Git and GitHub: A Comprehensive Overview

Git is a powerful version control system, while GitHub platform for hosting and collaborating on Git repositories.

by Thamarai Manalan

git

created by
Linus Torvalds in 2005

Initial configuration

Why Use Git and GitHub?

1
Version Control

Track changes to your code, revert to previous versions, and collaborate effectively.

2
Collaboration

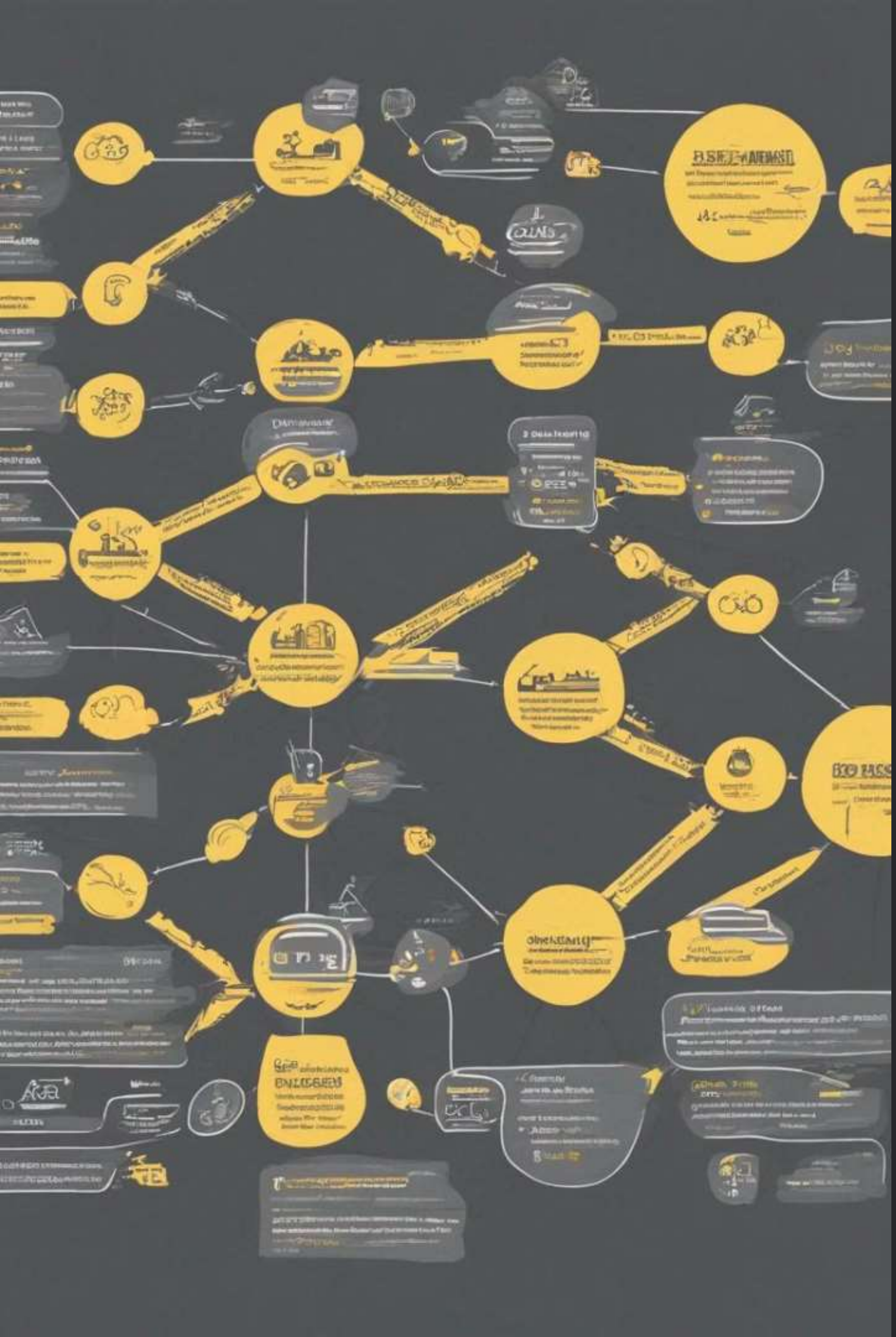
Work on projects with others, share code, and review changes efficiently.

3
Open Source

Contribute to and learn from a vast community of developers.

4
Backup and Security

Securely store your code in a cloud-based repository for easy access and disaster recovery.



Git Basics: Repositories, Branches, Commits

1 Repositories

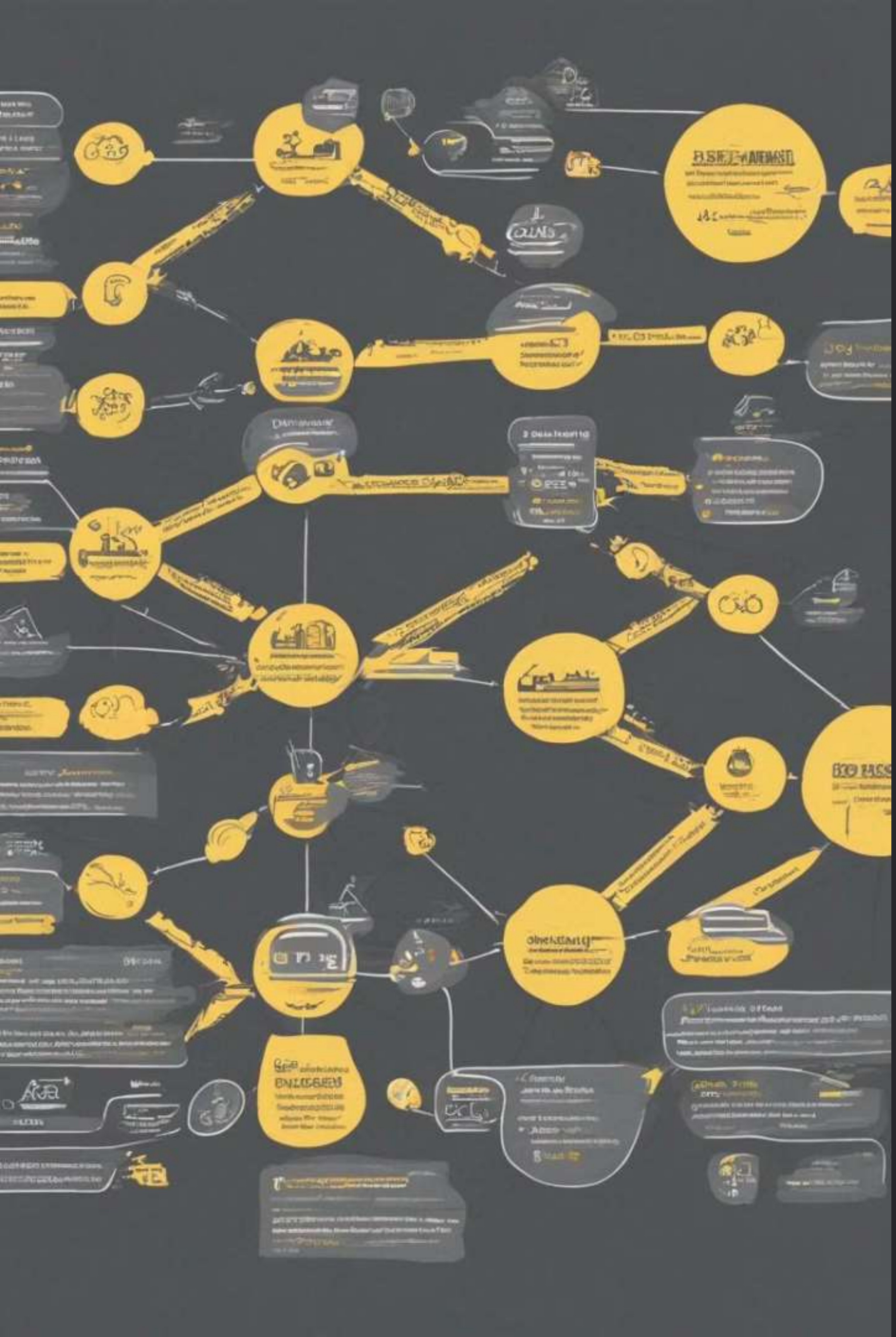
A central location for storing your code and its history.

2 Branches

Create isolated copies of your code to work on new features or bug fixes.

3 Commits

Snapshots of your code at specific points in time, capturing changes and progress.



Git Basics Commands

1 git init

Initialize a new Git repository

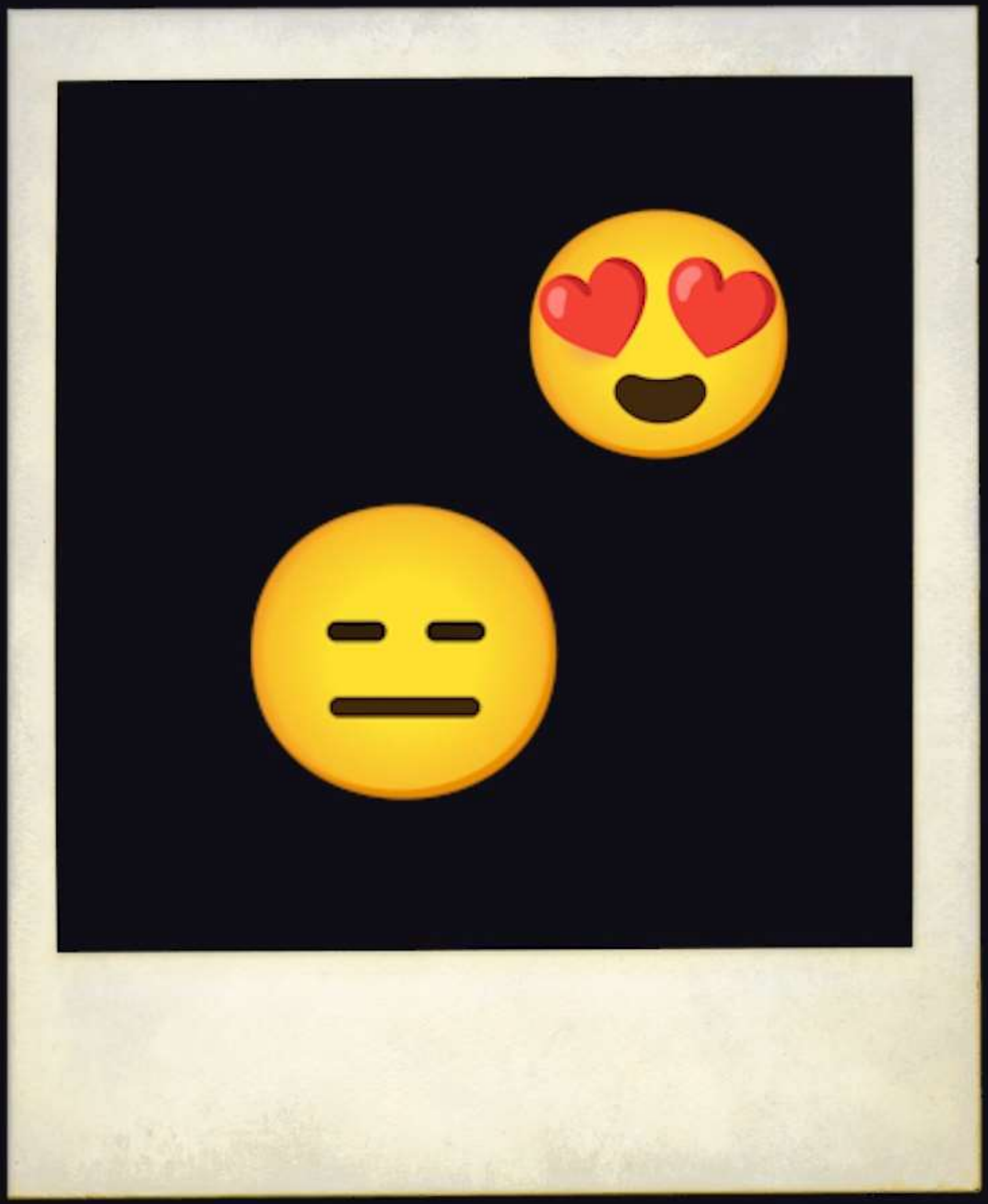
2 git status

Check the status of the working directory

3 git clone <repository-url>

Clone a remote repository

Understanding Staging



VISUAL STUDIO CODE GUI FEATURES

GITHUB

Tip : press . To open cloud based vs code

Collaborating with Git: Cloning, Pushing, Pulling

1

Cloning

Create a local copy of a remote repository to start working on a project.

2

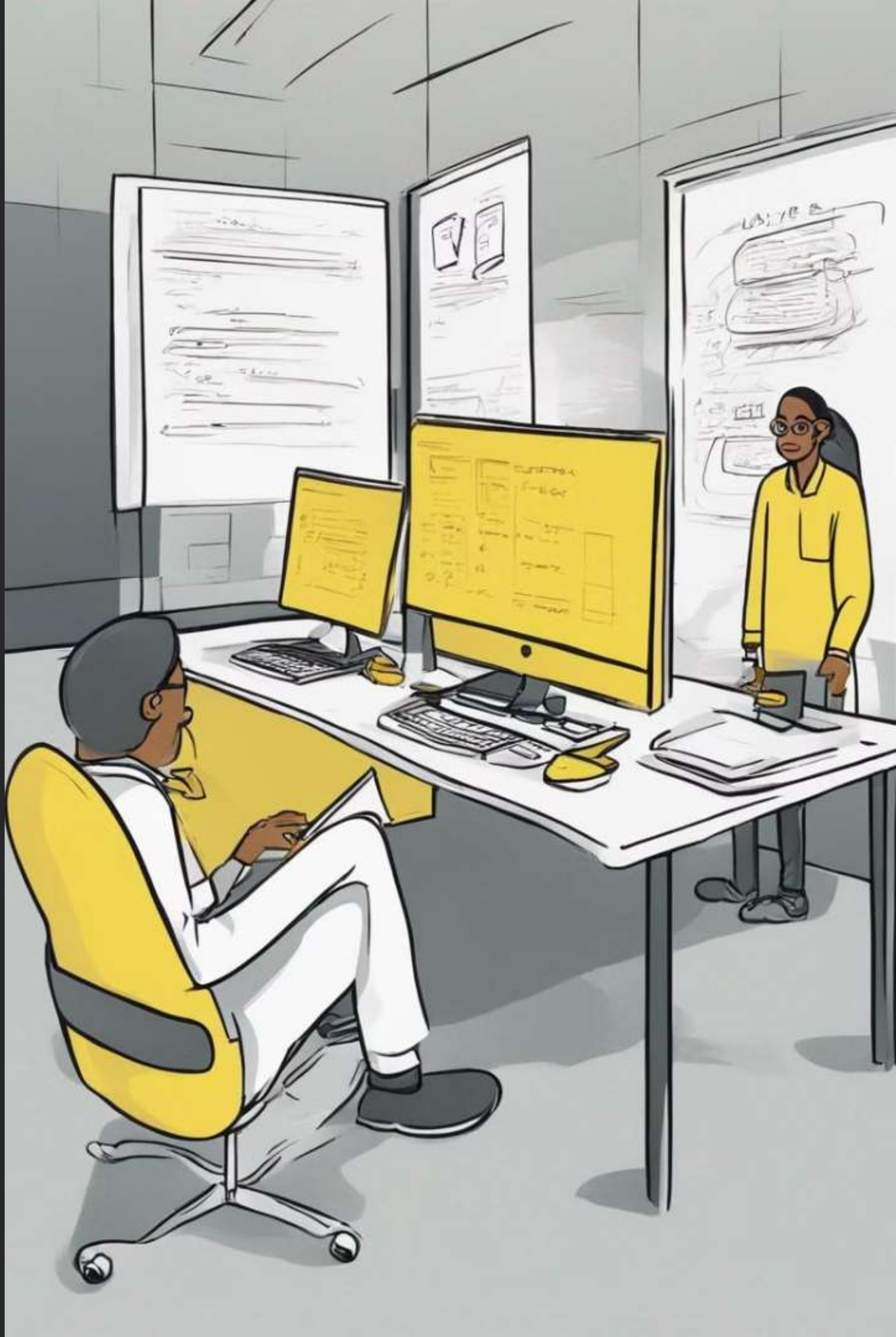
Pushing

Upload your changes from your local repository to the remote repository.

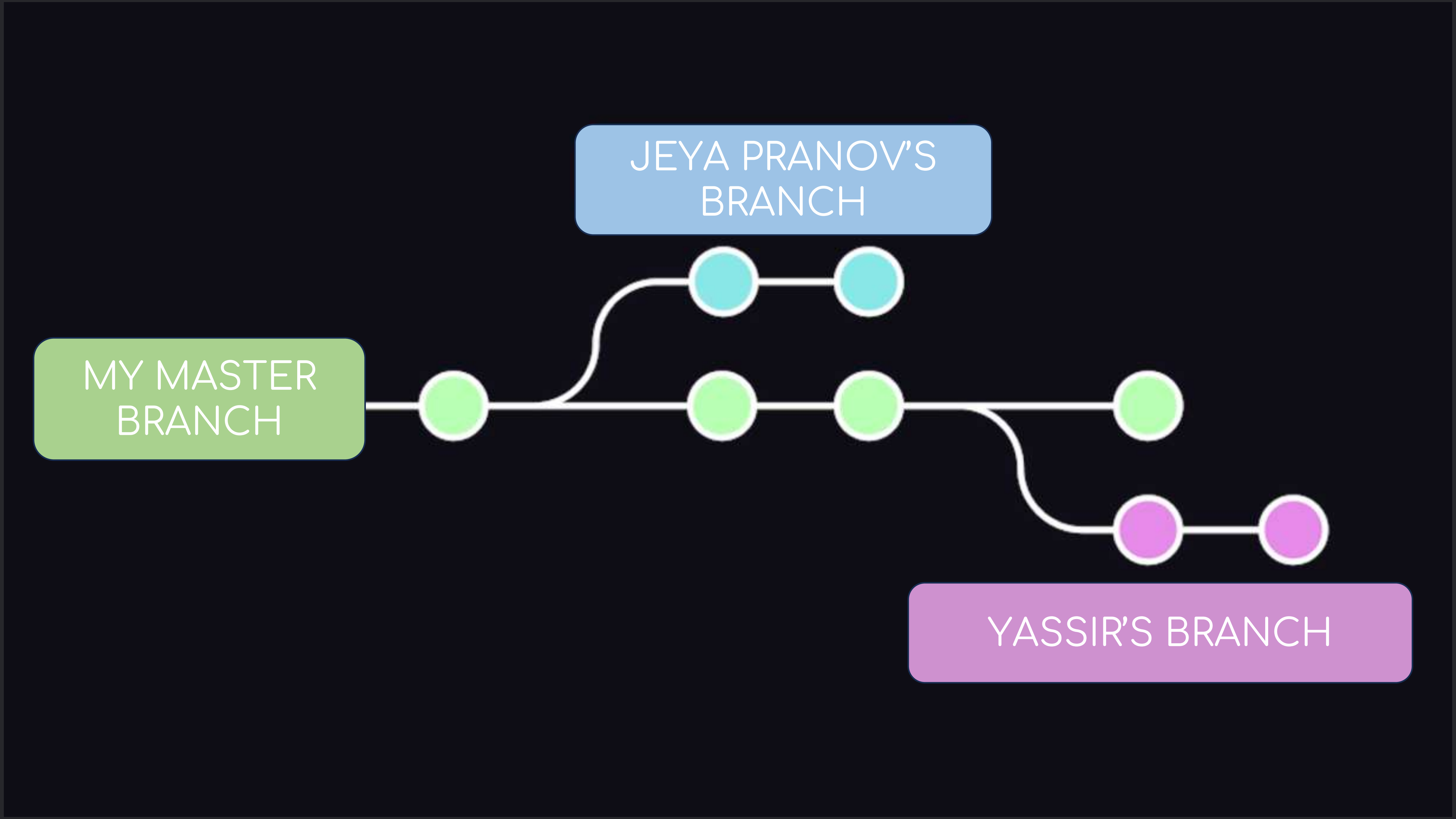
3

Pulling

Download changes from the remote repository to your local repository.



BRANCH



JEYA PRANOV'S BRANCH

MY MASTER BRANCH

YASSIR'S BRANCH

Git Branching Workflows

Gitflow

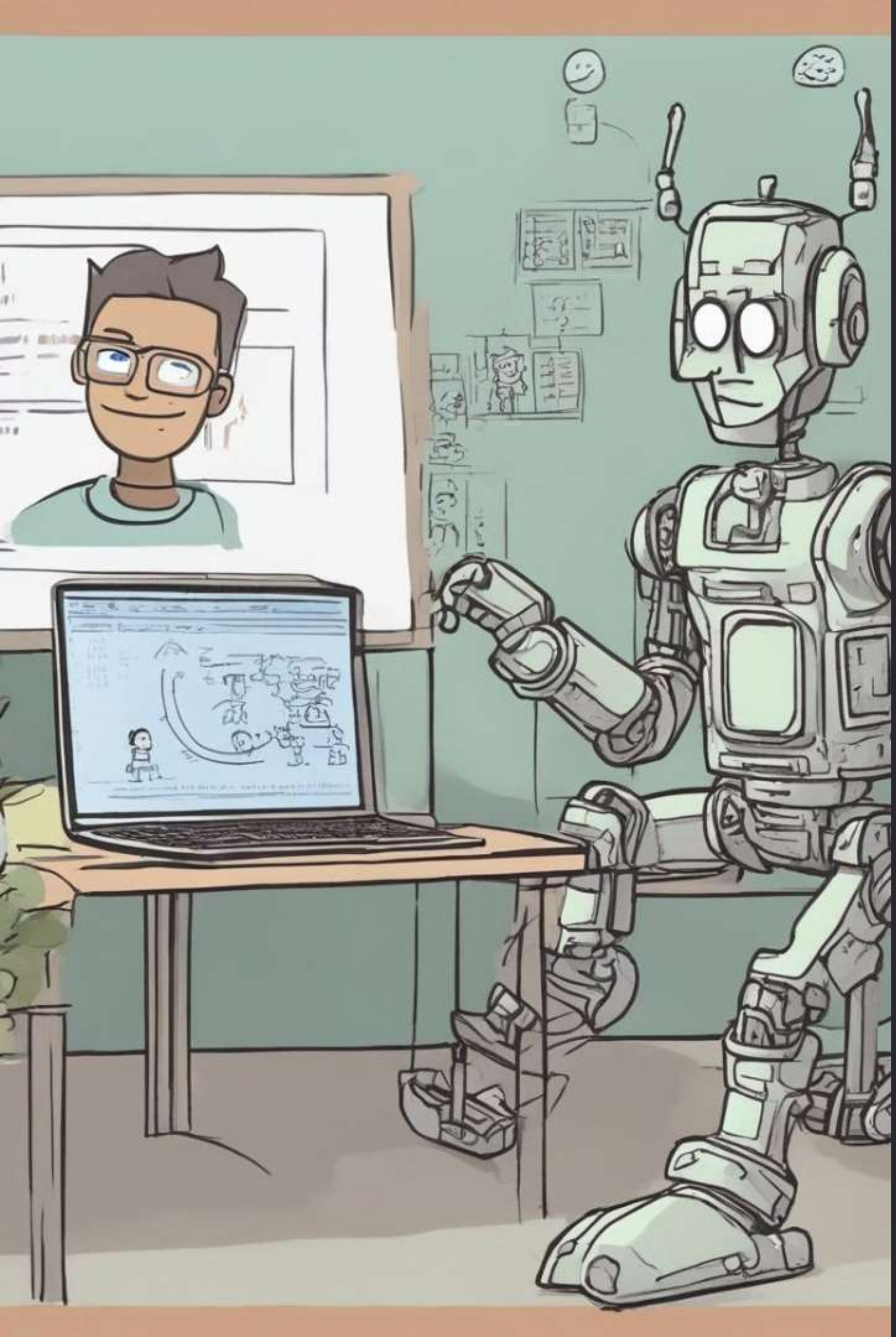
A structured workflow with dedicated branches for development, releases, and hotfixes.

Feature Branches

Create isolated branches for each new feature, allowing parallel development and easy merging.

Hotfix Branches

Quickly fix bugs in production without disrupting ongoing development.



Resolving Merge Conflicts

Identify Conflicts

Git will alert you to conflicts when merging branches.

Review Changes

Manually review conflicting lines and choose the correct version.

Resolve and Commit

Resolve conflicts, commit the changes, and continue merging.

FORK AND PULL REQUEST

Best Practices for Working with Git and GitHub

1 **Write Meaningful Commit Messages**

Clear and concise commit messages help you understand the changes made.

2 **Use Small, Focused Commits**

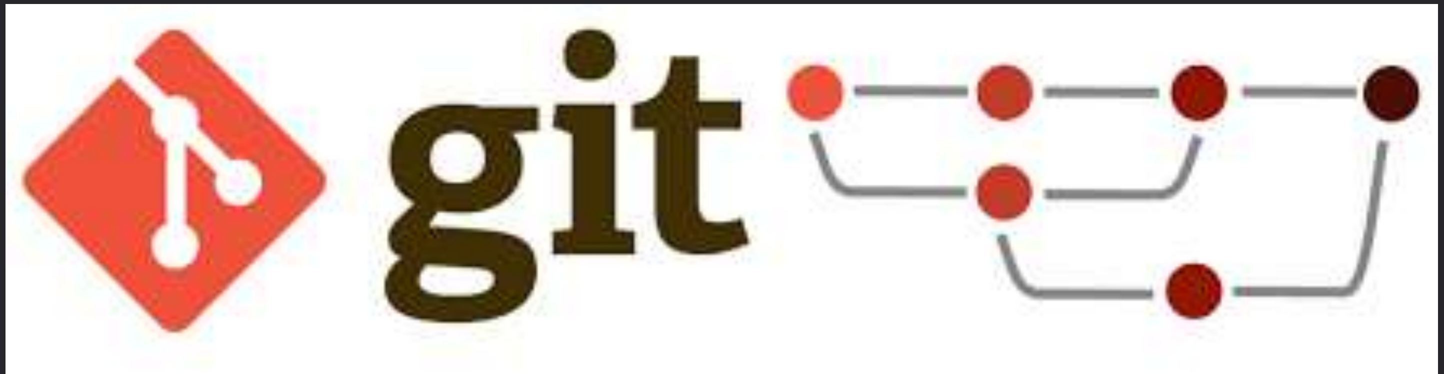
Commit changes in small increments for easier review and troubleshooting.

3 **Regularly Push Your Changes**

Push changes to the remote repository frequently to avoid conflicts.

4 **Collaborate Effectively**

Communicate with your team, review code, and give constructive feedback.



THANK YOU !!