Conflicts

- ▶ Scenario – While merging branch A to branch B, we get conflicts.
  - ▶ In the local, pull the recent changes.
  - ▶ Merge the branches -> it will not be merged because of the conflicts.
  - ▶ Manually make necessary changes to the file.
  - ▶ Create a new commit and start a pull request and continue the process of merging.
- ▶ Some important Commands –
  - ▶ `git log –merge` [list of commits that are creating conflicts]
  - ▶ `git diff` [ identify differences between states/repo/files]
  - ▶ `git merge –abort` [exiting current merge process and return to the state before merging]

# Some important Commands

- ▶ `git pull` -> fetch and merge any commits from the tracking remote branch
- ▶ `git fetch` -> fetch down all the branches from that Git remote
- ▶ `git rebase` -> apply any commits of current branch ahead of specified one
- ▶ `git stash` -> Save modified and staged changes
- ▶ `git diff branchB...branchA` -> show the diff of what is in branchA that is not in branch
- ▶ `git log` -> show the commit history for the currently active branch
  - ▶ Source - <https://education.github.com/git-cheat-sheet-education.pdf>

Thank You