**SDM COLLEGE OF ENGINEERING AND TECHNOLOGY**

Dhavalagiri, Dharwad-580002, Karnataka State, India.

**Email: cse.sdmcet@gmail.com**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**A** [**R**](https://docs.google.com/document/d/18cG_WN77Px7SVJtc39twRAwvXtnD0ErqwRTTDQc0XUg/edit#bookmark=id.30j0zll)**eport**

**On**

**GitHub – Collaborative Code Hub**

**COURSE CODE: 22UCSE421**

**COURSE TITLE: Project Management Tools**

**SEMESTER: IV**

**DIVISION: CSE-A**

**COURSE TEACHER:  Prof. Yashoda S.**

  
**[ Academic Year- 2023-24 ]**

**Date of Submission:  04/06/2024**

Submitted By:

**USN NAME**

2SD22CS011 Ananya D Prabhu

2SD22CS012 Ananya U Gaonkar

2SD22CS013 Ankith S Naik

2SD22CS014 Anusha A Bennal

2SD22CS015 Anusha U Hegde

**Table of Contents:**

**1. ABOUT GITHUB:**

GitHub is a hosting platform used by developers to save their code online and track its changes to see what will work when it comes to them working on their projects. GitHub provides an array of functionalities and boasts a supportive community to help any new developers just starting GitHub. It is a code-hosting platform where users can update codes or files and track their updates.

GitHub allows developers to create and manage the code in the repository in the remote location where others can access the code or GitHub is a collection of repositories which contains the files of the project.

**Two connected principles to understand GitHub:**

* **Version control System (VCS):**

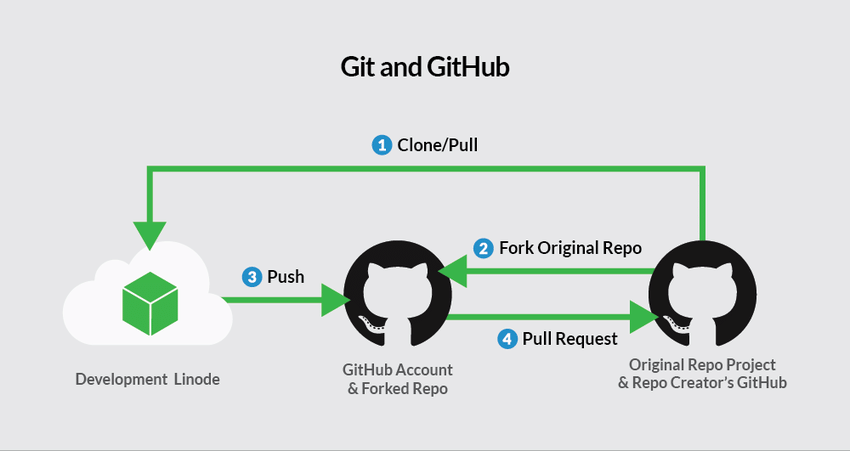
It is a tool used to record any changes made in files and keeps track of any changes made in a file or folder. The most popular Version Control System (VCS) is Git and its platform, GitHub. It provides a version control system called Git which is used to track any updates or commit changes performed in a user’s repository.

* **Git:**

Git is an open-source distributed version control system. It is designed to handle minor to major projects with high speed and efficiency. It is developed to co-ordinate the work among the developers. The version control allows us to track and work together with our team members at the same workspace.

**How do Git and GitHub work together?**

GitHub hosts Git repositories and provides developers with tools to ship better code through command line features, issues (threaded discussions), pull requests, code review, or the use of a collection of free and for-purchase apps in the GitHub Marketplace.

**Fig.:** Git and GitHub Workflow

**How They Work Together:**

1. **Clone Repository**: Developers can clone a repository from GitHub to their local machine using Git. This creates a local copy of the project that includes the entire version history.

git clone https://github.com/username/repository.git

1. **Make Changes Locally**: Developers make changes to the code in their local repository, committing those changes to keep track of different versions.

git add .

git commit -m "Description of changes"

1. **Push Changes**: After making changes locally, developers push their changes to the remote repository on GitHub, making them available to other collaborators.

git push origin branch-name

1. **Pull Requests**: When a developer wants to contribute to a project, they can create a fork of the repository, make changes in their fork, and then submit a pull request to the original repository. The project maintainer reviews the changes and can merge them into the main project.

Steps for creating a pull request:

git fork repository

git clone https://github.com/username/forked-repository.git

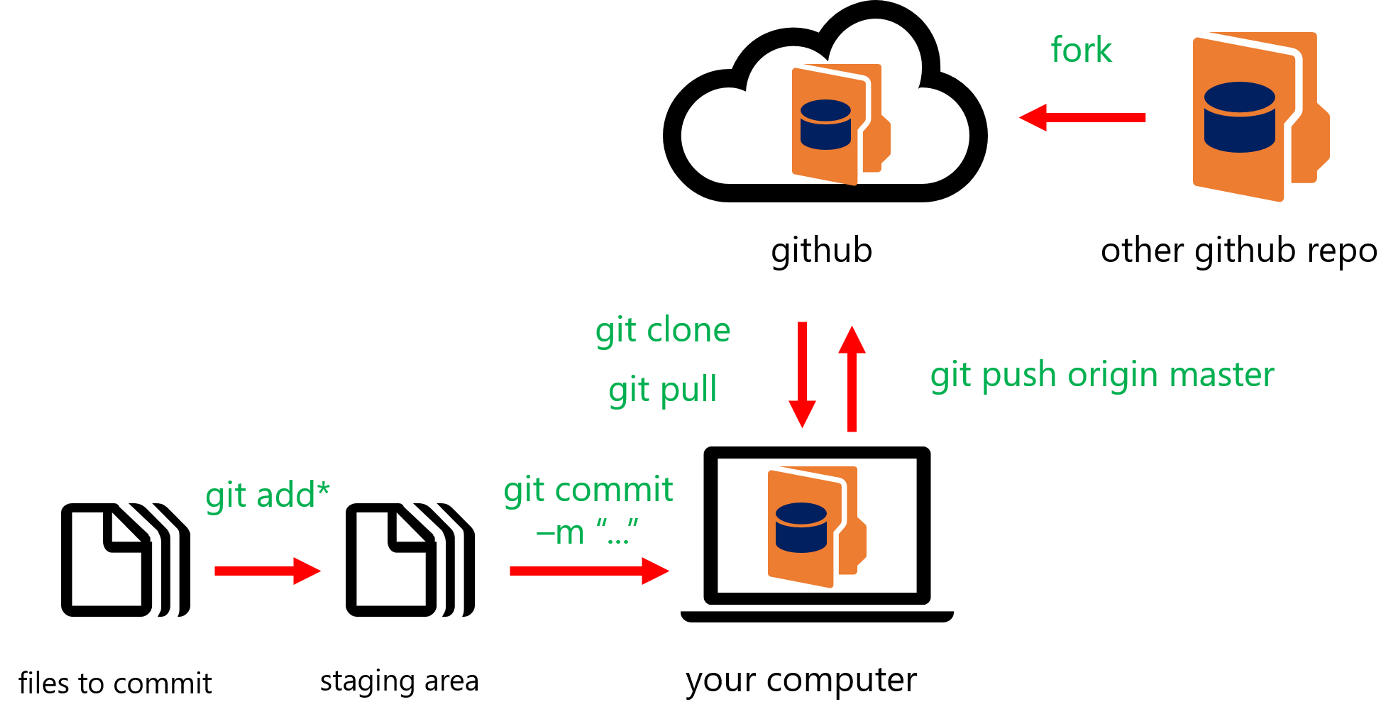
git checkout -b feature-branch

Make changes:

git push origin feature-branch

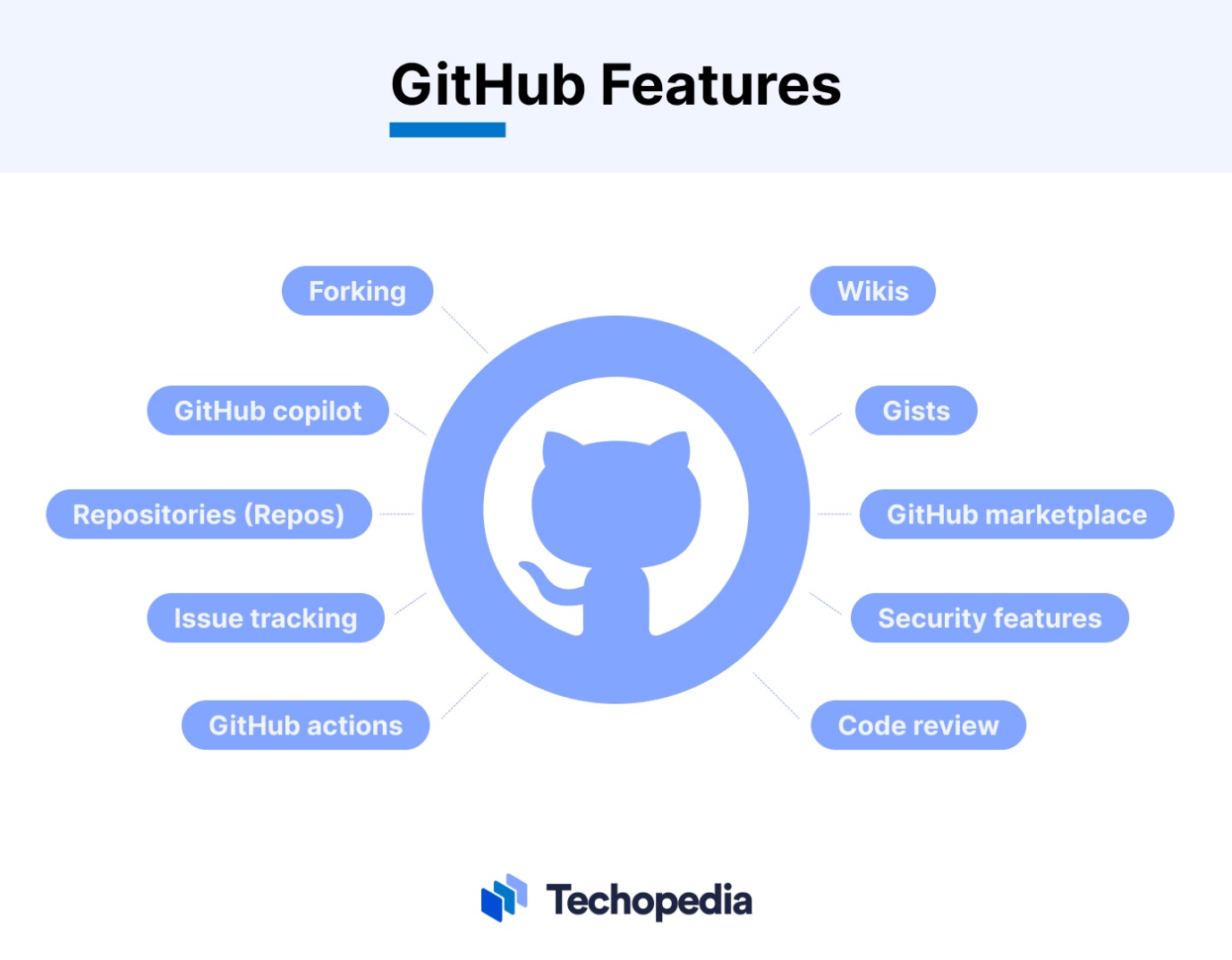
Create a pull request on GitHub.

1. **Collaboration and Review**: GitHub's features, such as code review and inline comments, make it easy for teams to collaborate and review each other's work before merging changes.

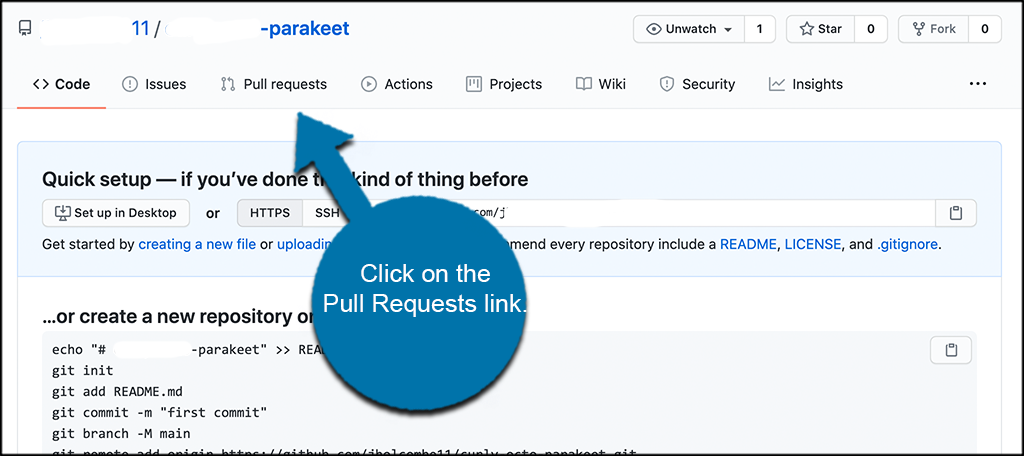


**Fig.:** Stages of Workflow

**2. Features of GitHub:**

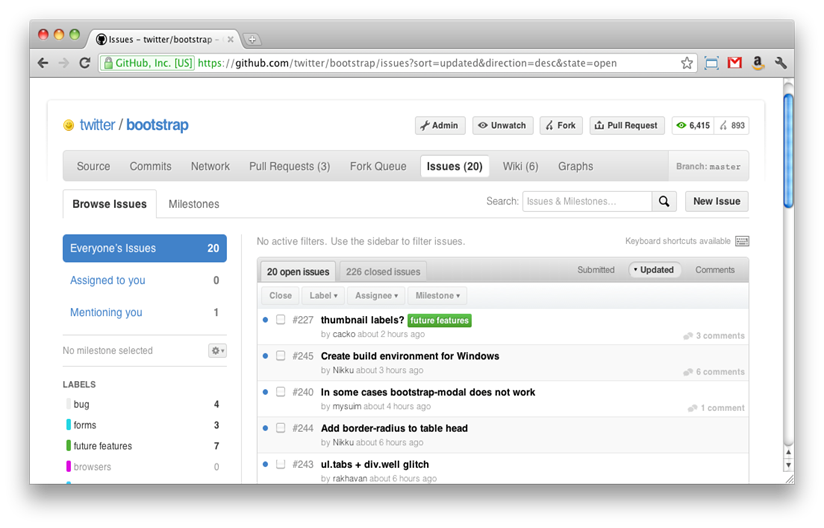


**Fig.:** GitHub Features

1. **Collaboration Features:**
   1. **Pull requests**: This is a formal way for developers to propose changes to the codebase. Others can review the proposed changes and discuss them before merging them into the main project.

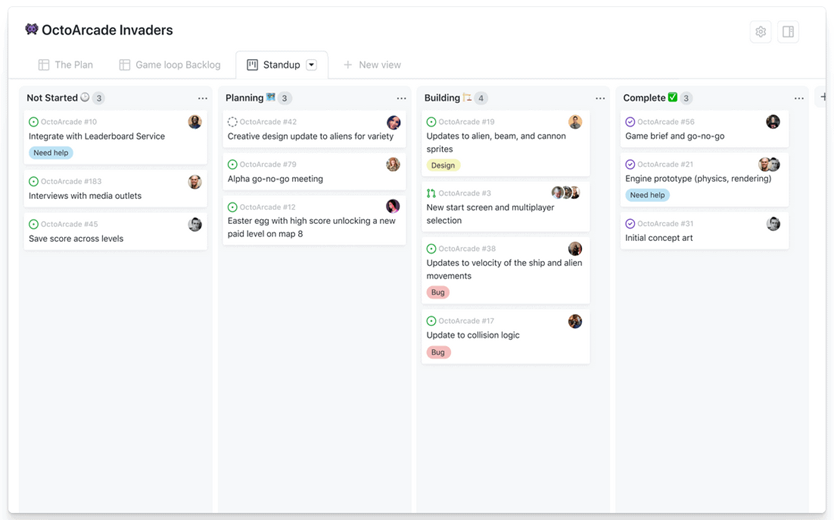
**Fig.:** Pull Request

* 1. **Issue tracking**: Manage bugs, tasks, and feature requests. Assign them to team members and track their progress.



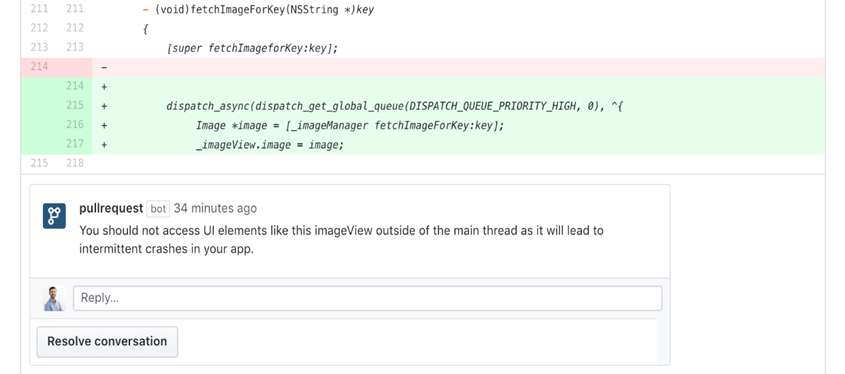
**Fig.:** Issue Tracking

* 1. **Project management tools**: Organize your workflow with features like project boards, kanban boards, and wikis.



**Fig.:** GitHub asProject Management Tool

* 1. **Code review**: Facilitate code quality and knowledge sharing by allowing team members to review each other's code.



**Fig.:** Code Review in GitHub

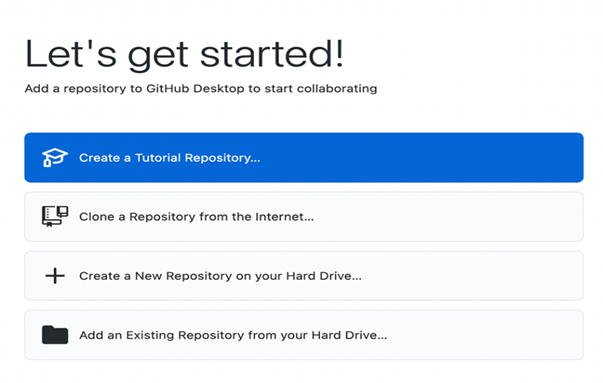
1. **Creating a Repository:**
   1. Go to GitHub and log in to your account. In the upper-right corner, click the “+” icon, then select “New repository.”
   2. **Fill in the repository details:**

Repository name: Choose a short, memorable name.

Description: Optionally, add a brief description.

Visibility: Decide whether it’s public or private.

* 1. **Initialize with a README:** If you want to create a README file, check this box.
  2. **Add gitignore:** If you are working with a specific language or framework, select an appropriate .gitignore template.



**Fig.:** Creating Repository

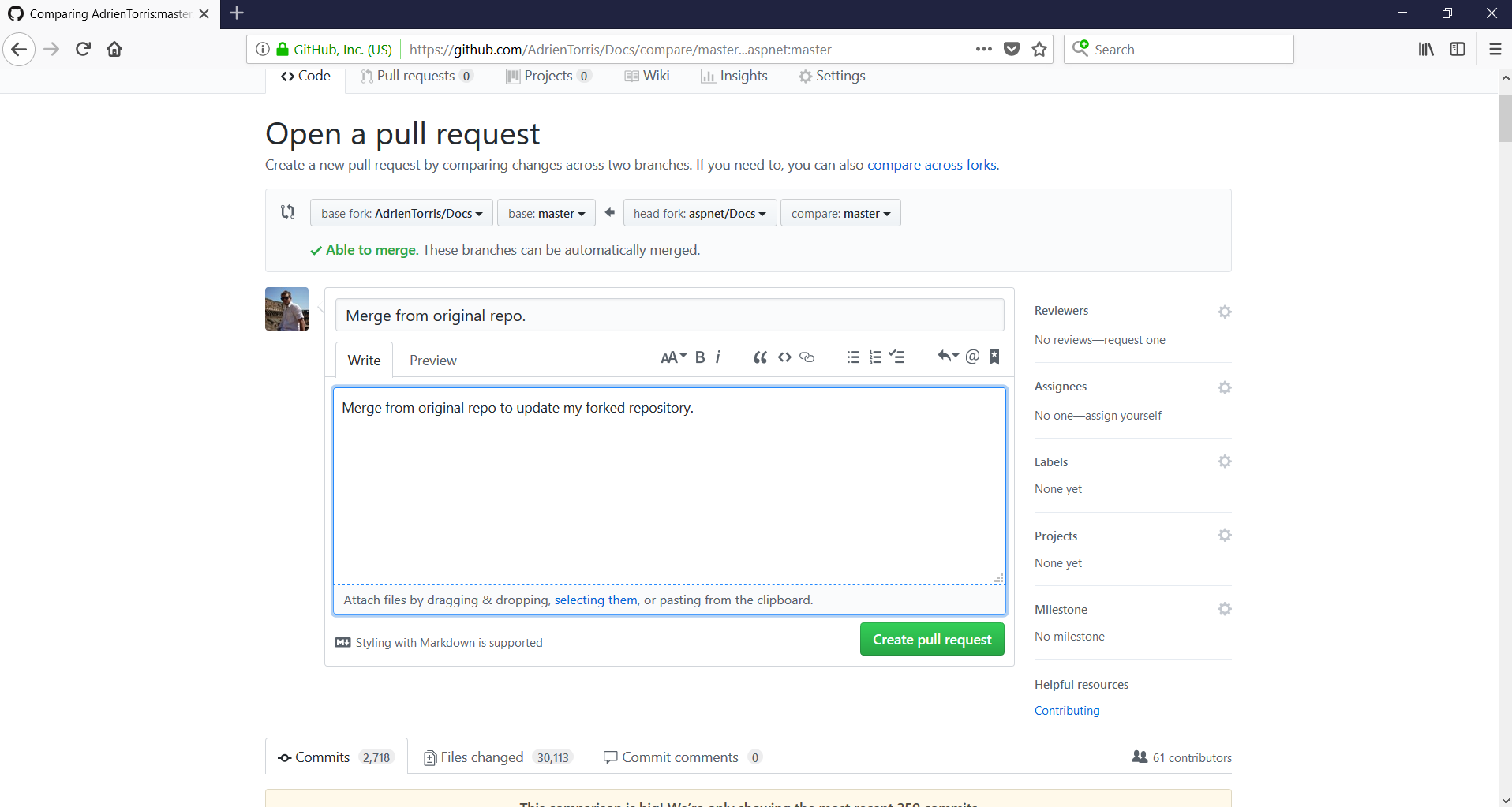
1. **Updating a Repository:**

To update an existing repository, follow these steps:

* 1. **Clone the Repository:** If you haven’t already, clone the repository to your local machine

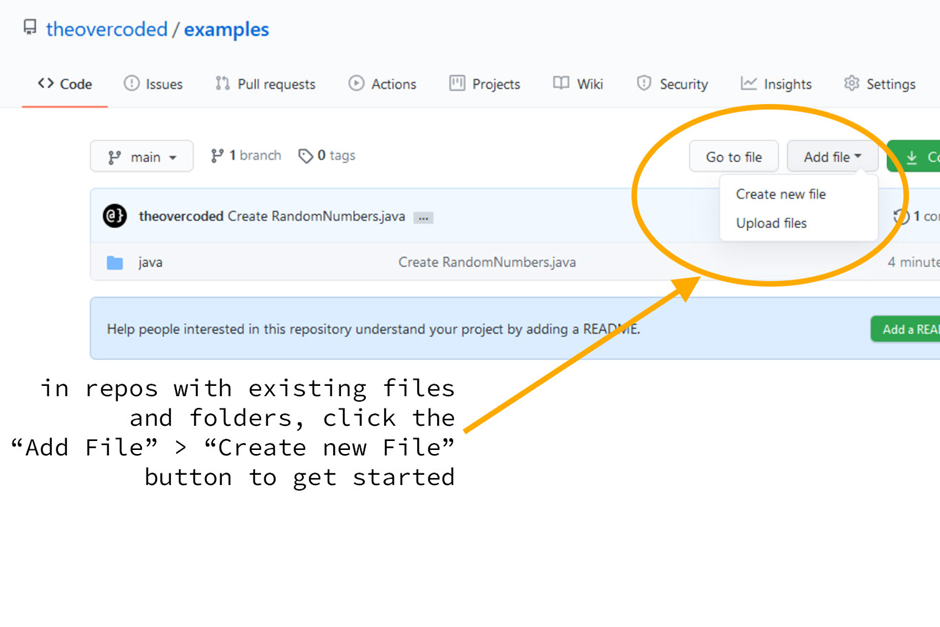
git clone [remote repository URL]

* 1. **Changes:** Edit files, add new ones, or make any necessary updates.
  2. Add Changes to the Next Commit
  3. Commit Your Changes
  4. Push Changes to GitHub: Push your changes to the remote repository.



**Fig.:** Updating GitHub Repository

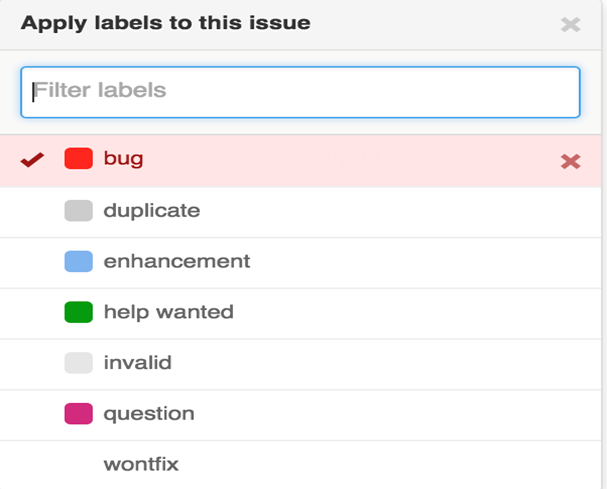
1. **Add files to your repository**
   1. **Navigate to your repository:** Go to your GitHub account and open the desired repository.
   2. **Add files:** Click the "Add file" dropdown menu near the top of the page. You'll see two options:
   3. **Upload files:** Click this option to select files from your computer and upload them directly to the repository. You can drag and drop files or use the file selection dialog.
   4. **Create new file**: This option allows you to create a new file directly within the GitHub web interface. Provide a filename and content, and it will be added to the repository.
   5. **Commit changes:** Once you've added your files, you'll need to commit them. Write a clear and concise message describing the changes you've made. Optionally, you can choose a branch to commit to. Finally, click the "Commit changes" button.



**Fig.:** Adding Files to GitHub

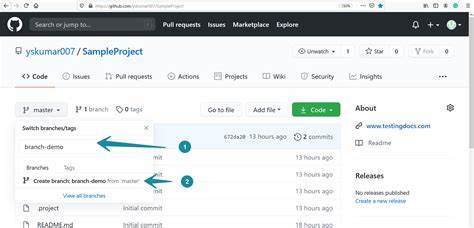
1. **Creating Labels:** Anyone with write access to a repository can create labels. You can customize the label name, description (to explain the purpose of the label), and choose a distinct color for better visual identification.

Applying Labels: Labels are applied to individual issues or pull requests. This allows you to categorize issues by priority

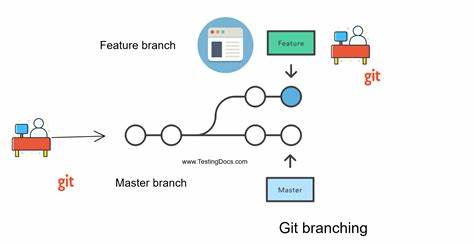


**Fig.:** Labels in GitHub

1. **Operations in GitHub:**
   1. **Branches:** GitHub creates the repository with a single branch. This first branch in the repository is the default branch. The default branch is the branch that GitHub displays when anyone visits your repository. The default branch is also the initial branch that Git checks out locally when someone clones the repository. Unless you specify a different branch, the default branch in a repository is the base branch for new pull requests and code commits.



**Fig.:** Branches in GitHub

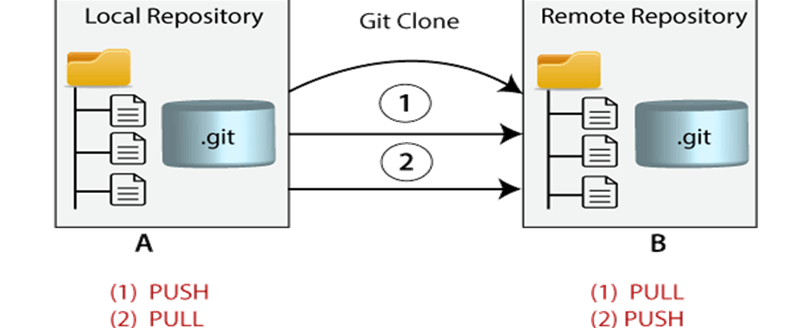


**Fig.:** Git Branching

* 1. **Cloning repositories:** Cloning a repository in GitHub refers to creating a local copy of a remote repository hosted on GitHub on your computer. This allows you to work on the project files offline and then push your changes back to the remote repository when you're ready.

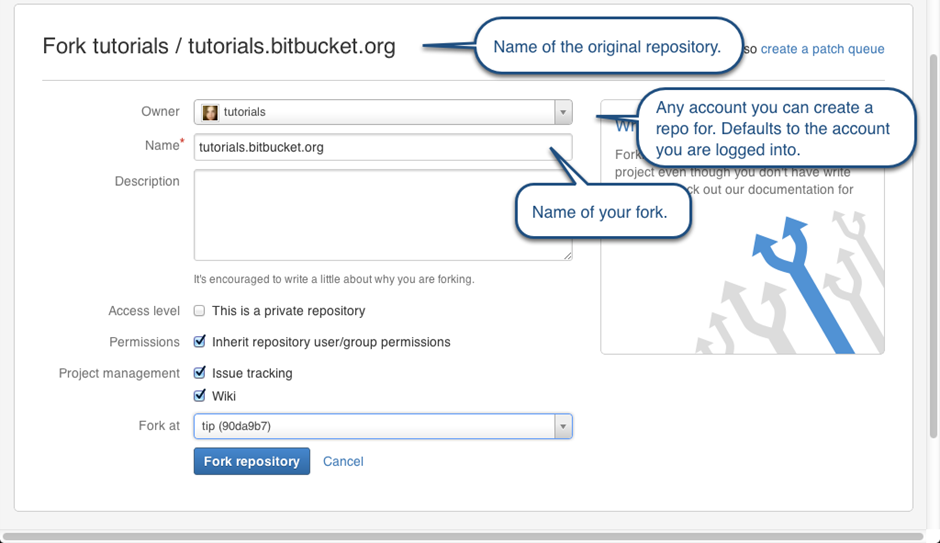
Go to the desired repository on GitHub.

Click on the "Code" button.

Under the "clone" section, you'll see options for HTTPS, SSH, or GitHub CLI. Choose the method that suits your setup. Each method will provide a corresponding URL that you'll need in the next step. 

**Fig.:** Cloning Repositories

* 1. **Forking a repository:** Forking a repository on GitHub creates a copy of an existing repository under your own account.



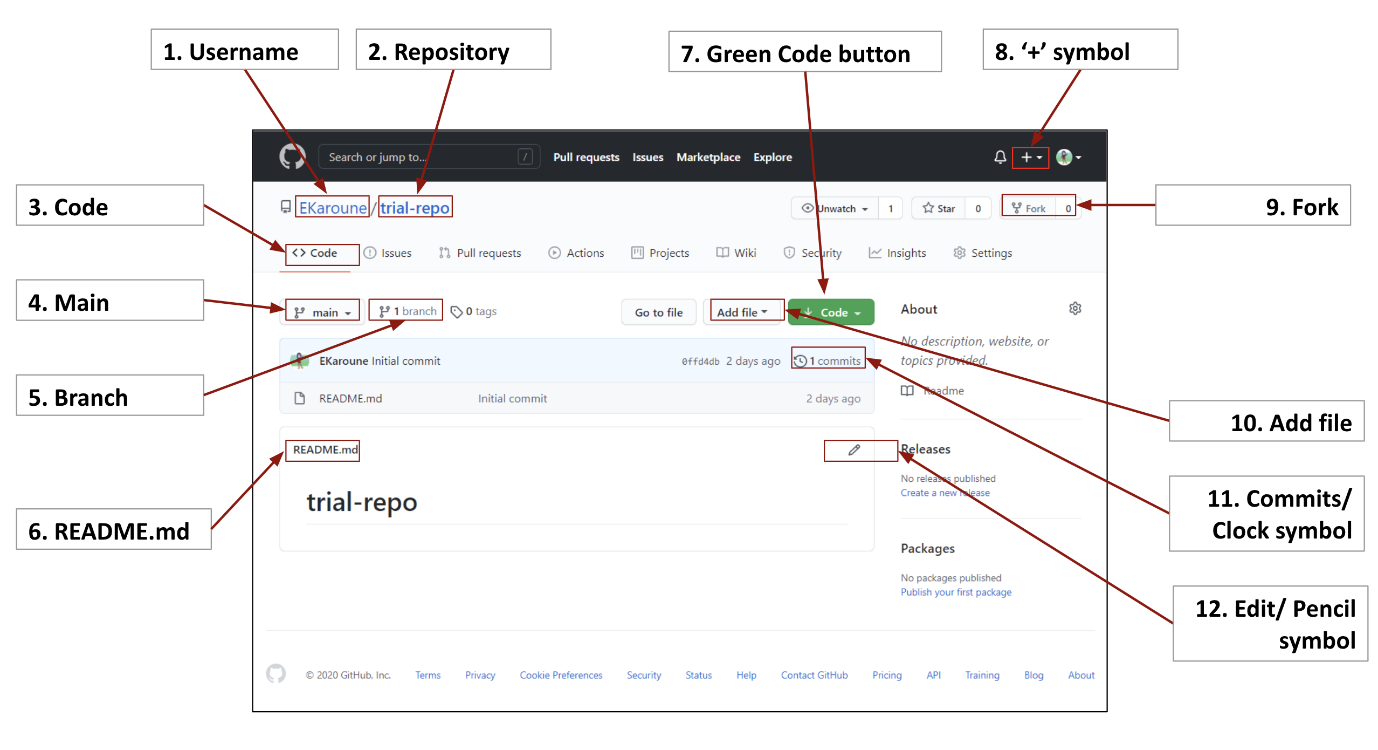
**Fig.:** Forking a repository

1. In the comment body field, type a description of your issue.

2. If you're a project maintainer, you can assign the issue to someone, add it to a project (classic), associate it with a milestone, or apply a label.

3. When you're finished, click Submit new issue.

**FLOWCHART – FEATURES OF GITHUB**



**How to Install Git on Windows:**