

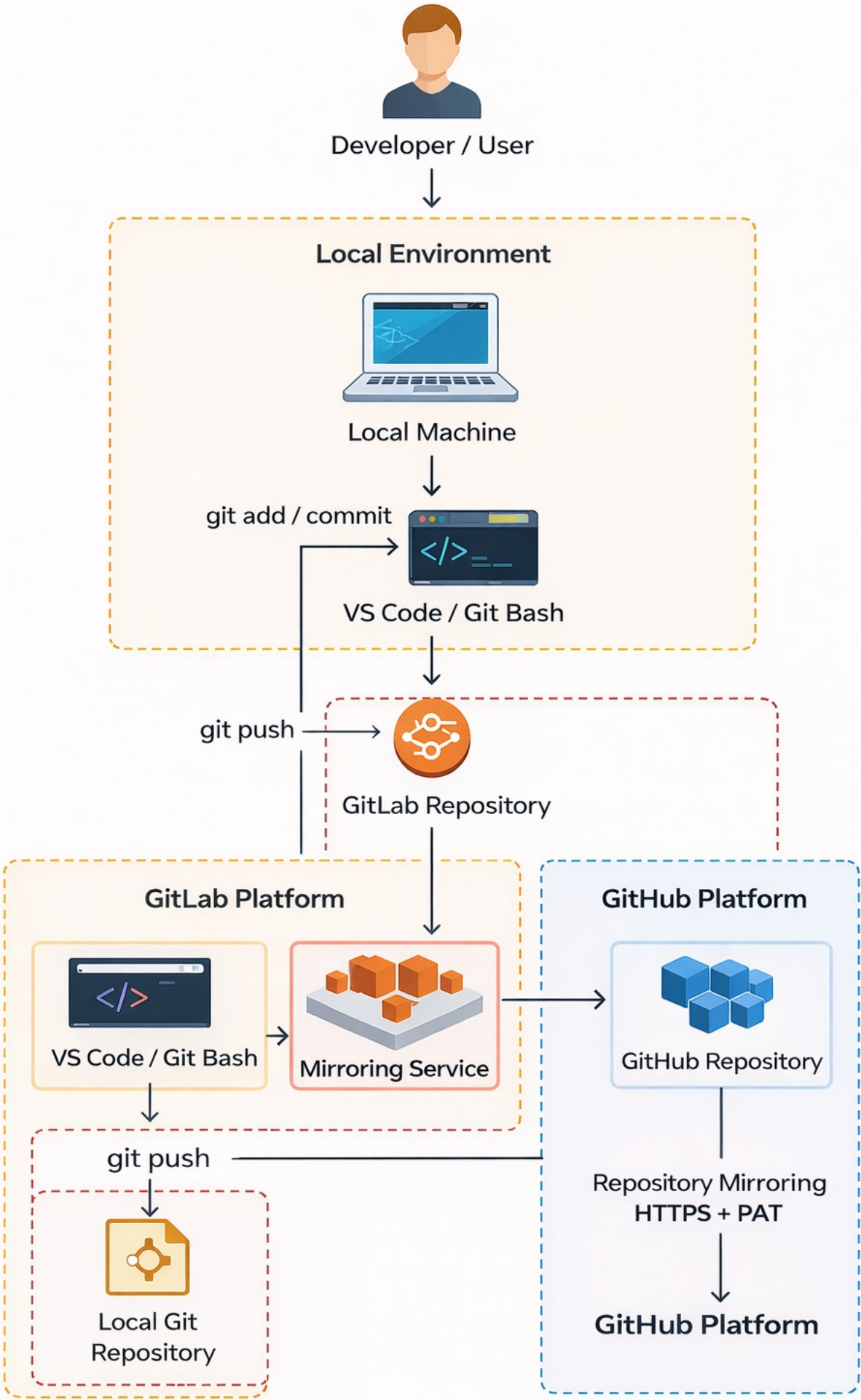
# Mirroring Of Repository From Local Machine --GitLab and GitHub

---

## 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 image shows two screenshots of the GitLab web interface for creating a new project.

**Top Screenshot: 'Create new project' page**

- Left sidebar: 'Your work' section with links to Home, Projects, Groups, Issues, Merge requests, To-Do List, Milestones, Snippets, Activity, Import history, Workspaces, Environments, Operations, Security, What's new, and Help.
- Main content area: 'Create new project' header with two options: 'Create blank project' (with a plus icon) and 'Create from template' (with a plus icon).


**Bottom Screenshot: 'Create blank project' form**

- Left sidebar: Same as the top screenshot, but with a red arrow pointing to the 'Merge requests' link.
- Main content area: 'Create blank project' form with the following fields:
  - Project name:** A text input field containing 'Gitlab-Mirror-Repo'. A red arrow points to this field.
  - Project URL:** A dropdown menu showing 'https://gitlab.com/' and 'pranav-khade-tech'. A red arrow points to this dropdown.
  - Project slug:** A text input field containing 'gitlab-mirror-repo'. A red arrow points to this field.
  - Project deployment target (optional):** A dropdown menu with 'Select the deployment target'.
  - Visibility Level:** Three radio buttons: 'Private' (selected), 'Internal', and 'Public'. A red arrow points to the 'Public' radio button.

## Step 2 : Create Repository on GitHub

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

1 General

Owner \*  / Repository name \*  ←


✓ Github-Mirror-Repo is available.

Great repository names are short and memorable. How about [cautious-octo-train](#)?


Description

0 / 350 characters


2 Configuration

Choose visibility \*  Public


Choose who can see and commit to this repository

Add README  ←

READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore 

.gitignore tells git which files not to track. [About ignoring files](#)

Add license 

Licenses explain how others can use your code. [About licenses](#)

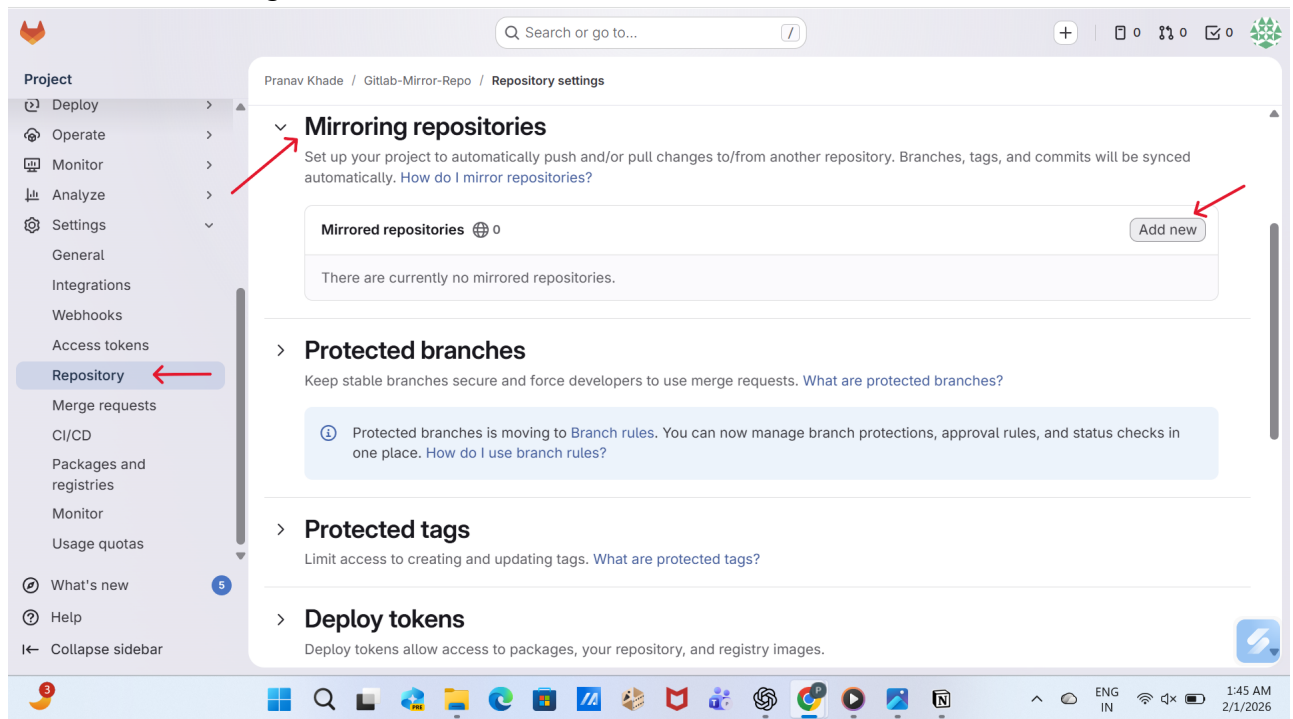
Create repository ←

---

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

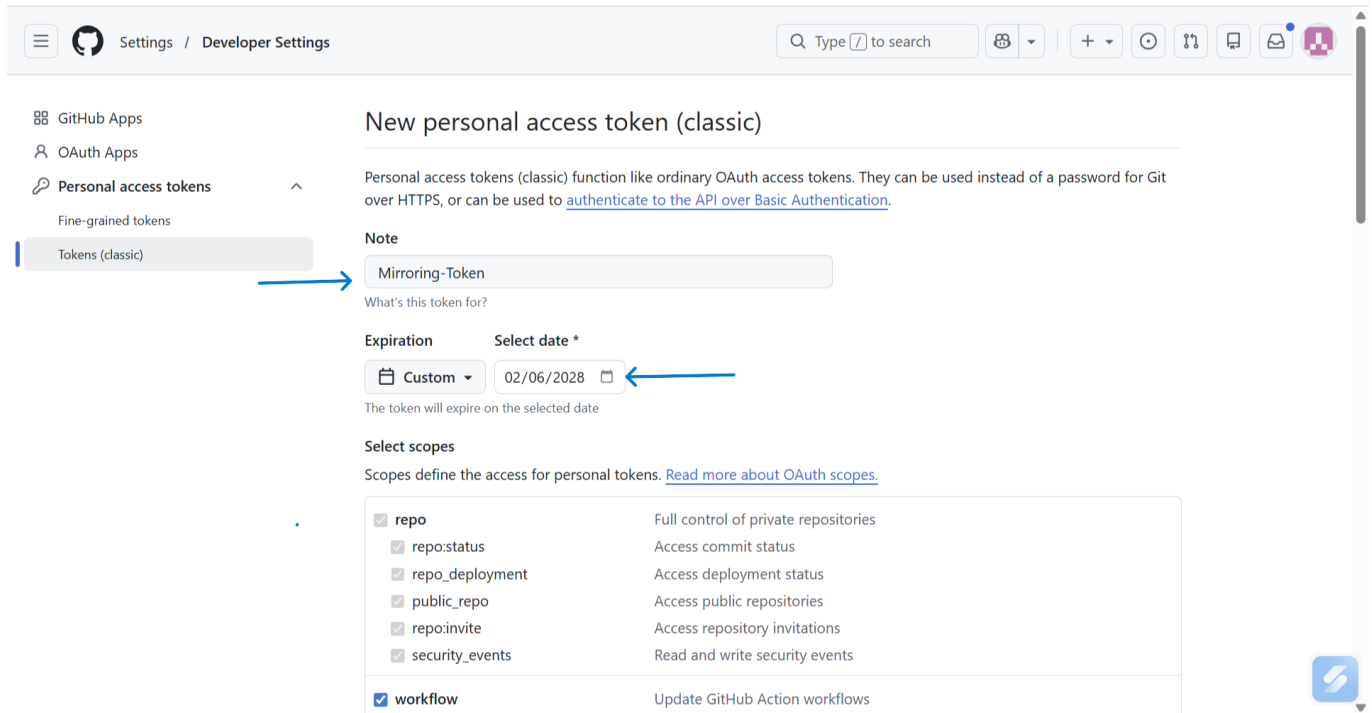
1. Open your GitLab repository
2. Go to Settings → Repository
3. Scroll to Mirroring repositories
4. Select add new
5. Paste the GitHub repository HTTPS URL

## 6. Save the mirror configuration



## 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

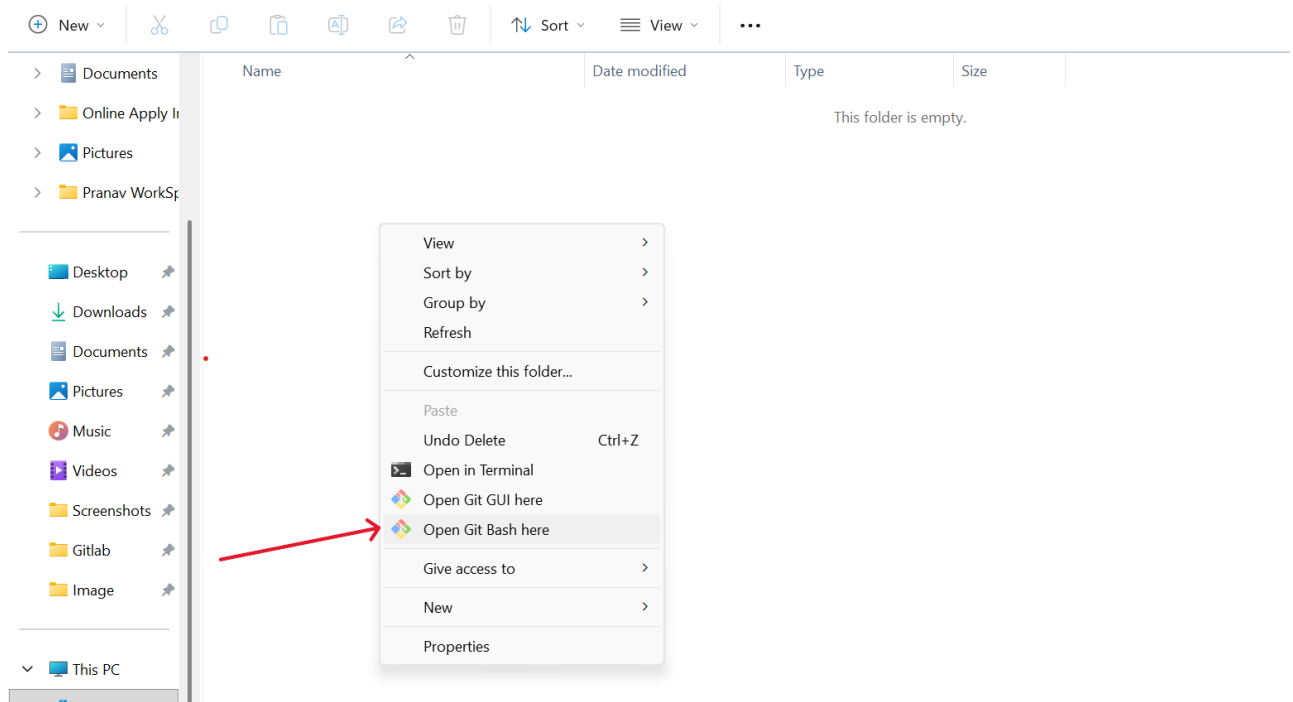


## Step 5 : Clone One Repository to Local Machine

You can clone either GitHub or GitLab repository

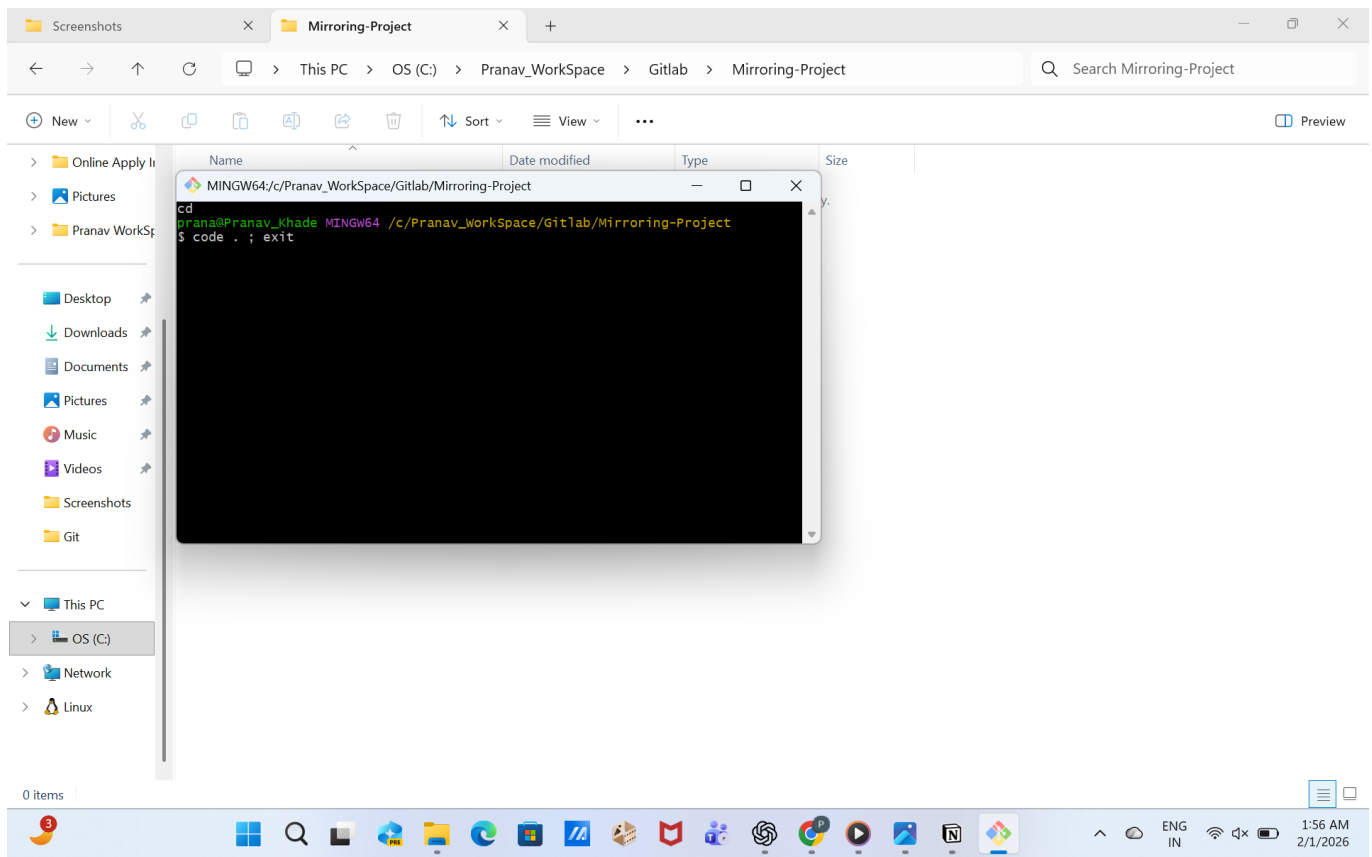
```
git clone https://gitlab.com/username/repo-name.git
cd repo-name
```

- 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

```
code . ; exit
```



## 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
```

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

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

prana@Pranav_Khade MINGW64 /c/Pranav_WorkSpace/Gitlab/Mirroring-Project/gitlab-mirror-repo (main)
○ $
```

```
prana@Pranav_Khade MINGW64 /c/Pranav_WorkSpace/Gitlab/Mirroring-Project/gitlab-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'.
```

/

pranav-khade-tech / Github-Mirror-Repo

Q 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

t

Add file

<> Code

pranav-khade-tech

first commit

94ab241 · 2 minutes ago

2 Commits

README.md

Initial commit

27 minutes ago

index.html

first commit

2 minutes ago



main

gitlab-mirror-repo


+

▼


Find file

Code



⋮

 **first commit**  
Pranav Khade authored 45 seconds ago

94ab2419



History

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.