



Dharmsinh Desai University

MCA Sem 2

Data structure using C

Report

Name: Mansi Nukani

Roll no: MA012

Student id: 18MAPOG005

Submitted to

Prof.Himansu Purohit

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.

What is Git?

- Git is a distributed revision control and source code management system with an emphasis on speed. Git was initially designed and developed by Linus Torvalds for Linux kernel development.
- ❖ **Version Control System (VCS)** is a software that helps software developers to work together and maintain a complete history of their work.
 - Listed below are the functions of a VCS –
 - Allows developers to work simultaneously.
 - Does not allow overwriting each other's changes.
 - Maintains a history of every version.
 - Following are the types of VCS –
 - Centralized version control system (CVCS).
 - Distributed/Decentralized version control system (DVCS).
- ❖ **Distributed Version Control System**
 - DVCS clients not only check out the latest snapshot of the directory but they also fully mirror the repository. If the server goes down, then the repository from any client can be copied back to the server to restore it.
 - Every checkout is a full backup of the repository. Git does not rely on the central server and that is why you can perform many operations when you are offline. You can commit changes, create branches, view logs, and perform other operations when you are offline.
 - You require network connection only to publish your changes and take the latest changes.

What is the use of GitHub?

- GitHub is an online; browser based distributed version control system for software developers using the Git revision control system. The service provides free public repositories, issue tracking, graphs, code review, downloads, wikis, collaborator management, and more.

- There are also social networking elements of the service including feeds, the ability to follow other users, and network related graphs. This article will walk you through the process of signing up for GitHub.

Advantages of Git

- Free and open source
- Fast and small
- Implicit backup
- Security
- No need of powerful hardware
- Easier branching

Disadvantages

- Steep learning curve: Many commands with many options, some commands are non-intuitive and need a level of understanding the internals of git, commands and arguments are inconsistent to some degree
- Binary files are a big no: If your project has non-text files that are updated frequently (images for websites or MS Office documents), then git becomes bloated and slow. (I believe it still does better than most systems).

Steps to create account

1. go to the github sign up page

The screenshot shows the GitHub sign-up process, specifically Step 1: Create personal account. The page is divided into three steps: Step 1 (Create personal account), Step 2 (Choose your plan), and Step 3 (Tailor your experience). The main form for Step 1 includes fields for Username, Email Address, and Password. To the right, there's a section titled 'You'll love GitHub' listing benefits like unlimited collaborators, unlimited public repositories, great communication, frictionless development, and open source community. At the bottom, there's a green 'Create an account' button and a link to the Terms of Service and Privacy Policy.

Step 1:
Create personal account

Step 2:
Choose your plan

Step 3:
Tailor your experience

Create your personal account

Username

This will be your username — you can enter your organization's username next.

Email Address

You will occasionally receive account related emails. We promise not to share your email with anyone.

Password

Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).

Create an account

You'll love GitHub

- Unlimited collaborators
- Unlimited public repositories
- ✓ Great communication
- ✓ Frictionless development
- ✓ Open source community

2. Enter a username, valid email address, and password. Use at least one lowercase letter, one numeral, and seven characters.

Create your personal account

Username



This will be your username — you can enter your organization's username next.

Email Address



You will occasionally receive account related emails. We promise not to share your email with anyone.

Password



Use at least one lowercase letter, one numeral, and seven characters.

By clicking on "Create an account" below, you are agreeing to the [Terms of Service](#) and the [Privacy Policy](#).



- 3 Review carefully the [GitHub Terms of Service](#) and [Privacy Policy](#) before continuing. Upon clicking the "Create an account" button you will simultaneously be agreeing to these documents.

4 **Choose a plan.** You have two choice: Free and paid, the paid version has private repositories with \$7/month. You should try the free version then have the suitable choice.

Choose your personal plan

☒ Unlimited public repositories for free.

☐ Unlimited private repositories for \$7/month.


Don't worry, you can cancel or upgrade at any time.


☐ **Help me set up an organization next**
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.
[Learn more about organizations.](#)

 **Continue**

5 **Tailor experience.** If you have time, fill in the survey or skip it

✓ Completed
Set up a personal account

 Step 2:
Choose your plan

 Step 3:
Tailor your experience

How would you describe your level of programming experience?

☐ Totally new to programming ☐ Somewhat experienced ☐ Very experienced

What do you plan to use GitHub for? (check all that apply)



☐ Design ☐ Development ☐ Project Management
☐ School projects ☐ Research ☐ Other (please specify)

Which is closest to how you would describe yourself?

☐ I'm a student ☐ I'm a hobbyist ☐ I'm a professional
☐ Other (please specify)

What are you interested in?

e.g. tutorials, android, ruby, web-development, machine-learning, open-source

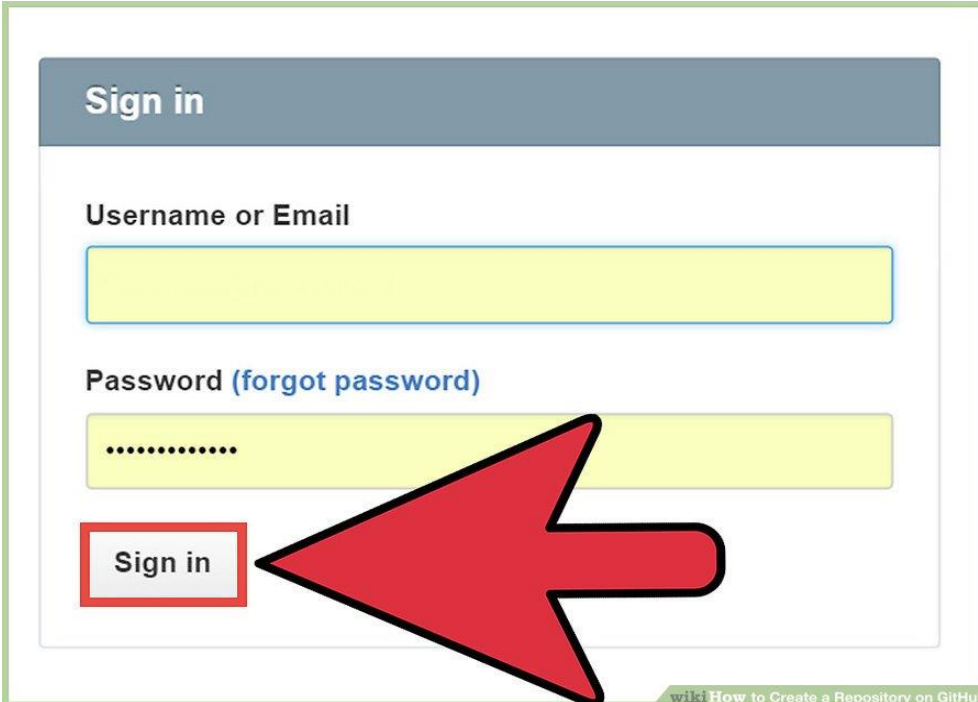
 **Submit** [skip this step](#) 

[Sign in now to](#)

6 You finished! Your Github account created!

How to create repository

1. Sign up for a GitHub account, if you don't already have one.
2. Go to the [GitHub login page](#) and log in using the email and password associated with your account.



The image shows a screenshot of the GitHub 'Sign in' page. The page has a dark blue header with the text 'Sign in' in white. Below the header, there are two input fields: 'Username or Email' and 'Password (forgot password)'. Both fields are highlighted in yellow. The 'Password' field contains a series of dots. Below the password field, there is a 'Sign in' button with a red border. A large red arrow with a black outline points from the right towards the 'Sign in' button. At the bottom right of the screenshot, there is a small green banner with the text 'wiki How to Create a Repository on GitHub'.

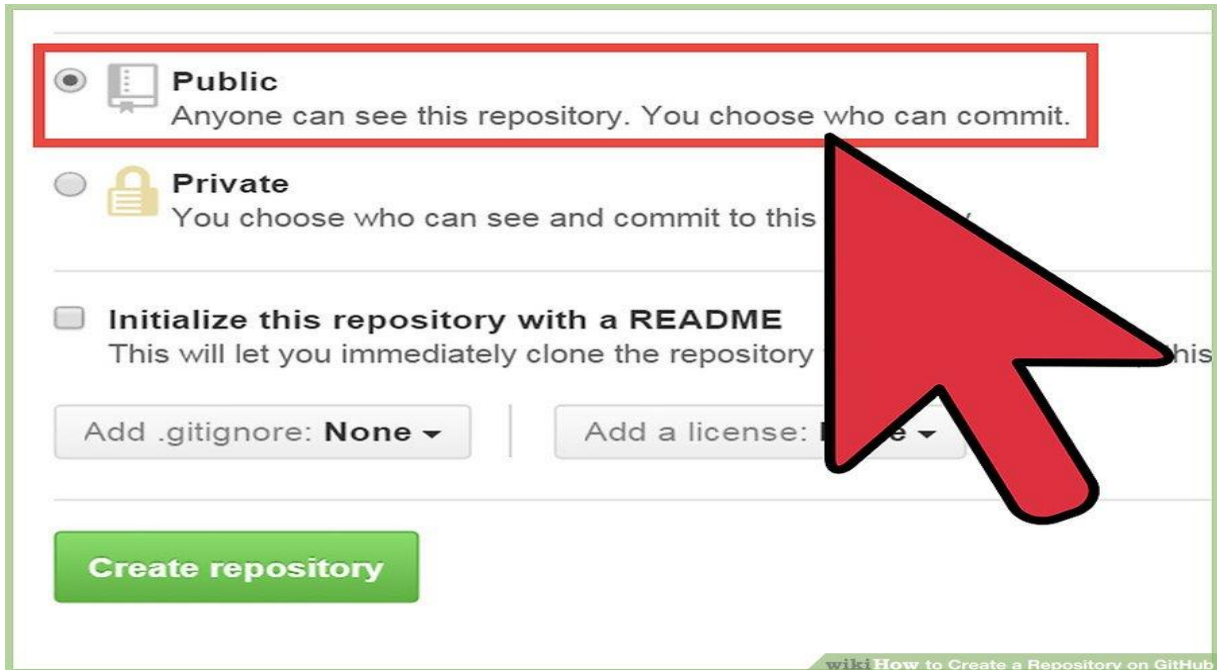
3. Click the “New Repository” button in the “Your Repositories” pane on right side of the page.



4. Enter a project name, and an optional description and homepage URL.

A screenshot of the GitHub 'Create new repository' form. The form has two main sections: 'Owner' and 'Repository name'. The 'Owner' section shows the GitHub logo and a dropdown menu. The 'Repository name' section has a text input field (highlighted with a red box) and a note: 'Great repository names are short and memorable. Need inspiration?'. Below this is a 'Description (optional)' text area. At the bottom, there are two radio button options: 'Public' (selected) and 'Private'. The 'Public' option is described as 'Anyone can see this repository. You choose who can commit.' and the 'Private' option is described as 'You choose who can see and commit to this repository.' A large red arrow points from the right towards the 'Repository name' input field. A small text at the bottom right of the screenshot reads 'wiki: How to Create a Repository on GitHub'.

5. Select the “Public” option to create a free public repo that anyone can access, or select the “Private” option to upgrade to a paid account and create a private repository.



☒ **Public**
Anyone can see this repository. You choose who can commit.

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

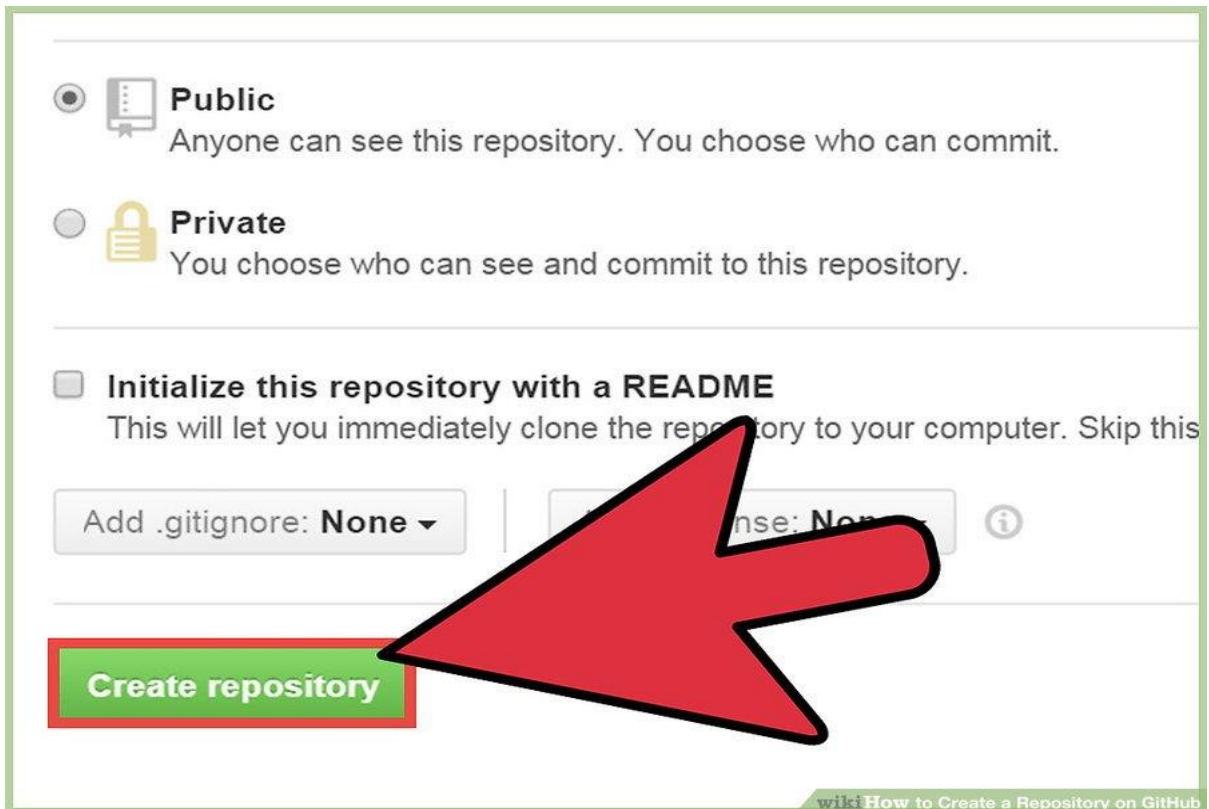
☐ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you want to add files manually.

Add .gitignore: **None** ▼ | Add a license: **None** ▼

Create repository

wiki How to Create a Repository on GitHub

6. Click the “Create Repository” button to create your repo.



☒ **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 want to add files manually.

Add .gitignore: **None** ▼ | Add a license: **None** ▼ ⓘ

Create repository

wiki How to Create a Repository on GitHub

Adding file to repository

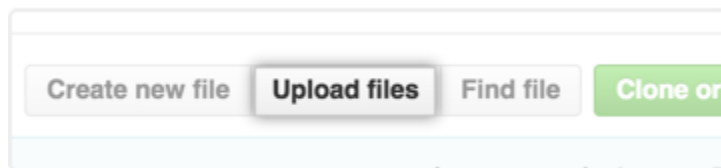
- You can upload and commit an existing file to a GitHub repository. Drag and drop a file to any directory in the file tree, or upload files from the repository's main page.
- Files that you add to a repository via a browser are limited to 25 MB per file. You can add larger files, up to 100 MB each, via the command line. For more information, see "[Adding a file to a repository using the command line](#)."

Tips:

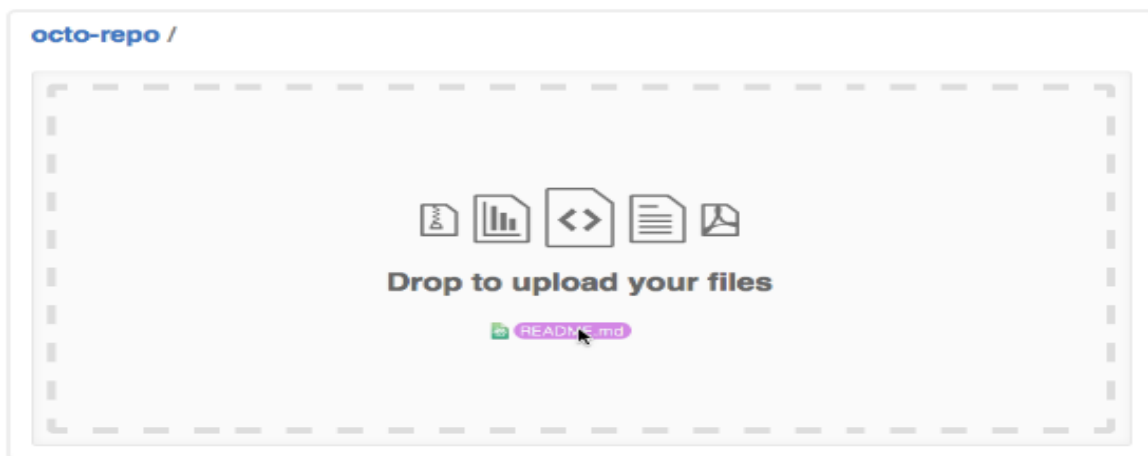
- You can upload multiple files to GitHub at the same time.
- If a repository has any [protected branches](#), you can't edit or upload files in the protected branch using the web interface.

1. On GitHub, navigate to the main page of the repository.

2. Under your repository name, click **Upload files**.

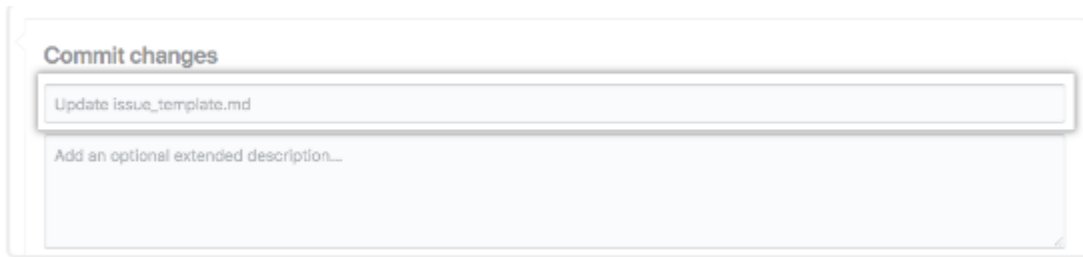


3. Drag and drop the file or folder you'd like to upload to your repository onto the file tree.



4. At the bottom of the page, type a short, meaningful commit message that describes the change you made to the file. You can attribute the commit to more than one author in

the commit message. For more information, see "[Creating a commit with multiple co-authors](#)."

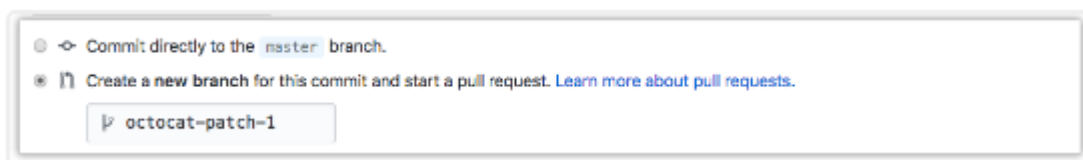


Commit changes

Update issue_template.md

Add an optional extended description...

5. Below the commit message fields, decide whether to add your commit to the current branch or to a new branch. If your current branch is master, you should choose to create a new branch for your commit and then [create a pull request](#).

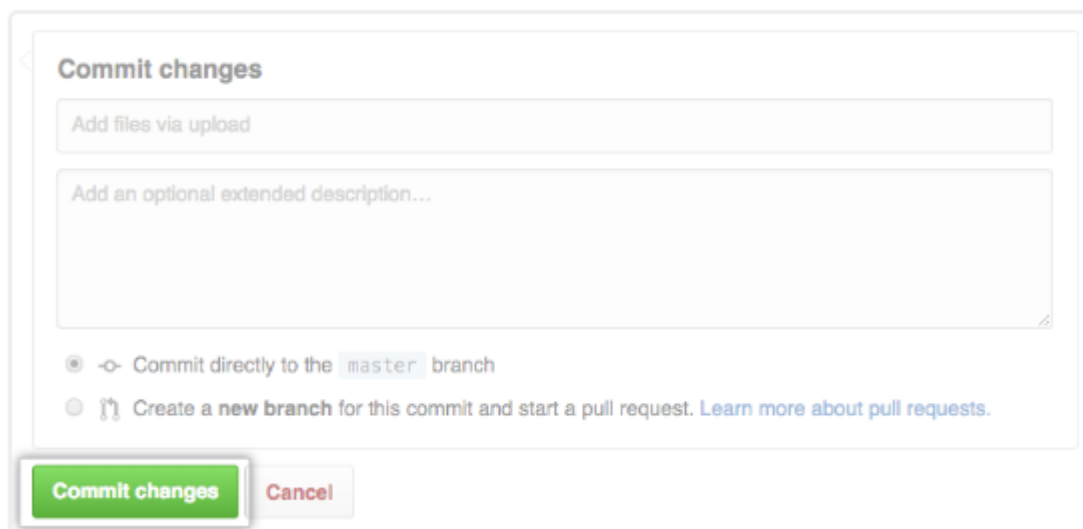


☐ Commit directly to the `master` branch.

☒ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

octocat-patch-1

- 6 Click **Commit changes**.



Commit changes

Add files via upload

Add an optional extended description...

☒ Commit directly to the `master` branch

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

- You can upload an existing file to a GitHub repository using the **command line**.
 - You have to create repository
1. On your computer, move the file you'd like to upload to GitHub into the local directory that was created when you cloned the repository.
 2. Open Git Bash.
 3. Change the current working directory to your local repository.
 4. Stage the file for commit to your local repository.

```
$ git add .
```

```
# Adds the file to your local repository and stages it for commit.
```

```
#To unstage a file, use 'git reset HEAD YOUR-FILE'.
```

5. Commit the file that you've staged in your local repository.

```
$ git commit -m "Add existing file"
```

```
# Commits the tracked changes and prepares them to be pushed to a remote  
#repository. To remove this commit and modify the file, use 'git reset --  
#soft HEAD~1' and commit and add the file again.
```

6. [Push the changes](#) in your local repository to GitHub.

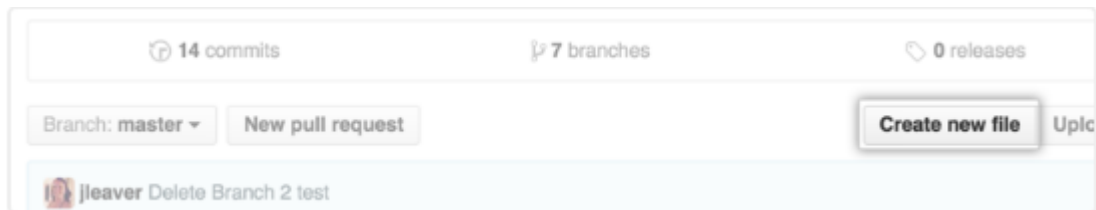
```
$ git push origin your-branch
```

```
# Pushes the changes in your local repository up to the remote repository
```

```
#you specified as the origin
```

Creating new files

- You can create new files directly on GitHub in any repository you have write access to.
1. On GitHub, navigate to the main page of the repository.
 2. In your repository, browse to the folder where you want to create a file.
 3. Above the file list, click **Create new file**.



4. In the file name field, type the name and extension for the file. To create subdirectories, type the / directory separator.



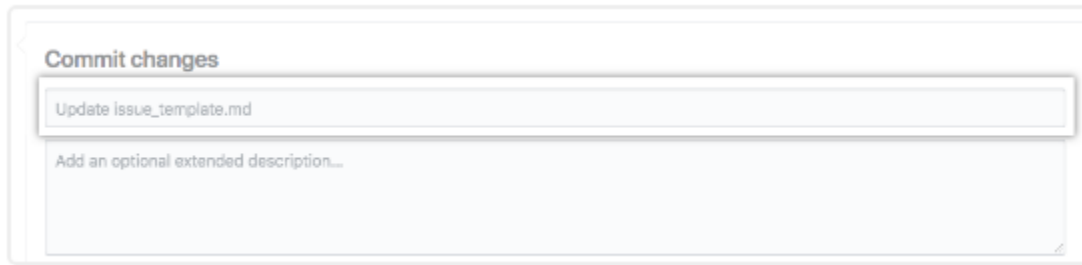
5. On the **Edit new file** tab, add content to the file.



6. To review the new content, click **Preview**.

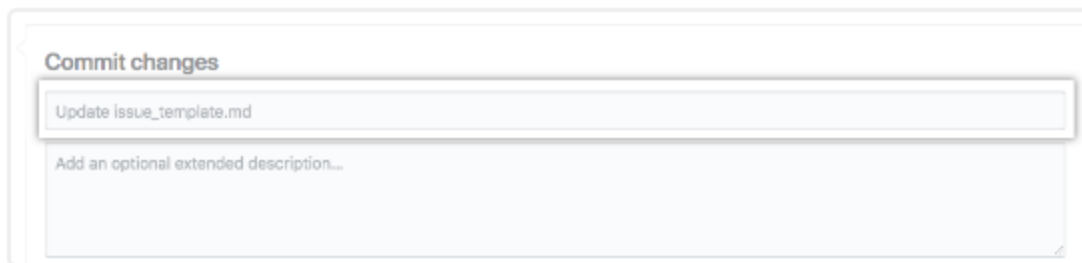


7. At the bottom of the page, type a short, meaningful commit message that describes the change you made to the file. You can attribute the commit to more than one author in the commit message. For more information, see "[Creating a commit with multiple co-authors](#)."

A screenshot of the GitHub 'Commit changes' form. It features a title 'Commit changes' in bold. Below the title is a text input field containing 'Update issue_template.md'. Underneath the input field is a larger text area with the placeholder text 'Add an optional extended description...'. The form has a light gray border and a subtle shadow.

Deleting files

- You can delete any file within your repositories on GitHub.
 1. Browse to the file in your repository that you want to delete.
 2. At the top of the file, click delete
 3. At the bottom of the page, type a short, meaningful commit message that describes the change you made to the file. You can attribute the commit to more than one author in the commit message. For more information, see ["Creating a commit with multiple co-authors."](#)

A screenshot of the GitHub 'Commit changes' form, identical to the one above. It shows the 'Commit changes' title, a text input field with 'Update issue_template.md', and a text area with the placeholder 'Add an optional extended description...'.