

## **LAB # 1**

### **Introduction to GitHub**

#### **OBJECTIVE**

Introduction to GitHub by creating Git repository and cloning its configuration to eclipse project.

#### **You will learn about:**

- Basics of how GitHub works.

GitHub allows multiple developers to work on a single project at the same time, reduces the risk of duplicative or conflicting work, and can help decrease production time. With GitHub, developers can build code, track changes, and innovate solutions to problems that might arise during the site development process simultaneously. Non-developers can also use it to create, edit, and update website content.

Some of the common terms teams will need to understand when using GitHub are:

**Repository:** Every project on GitHub is called repository. It is a folder for your project that contains all files and their revision histories.

**Branch:** a workspace in which you can make changes that won't affect the live site.

**Commit Changes:** a saved record of a change made to a file within the repository.

**Pull Request (PR):** the way to ask for changes made to a branch to be merged into another branch that also allows for multiple users to see, discuss and review work being done.

**Merge:** after a pull request is approved, the commit will be pulled in (or merged) from one branch to another and then, deployed on the live site

**Issues:** how work is tracked when using git. Issues allow users to report new tasks and content fixes, as well as allows users to track progress on a project board from beginning to end of a specific project.

#### **STEPS OF CREATING GIT REPOSITORY:**

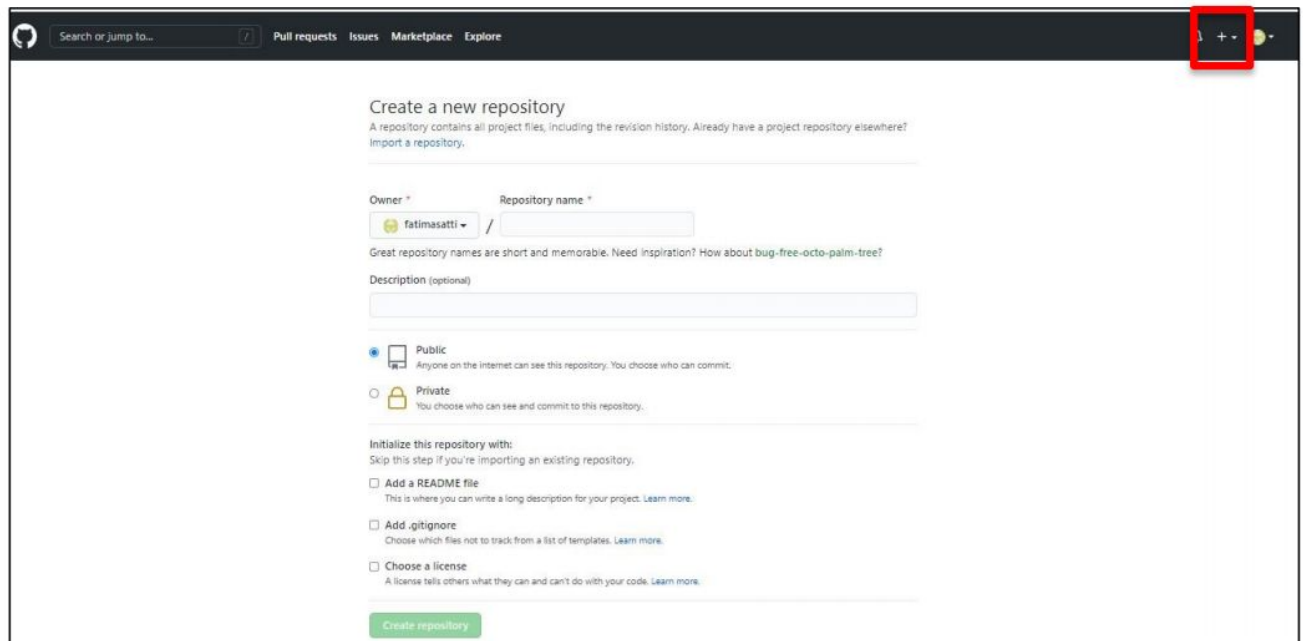
**Step # 01:** Create a GitHub account by clicking on link and sign up.

**<https://github.com/>**

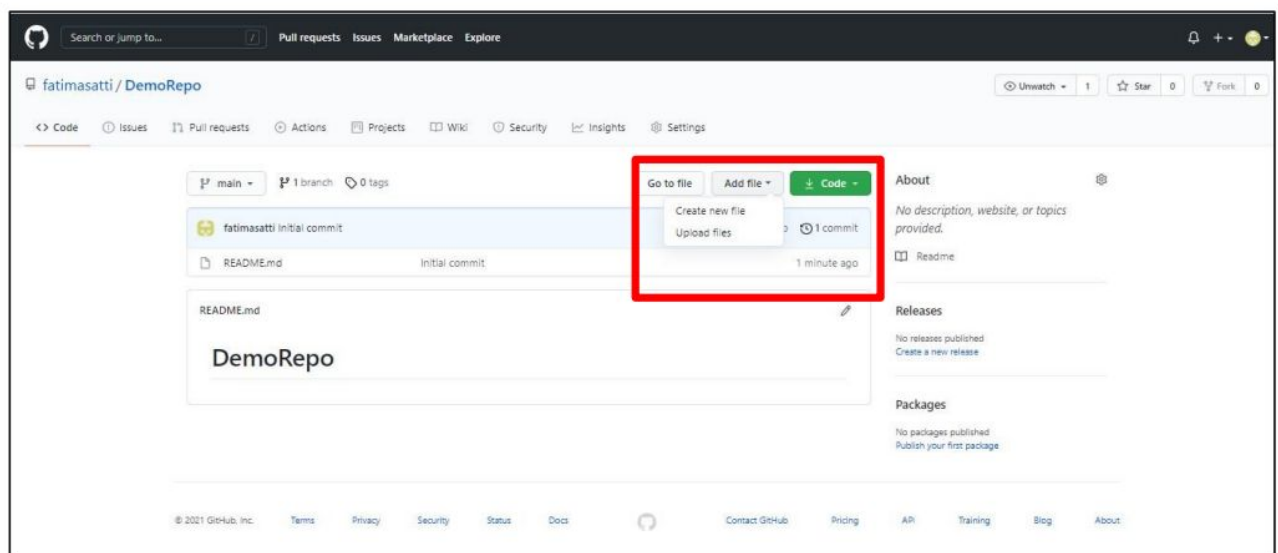
**Step # 2:** Create a repository by clicking (+) sign or by clicking the link below:

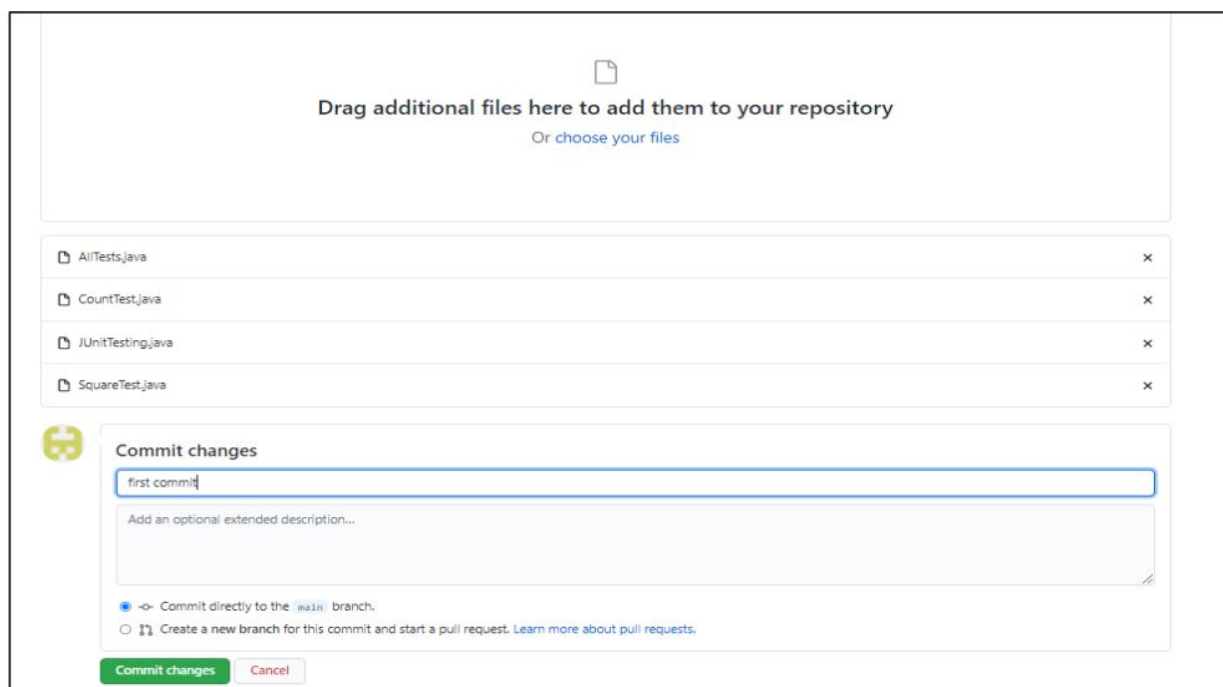
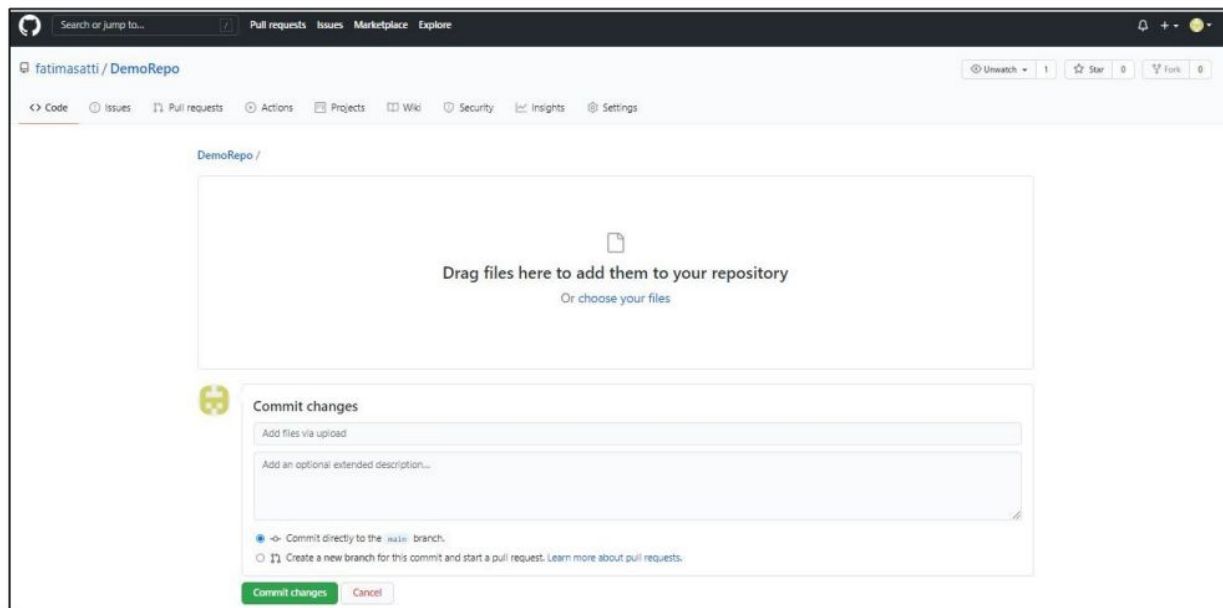
**<https://github.com/new>**

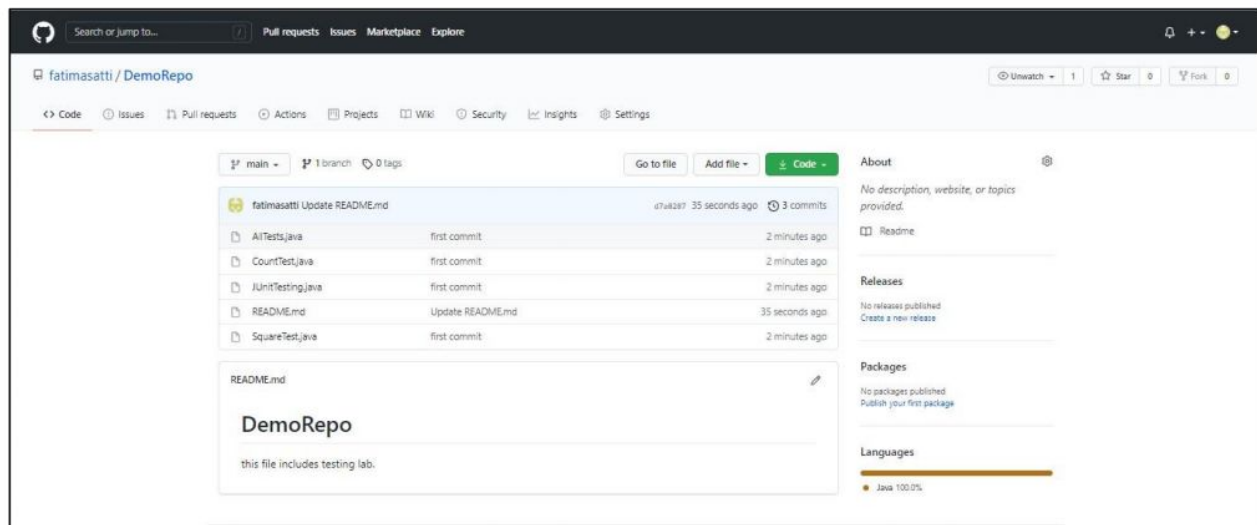
- Enter your Repository name.
- You can also add description (optional).
- Choose whether you want to make your repository Public or Private.
- Add a README file (optional).
- Click on Create repository.



**Step # 3: Upload files from any eclipse workplace folder and commit changes.**

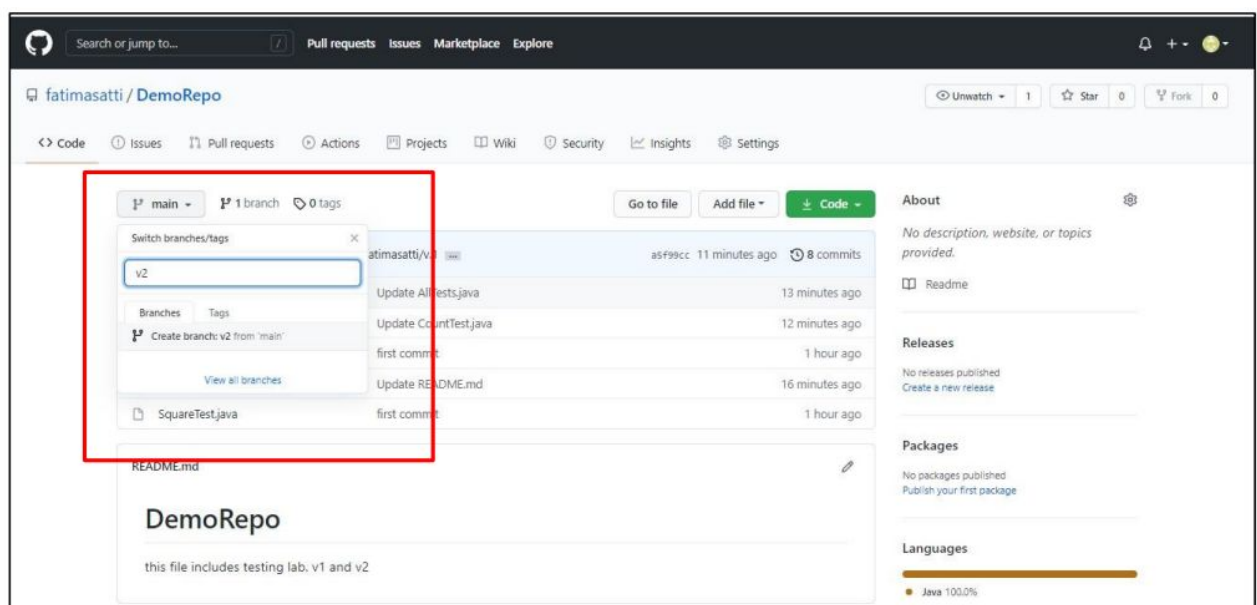






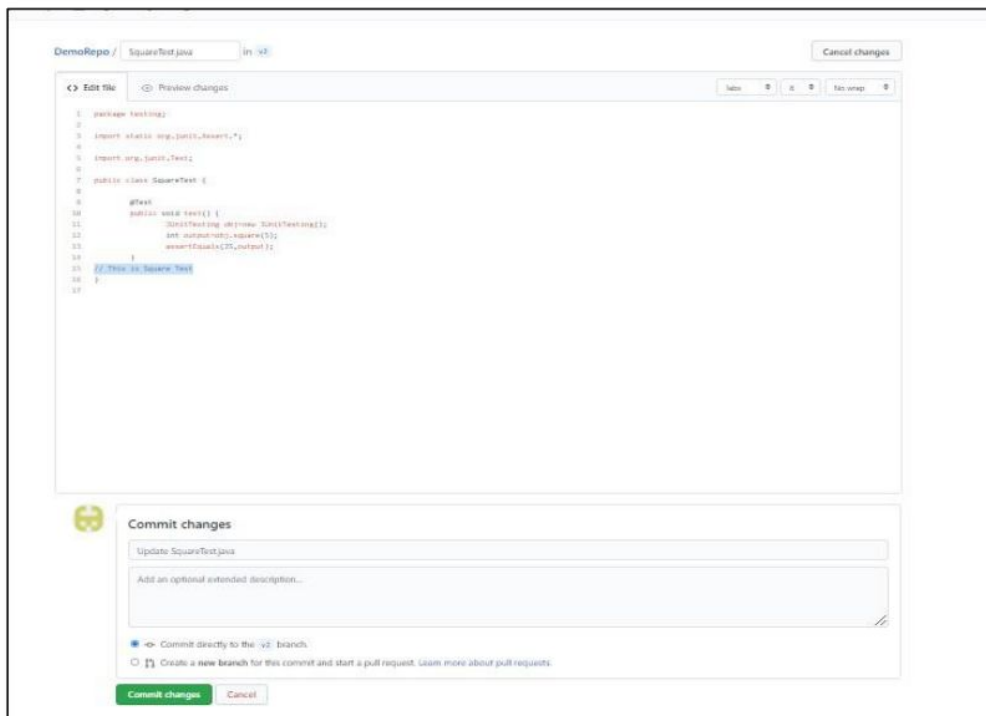
You can now see that all the files are uploaded in your current repository.

**Step # 4: Create a new branch and make changes in any file. Perform Commit, Pull requests and merge.**



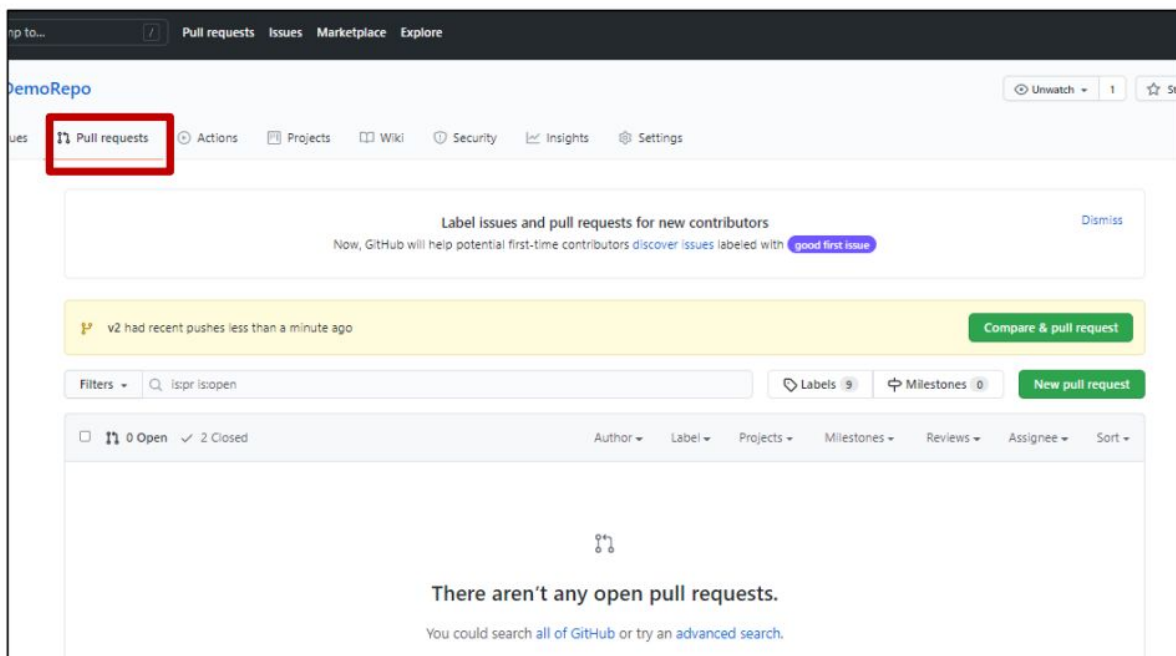
(In above screenshot, you can see a branch named **V2** is created).

Update any file in V2, then select Commit changes.

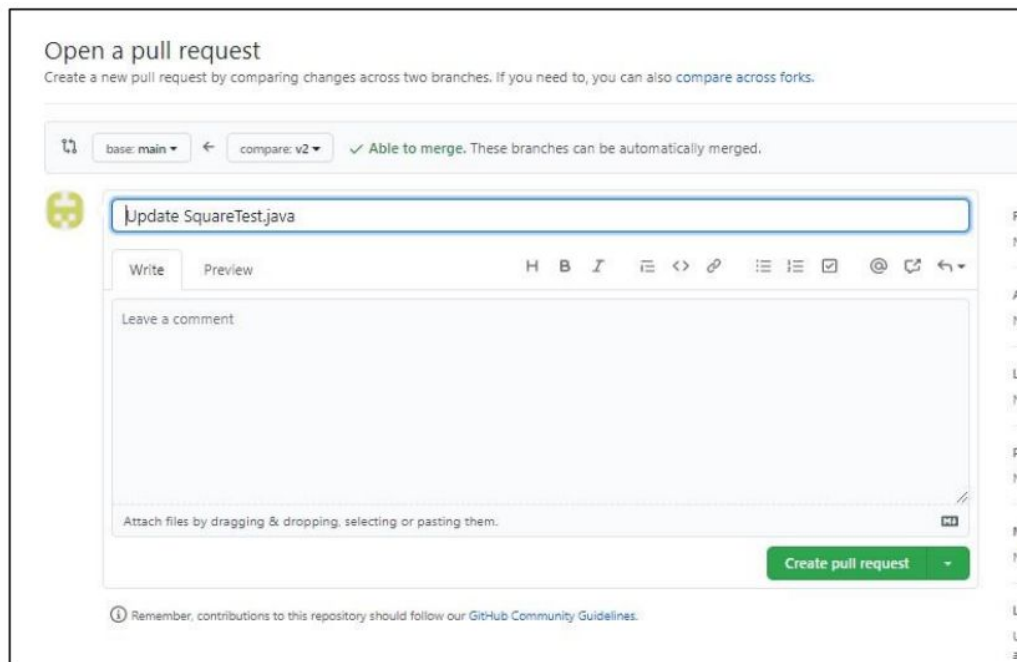


**Go to Pull Requests tab, select Compare and Pull request.**

// This will compare your V2 edited file with main branch and then will commit the changes.

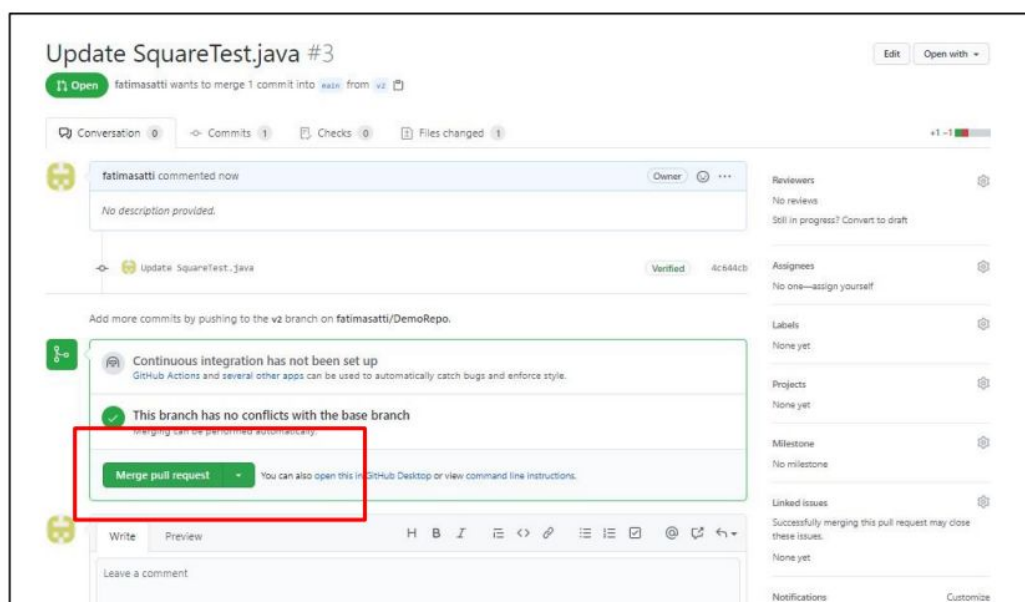


## Enter create pull request



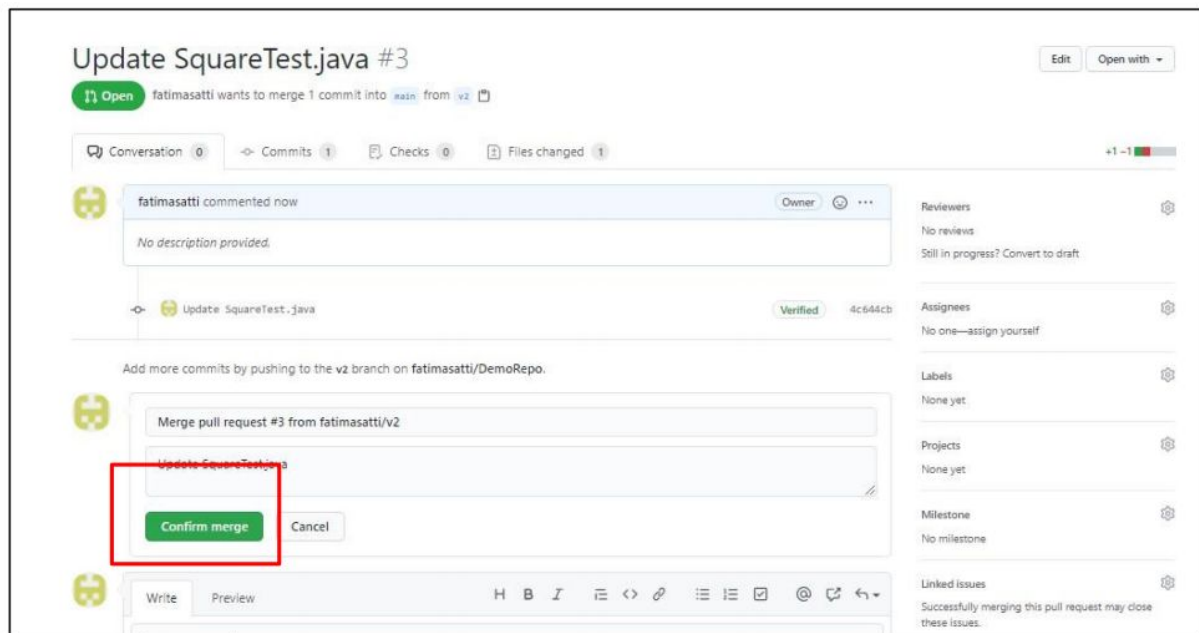
The screenshot shows the GitHub interface for creating a pull request. At the top, it says "Open a pull request" and "Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#)." Below this, there are dropdowns for "base: main" and "compare: v2", with a green checkmark and the text "Able to merge. These branches can be automatically merged." The main text area has a title "Update SquareTest.java" and a "Write" tab selected. Below the text area is a "Leave a comment" section and a "Create pull request" button. At the bottom, there is a note: "Remember, contributions to this repository should follow our [GitHub Community Guidelines](#)."

## Select Merge Pull request

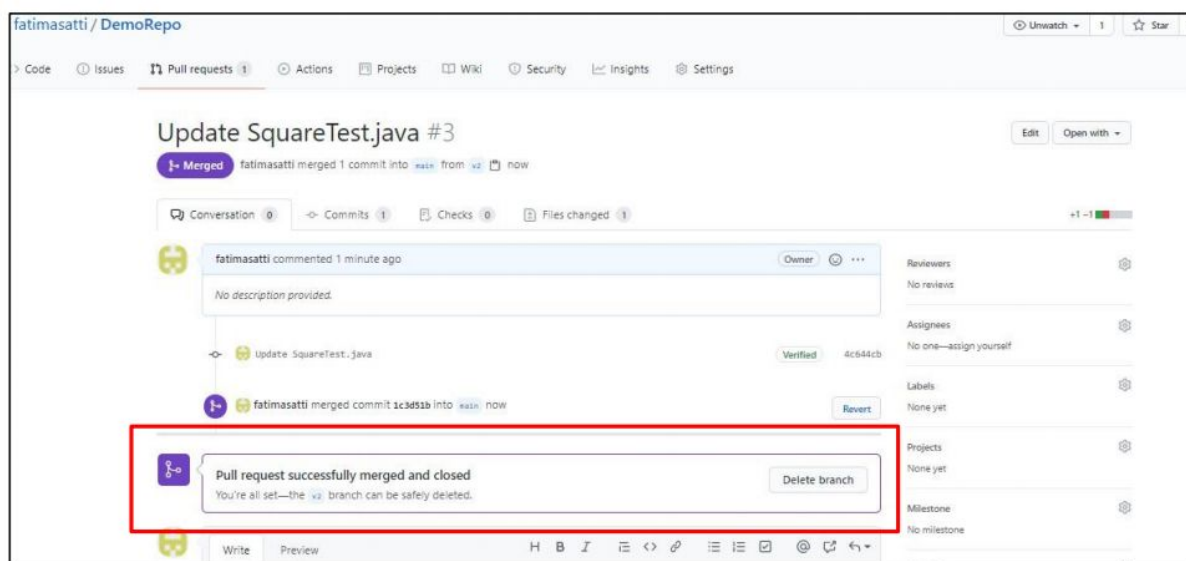


The screenshot shows the GitHub pull request page for "Update SquareTest.java #3". The page title is "Update SquareTest.java #3" with "Edit" and "Open with" buttons. Below the title, it says "fatimasatti wants to merge 1 commit into [main](#) from [v2](#)". The page has tabs for "Conversation", "Commits", "Checks", and "Files changed". The "Conversation" tab is selected, showing a comment from "fatimasatti" with the text "No description provided." Below the comment, there is a section for "Update SquareTest.java" with a "Verified" status and a commit hash "4c844cb". A green checkmark indicates "This branch has no conflicts with the base branch". A red box highlights the "Merge pull request" button. To the right of the main content, there are sections for "Reviewers", "Assignees", "Labels", "Projects", "Milestone", "Linked issues", and "Notifications".

## Confirm Merge



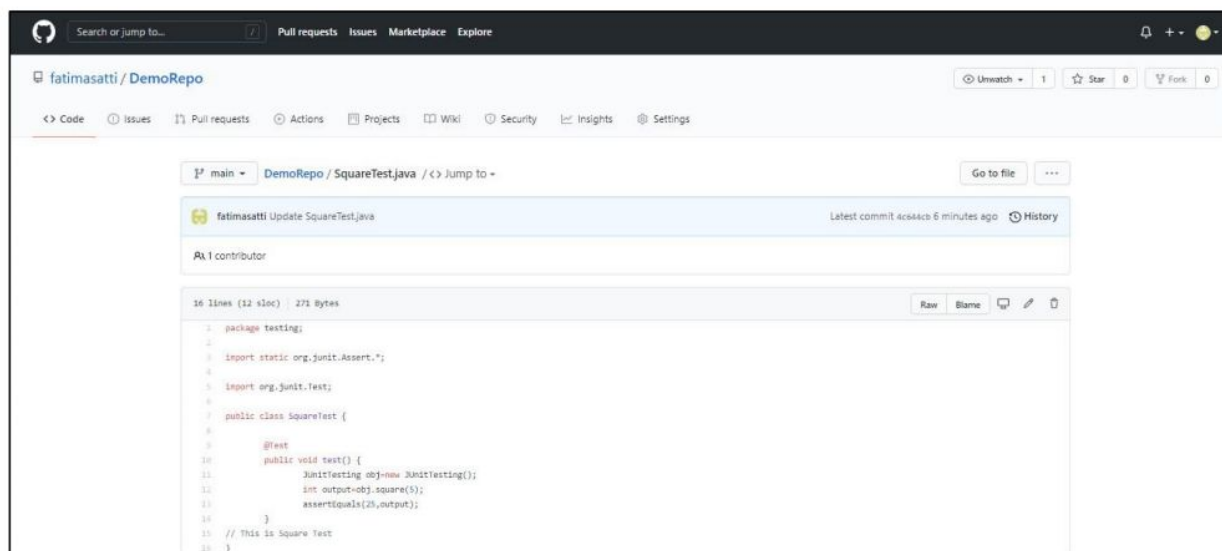
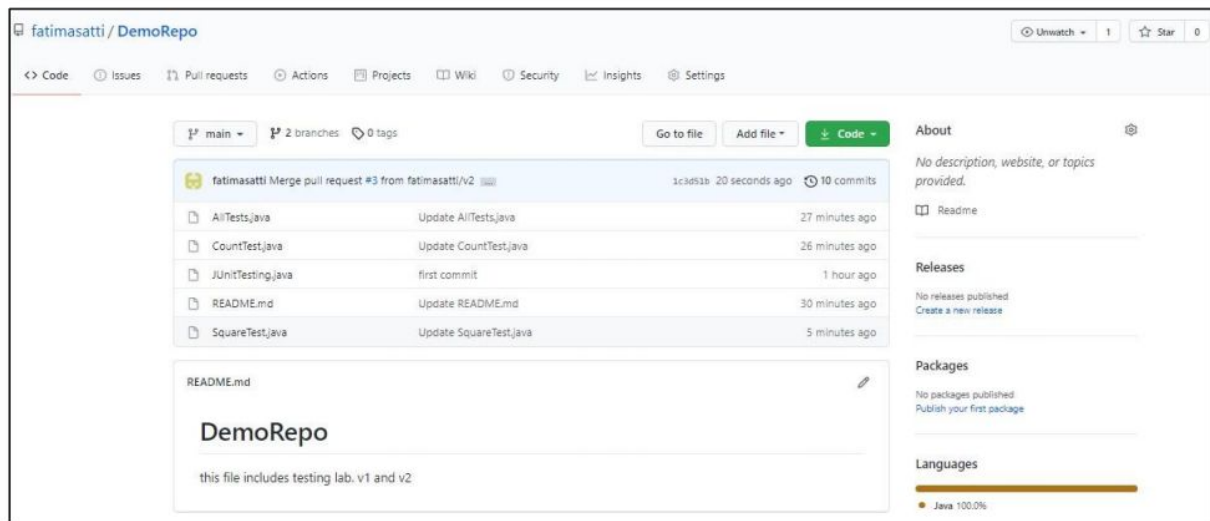
**You can see that Pull request is successfully merged and closed.**



You can now delete V2 branch as it is successfully merged into main branch.

SquareTest file is now updated.





### Lab Task:

Create account on GitHub.

Upload any of your Java Project files and perform commit, push, pull and merge operations.