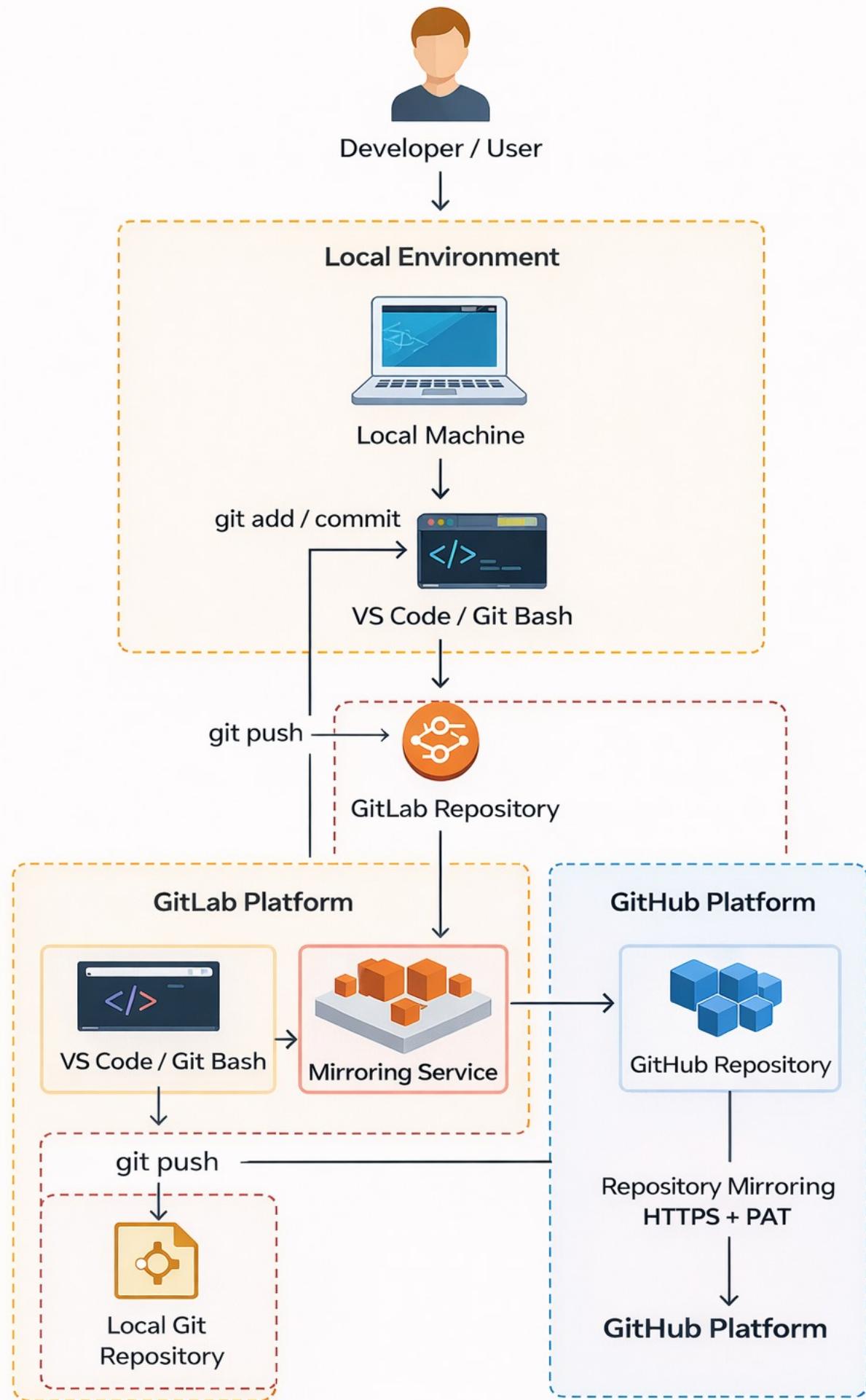


Mirroring Of Repository From Local Machine --GitLab and GitHub

Introduction

Repository mirroring is a technique used to keep multiple repositories synchronized automatically. In this project, a repository is maintained on GitLab as the main source, and GitHub is configured as a mirror repository. Any commit pushed from the local machine to the GitLab repository is automatically mirrored to the GitHub repository without requiring a separate push command. This approach helps in backup, platform availability, and collaboration across different Git platforms.

Architecture Diagram



Tools and Platforms Used

- Local Machine (Windows)
- Git Bash
- VS Code
- GitHub
- GitLab

Step-by-Step Mirroring Workflow

Step 1 : Create Repository on GitLab

1. Login to GitLab
2. Click New Project / Repository
3. Enter repository name
4. Create repository

The screenshots illustrate the steps to create a new repository on GitLab:

- Create new project:** The first screenshot shows the 'Create new project' interface. It offers two options: 'Create blank project' and 'Create from template'. Both options have a large plus icon.
- Create blank project:** The second screenshot shows the 'Create blank project' details page. The user has filled out the following fields:
 - Project name:** Gitlab-Mirror-Repo
 - Project URL:** https://gitlab.com/pranav-khade-tech
 - Project slug:** gitlab-mirror-repo
 - Visibility Level:** Public (selected)

Step 2 : Create Repository on GitHub

1. Login to GitHub
2. Click New Repository
3. Enter repository name
4. Create repository

The screenshot shows the GitHub repository creation interface. It consists of two main sections: 'General' (step 1) and 'Configuration' (step 2).
In the 'General' section, the 'Repository name' field contains 'Github-Mirror-Repo'. A red arrow points to this field. Below it, a note says 'Github-Mirror-Repo is available.'
In the 'Configuration' section, there are three settings:

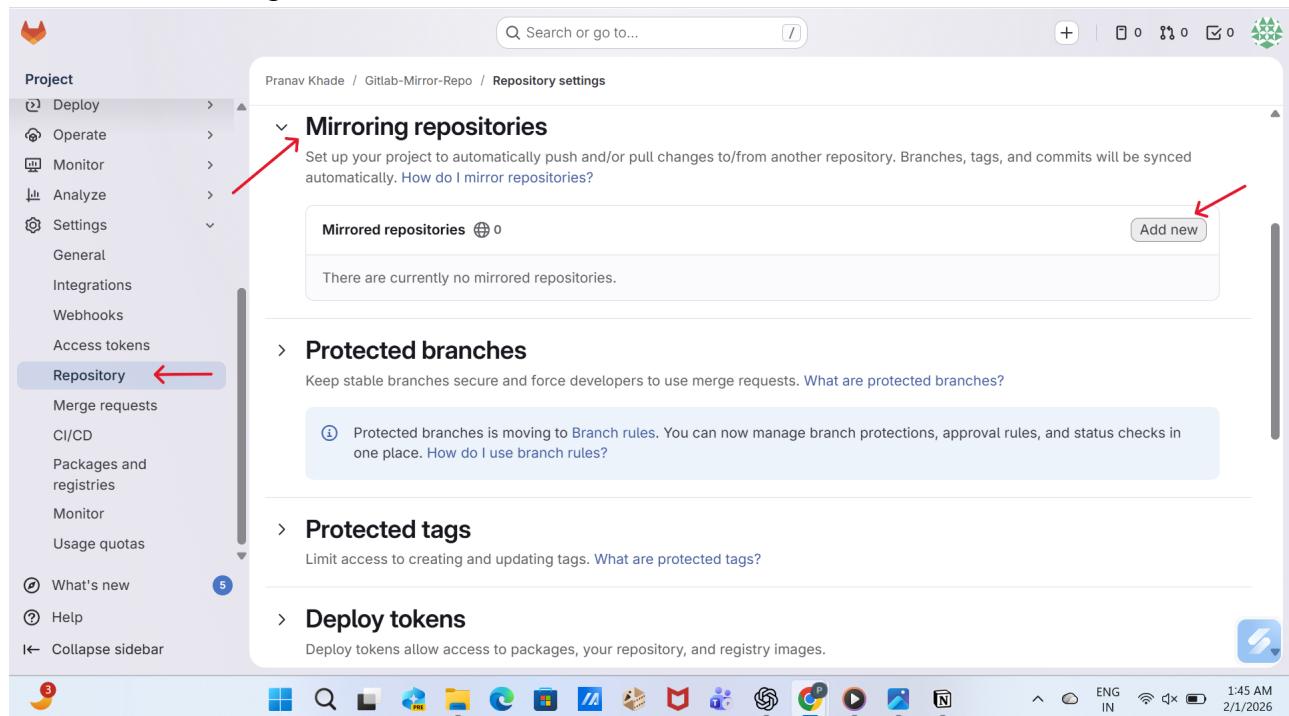
- 'Choose visibility' is set to 'Public' (indicated by a red arrow).
- 'Add README' is turned 'On' (indicated by a red arrow).
- 'Add .gitignore' is set to 'No .gitignore'.

At the bottom of the configuration section is a green 'Create repository' button, which also has a red arrow pointing to it.

Step 3 : Configure Repository Mirroring in GitLab (HTTPS URL)

1. Open your GitLab repository
2. Go to Settings → Repository
3. Scroll to Mirroring repositories
4. Select add new
5. Paste the GitHub repository HTTPS URL

6. Save the mirror configuration



The screenshot shows the 'Repository settings' page for a project named 'Gitlab-Mirror-Repo'. The left sidebar has a 'Repository' section highlighted with a red arrow. The main content area is titled 'Mirroring repositories' with a sub-section 'Protected branches'. A red arrow also points to the 'Add new' button in the 'Mirrored repositories' section.

Step 4 : Create GitHub Personal Access Token

1. Login to GitHub
2. Go to Settings → Developer settings
3. Open Personal access tokens
4. Select Generate new token (Classic)
5. Give a token name
6. Set token expiration (limit days as required)
7. Set token visibility as private
8. Select all required permissions (check all boxes)
9. Generate the token, copy it, and paste this token into the GitLab repository mirroring authentication section

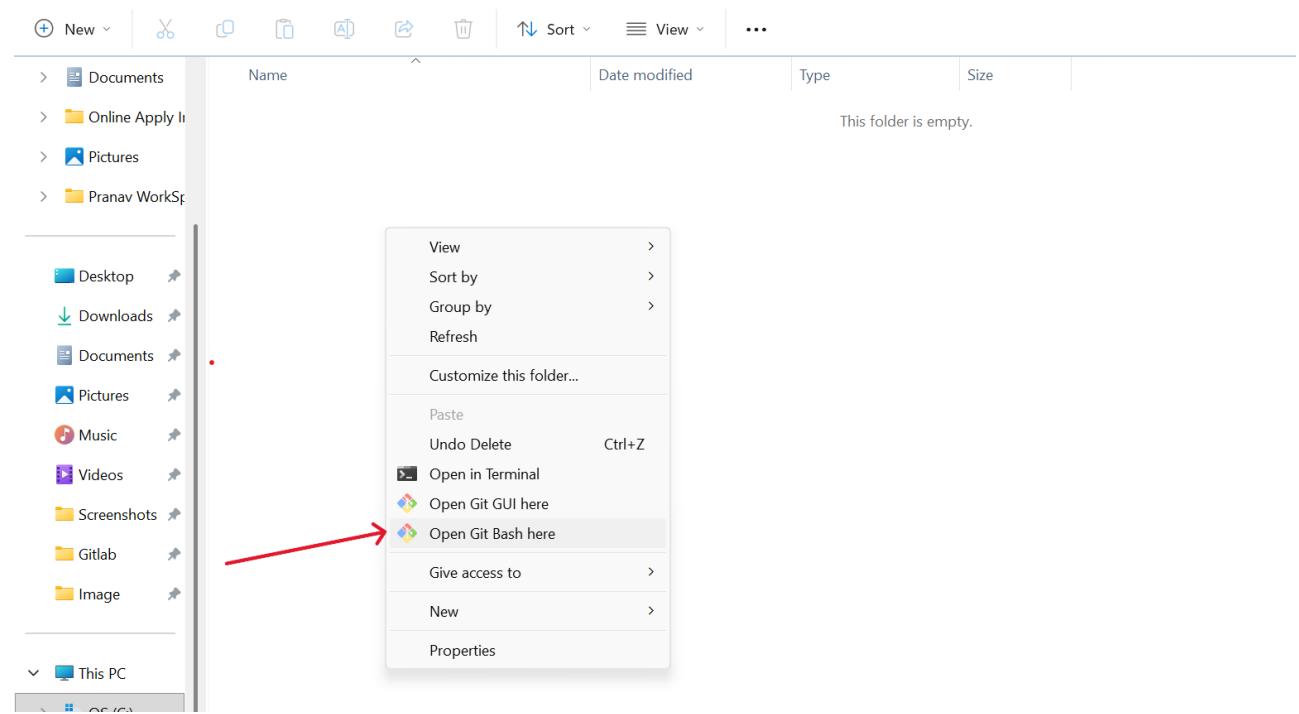
The screenshot shows the GitHub Developer Settings page. On the left, there's a sidebar with options like GitHub Apps, OAuth Apps, Personal access tokens (with a note about fine-grained tokens), and Tokens (classic). A blue arrow points from the sidebar to the 'Tokens (classic)' tab. In the main area, it says 'New personal access token (classic)'. Below that, a note explains what personal access tokens are and how they can be used. A 'Note' field contains 'Mirroring-Token'. Under 'Expiration', a date selector shows '02/06/2028' with a blue arrow pointing to the date input field. Under 'Select scopes', a table lists various GitHub permissions with checkboxes. Most checkboxes are checked, except for 'repo'. The table includes columns for scope name and description. A blue button with a gear icon is visible on the right.

Step 5 : Clone One Repository to Local Machine

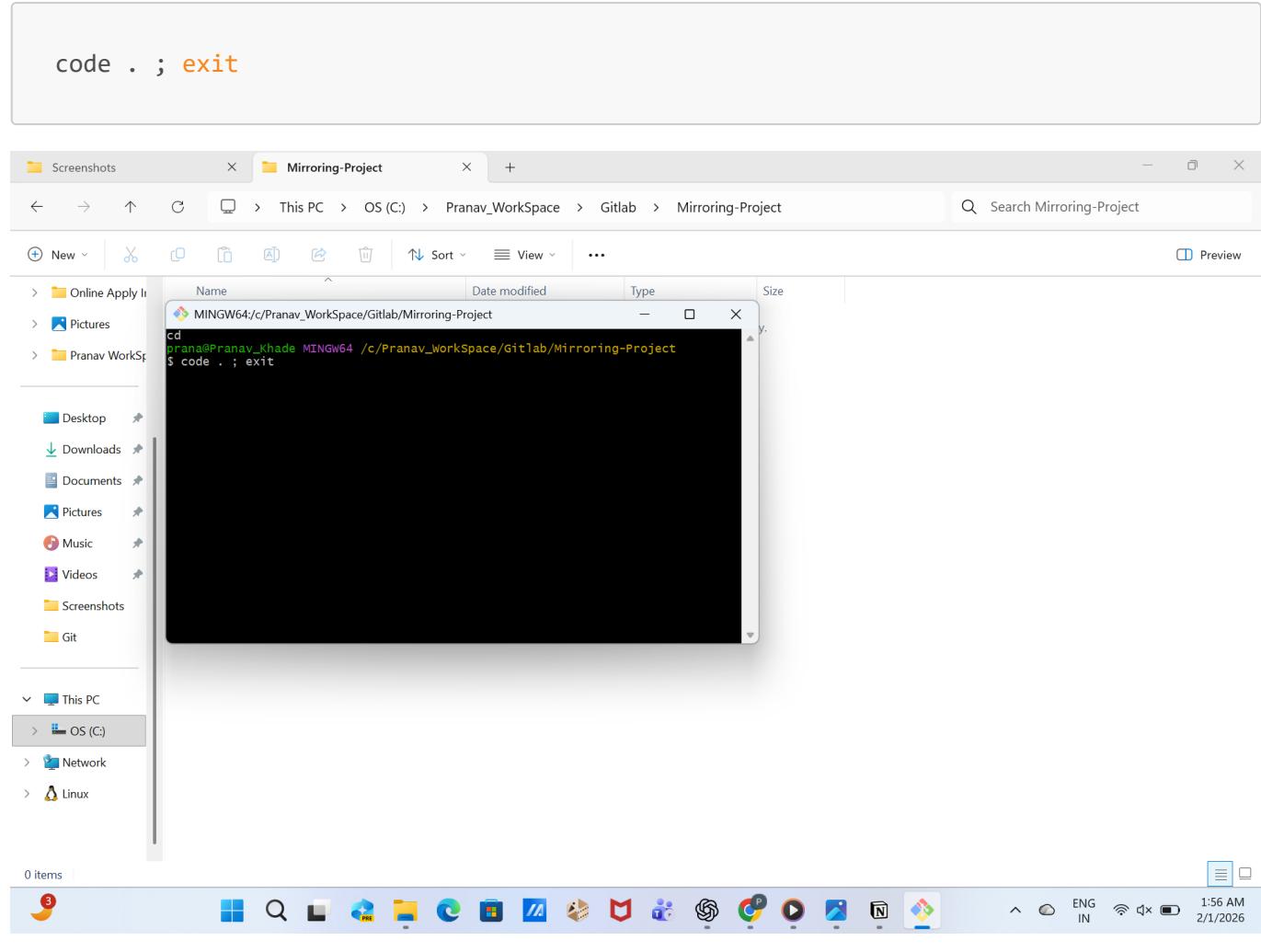
You can clone either GitHub or GitLab repository

```
git clone https://gitlab.com/username/repo-name.git
cd repo-name
```

- Right-click inside the folder, click Show more options, then select Open Git Bash here to open Git Bash in this directory



Step 6 : Open Project in VS Code



Step 7 : Create or Modify Files Locally

Examples:

- index.html
- README.md

Stage and Commit Changes

```
git status git add . git commit -m "Commit for mirroring to GitHub and GitLab"
```

Step 8 : Push File into Gitlab and Github

```
git push -u origin main
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ... bash - gitlab-mirror-repo + ⌂

```
prana@Pranav_Khade MINGW64 /c/Pranav_WorkSpace/Gitlab/Mirroring-Project
● $ cd gitlab-mirror-repo/

prana@Pranav_Khade MINGW64 /c/Pranav_WorkSpace/Gitlab/Mirroring-Project/git
lab-mirror-repo (main)
● $ ls -a
./ ../ .git/ README.md

prana@Pranav_Khade MINGW64 /c/Pranav_WorkSpace/Gitlab/Mirroring-Project/git
lab-mirror-repo (main)
○ $ [ ]
```

PROBLEMS OUTPUT TERMINAL ... bash - gitlab-mirror-repo + ⌂ ⌂ ... | [] X

```
prana@Pranav_Khade MINGW64 /c/Pranav_WorkSpace/Gitlab/Mirroring-Project/git
lab-mirror-repo (main)
$ git push -u origin main ←
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 312 bytes | 156.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://gitlab.com/pranav-khade-tech/gitlab-mirror-repo.git
  1830b0b..94ab241 main -> main
branch 'main' set up to track 'origin/main'.
```

\

Step 9 : See the Result File See Inside the GitLab and Github Repository

pranav-khade-tech / Github-Mirror-Repo Type / to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Github-Mirror-Repo Public Pin Watch 0

main 1 Branch 0 Tags Go to file Add file Code

pranav-khade-tech first commit 94ab241 · 2 minutes ago 2 Commits

File	Description	Time
README.md	Initial commit	27 minutes ago
index.html	first commit	2 minutes ago

The screenshot shows a GitLab repository interface. At the top, there's a navigation bar with a dropdown for 'main' and a search bar containing 'gitlab-mirror-repo'. To the right are buttons for '+', 'Find file', 'Code', and three vertical dots. Below the navigation is a commit history section. It shows a single commit by 'Pranav Khade' titled 'first commit' made 45 seconds ago. To the right of the commit are buttons for '94ab2419', a copy icon, and 'History'. Below the commit history is a table listing files with their last commit details:

Name	Last commit	Last update
README.md	Initial commit	26 minutes ago
index.html	first commit	46 seconds ago

Summary

This project shows how a single project maintained on a local machine can be kept in sync across GitLab and GitHub. All code updates are committed locally and pushed to GitLab, which then automatically mirrors the same changes to GitHub. This approach helps maintain identical code on multiple platforms, improves reliability through backup, and follows common Git and DevOps working practices used in real projects.