SEGi COLLEGE
KUALA LUMPUR

GIT for Coding

A Practical Workshop

Lesson Learning Outcomes

01

To provide students with fundamentals knowledge of GIT platform

02

To provide students with knowledge on how to use GIT for code management

03

To provide students with adequate knowledge on ways to publish and share their source codes

Workshop Agenda/Content

Git & GitHub

GitHub Account

Installing GitHub Desktop

Managing Repository

Add & Commit

Branch & Merge

Push & Pull

Fork & Contribute

Git & GitHub

Challenges in Coding

Apart from regular issues of coding such as syntax & semantics error, algorithm errors and unreadable codes, these are common issues of coding.

- Hard to track changes
- Require multiple trials/testing
- Accidentally damaging the code
- Multiple copies created
- Team collaboration

Introduction to Version Control

Version control is a critical practice in software development that involves tracking and managing changes to code and other documents over time.

This process enables developers to maintain a comprehensive history of modifications, facilitating collaboration and enhancing the overall efficiency of the development workflow.

Centralized Version Control Systems (CVCS)

Store all versions of files in a single central repository. Developers check out files for editing and check them back in after making changes.

Distributed Version Control Systems (DVCS)

Every developer has a complete copy of the repository on their local machine. Changes are synchronized between local copies and the central repository, allowing for greater flexibility and offline work capabilities.

Benefits of Version Control

Change Tracking

Version control systems maintain a detailed history of changes made to the codebase, including who made each change and when.

Collaboration

Multiple developers can work on different features simultaneously without overwriting each other's changes.

Error Recovery

If a mistake is made or an undesirable change is introduced, version control allows developers to revert back to previous versions of the code easily.

Branching and Merging

Developers can create branches to work on projects independently and merge once the work is completed.

Documentation

The commit history serves as documentation for the project, providing insights into why certain changes were made.

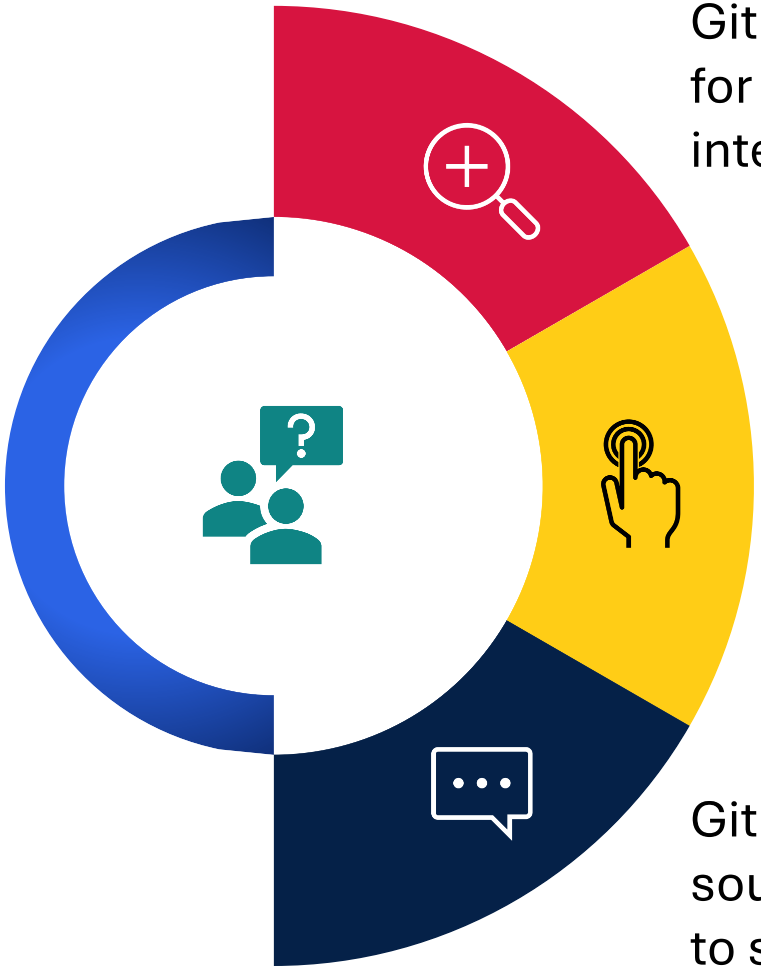
Introduction to Git

Git is a powerful version control system (VCS) that has become essential in modern software development.

Allows developers to track changes in their codebase, collaborate with others, and maintain a history of project modifications.

Created by Linus Torvalds in 2005 and is designed to handle everything from small to very large projects with speed and efficiency.

Introduction to GitHub



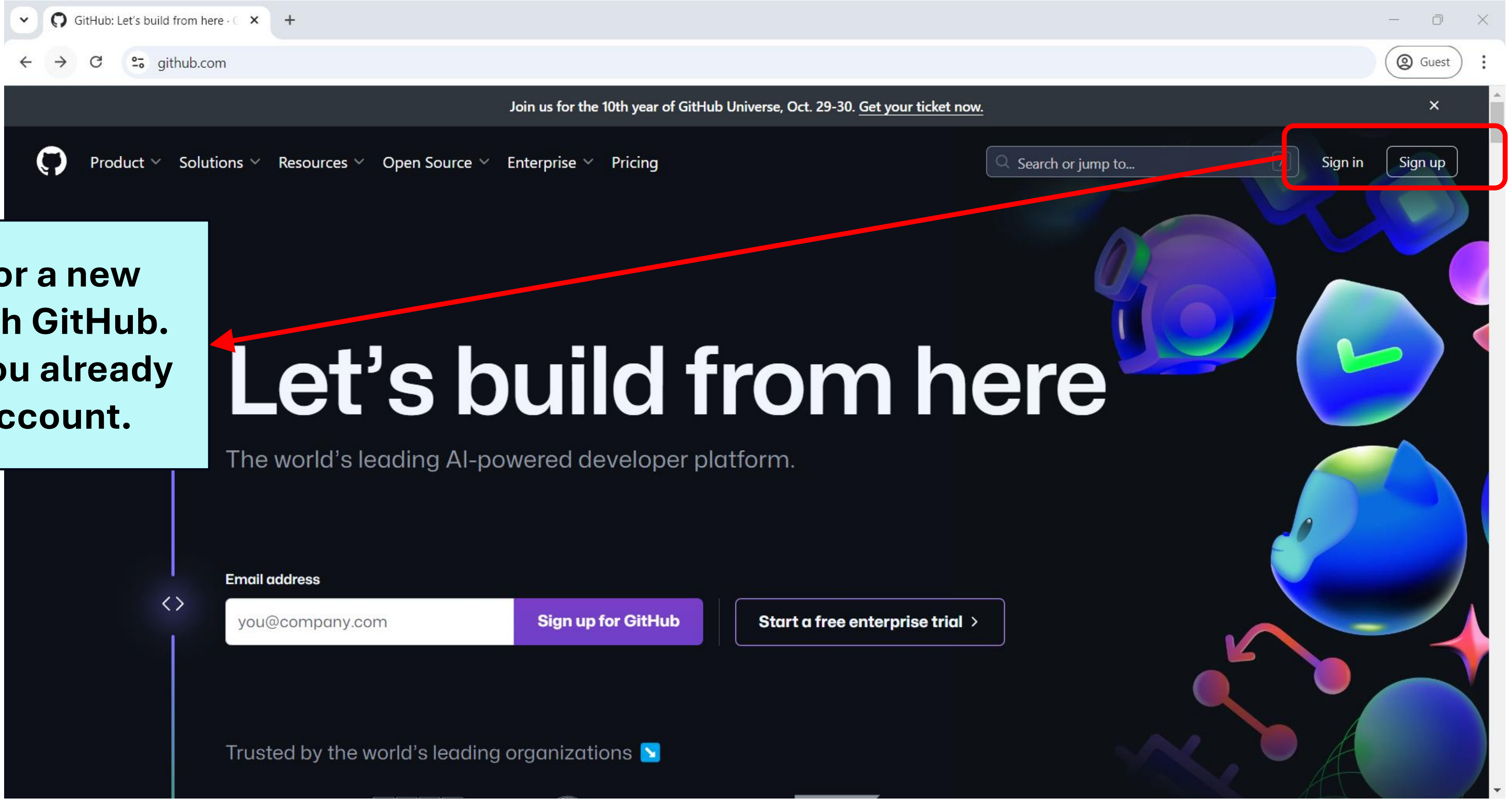
GitHub is a cloud-based hosting service for Git repositories that provides a web interface for managing Git repositories.

It offers features like pull requests, issues tracking, and project boards that facilitate teamwork and project management.

GitHub excels in supporting open-source projects, allowing developers to share their work and contribute to others' projects easily.

GitHub Account

Sign Up for GitHub



GitHub: Let's build from here

Join us for the 10th year of GitHub Universe, Oct. 29-30. Get your ticket now.

Let's build from here

The world's leading AI-powered developer platform.

Email address

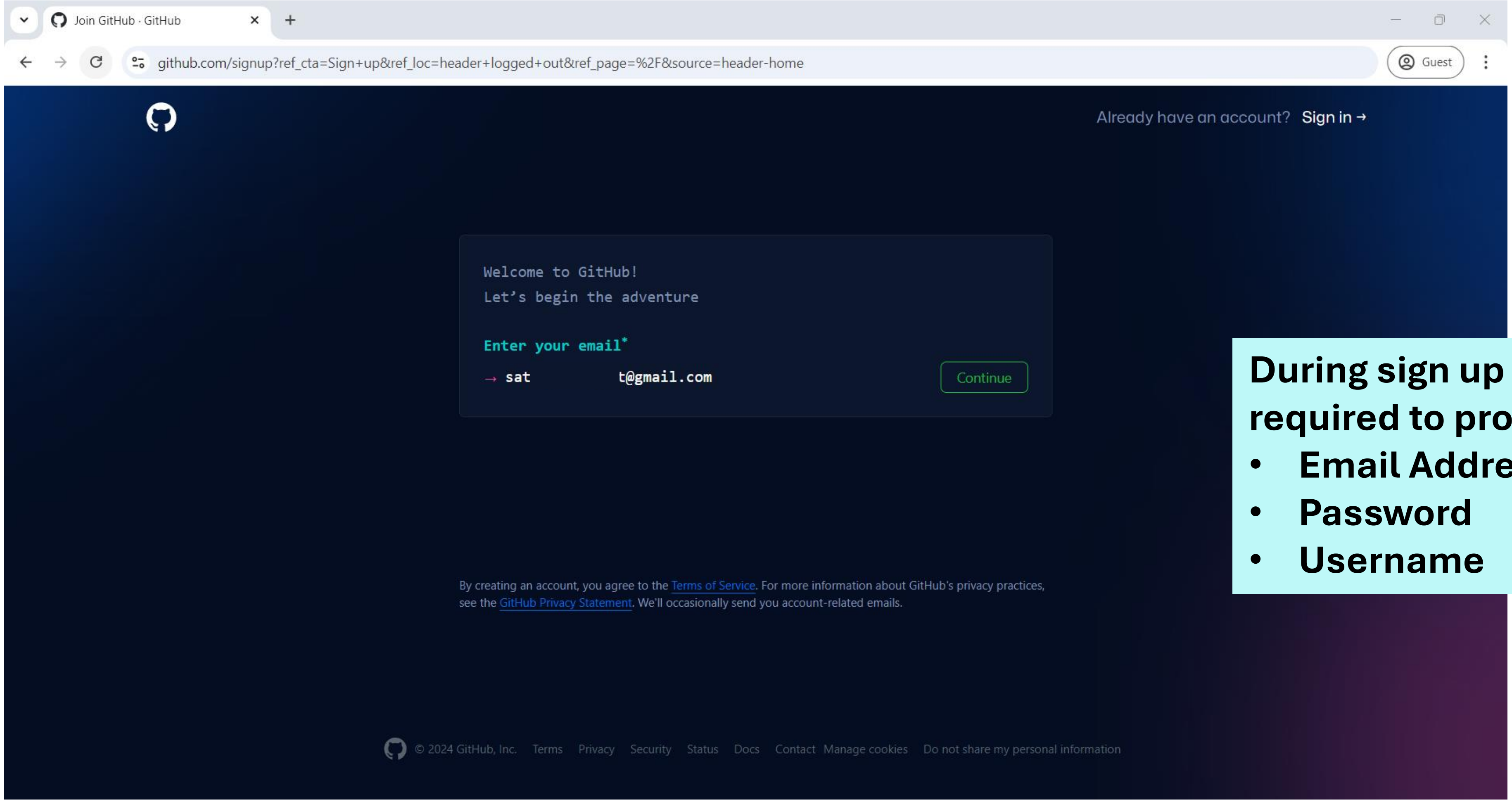
Start a free enterprise trial >

Trusted by the world's leading organizations

SIGN UP for a new account with GitHub. SIGN IN if you already have an account.

www.github.com

Sign Up for GitHub



github.com/signup?ref_cta=Sign+up&ref_loc=header+logged+out&ref_page=%2F&source=header-home

```
Welcome to GitHub!
Let’s begin the adventure

Enter your email*
→ sat[redacted]t@gmail.com
```

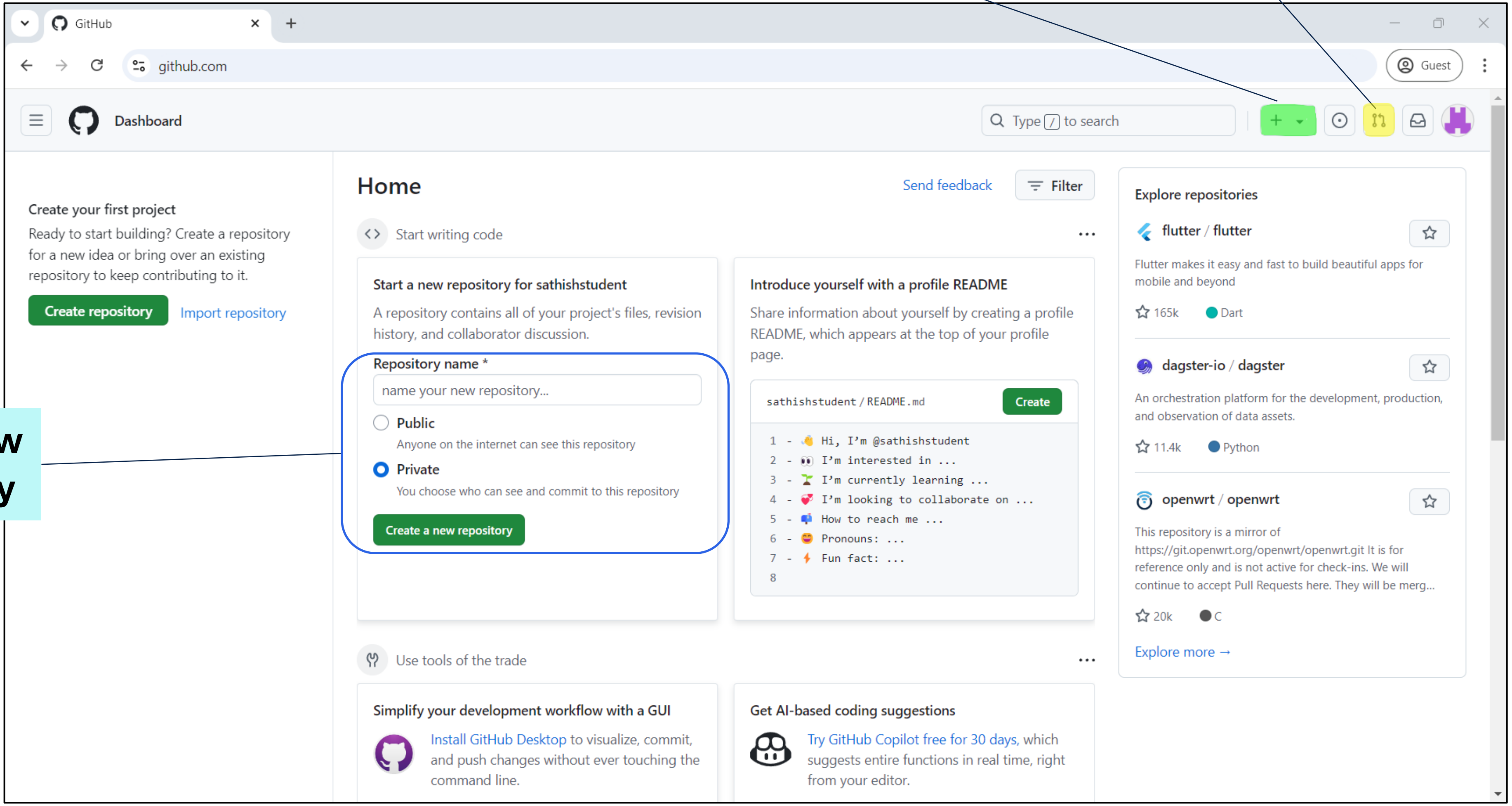
By creating an account, you agree to the Terms of Service. For more information about GitHub’s privacy practices, see the GitHub Privacy Statement. We'll occasionally send you account-related emails.

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

During sign up you are required to provide

- Email Address
- Password
- Username

GitHub Dashboard



Create New

Pull Requests

Create New Repository

Home

Send feedback · Filter

Create your first project

Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.

Create repository · Import repository

Start writing code

Start a new repository for sathishstudent

A repository contains all of your project's files, revision history, and collaborator discussion.

Repository name *

name your new repository...

○ Public
Anyone on the internet can see this repository

● Private
You choose who can see and commit to this repository

Create a new repository

Introduce yourself with a profile README

Share information about yourself by creating a profile README, which appears at the top of your profile page.

sathishstudent / README.md · Create

```
- 👋 Hi, I'm @sathishstudent
- 👀 I'm interested in ...
- 🌱 I'm currently learning ...
- 💞️ I'm looking to collaborate on ...
- 📫 How to reach me ...
- 😄 Pronouns: ...
- ⚡ Fun fact: ...
```

Use tools of the trade

Simplify your development workflow with a GUI

Install GitHub Desktop to visualize, commit, and push changes without ever touching the command line.

Get AI-based coding suggestions

Try GitHub Copilot free for 30 days, which suggests entire functions in real time, right from your editor.

Explore repositories

flutter / flutter
Flutter makes it easy and fast to build beautiful apps for mobile and beyond
☆ 165k · Dart

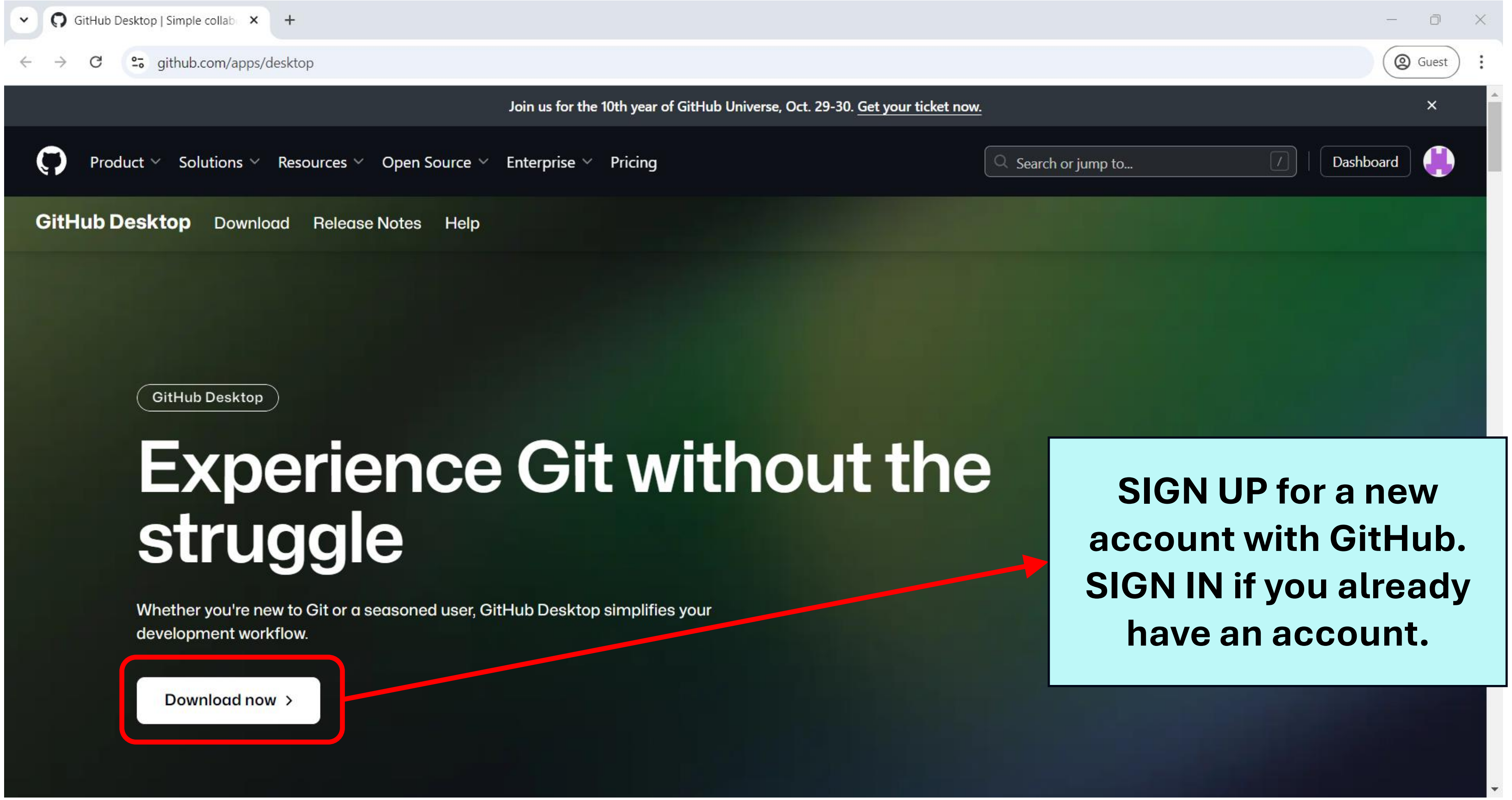
dagster-io / dagster
An orchestration platform for the development, production, and observation of data assets.
☆ 11.4k · Python

openwrt / openwrt
This repository is a mirror of https://git.openwrt.org/openwrt/openwrt.git It is for reference only and is not active for check-ins. We will continue to accept Pull Requests here. They will be merg...
☆ 20k · C

Explore more →

GitHub Desktop

GitHub Desktop



Join us for the 10th year of GitHub Universe, Oct. 29-30. Get your ticket now.

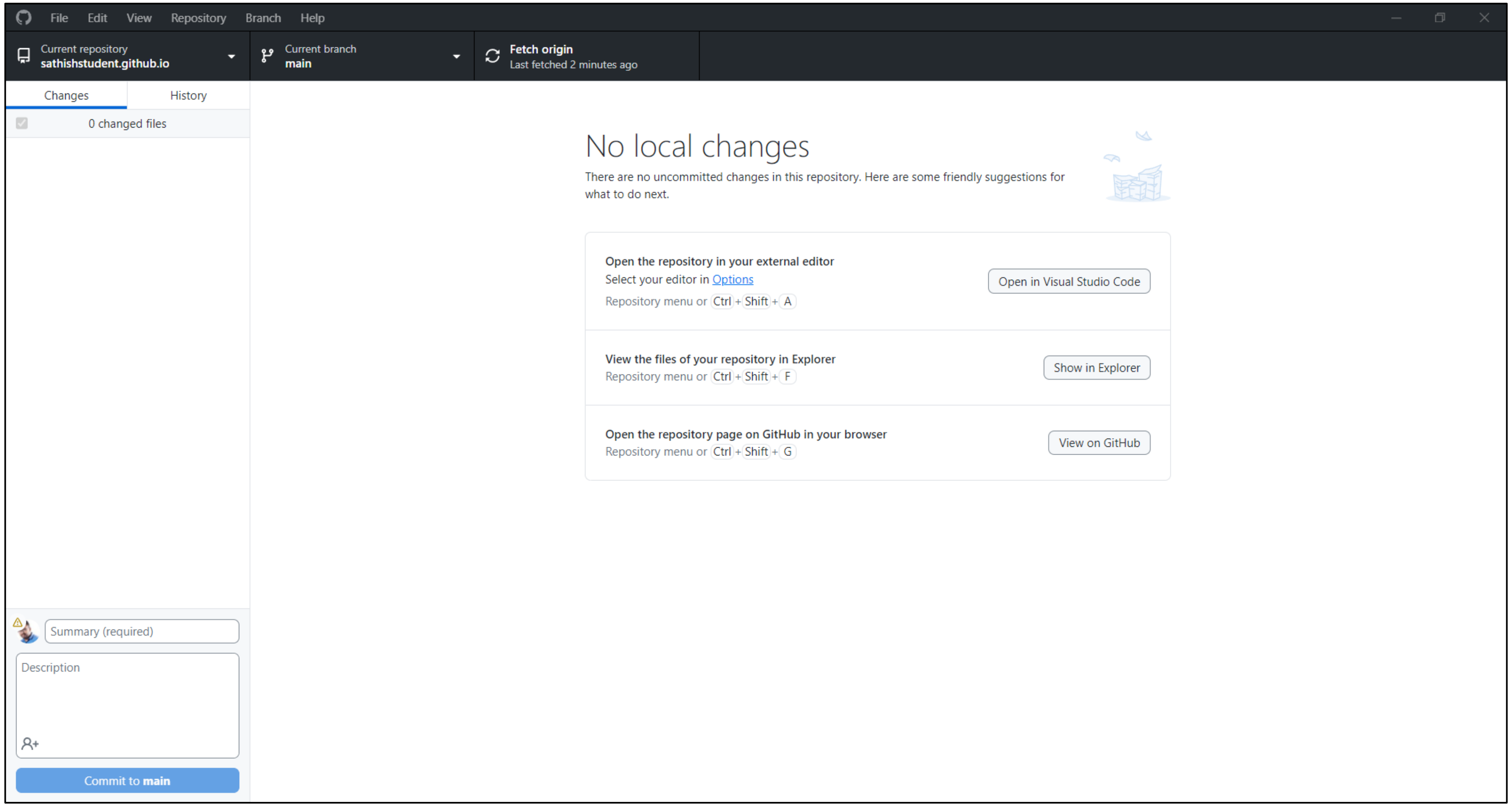
Experience Git without the struggle

Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow.

SIGN UP for a new account with GitHub. SIGN IN if you already have an account.

https://desktop.github.com

GitHub Desktop



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Open the repository in your external editor
Select your editor in Options
Repository menu or Ctrl + Shift + A

View the files of your repository in Explorer
Repository menu or Ctrl + Shift + F

Open the repository page on GitHub in your browser
Repository menu or Ctrl + Shift + G

GitHub Desktop for Local Repository

Concepts of Git

Aspects of Git : Repository

A repository on GitHub is a central location where all the files for a project are stored, along with the complete history of changes made to those files.

Serves as a workspace for developers to collaborate on code, track revisions, and manage project files.

Code Files

The actual source code and other related files that are uploaded into the repository for storing.

Commit History

A detailed log of all changes made, allowing developers to revert to previous versions if needed.

Branches

Separate lines of development that allow multiple features or fixes to be worked on simultaneously without interference.

Aspects of Git : Project

A project in GitHub is a feature that helps manage tasks and workflows within a repository.

It provides a visual interface for organizing issues, pull requests, and notes into columns, similar to a Kanban board.

Task Management

Projects help teams track progress on features or bugs by moving cards representing tasks through various stages (e.g., To Do, In Progress, Done).

Collaboration

Team members can comment on tasks, assign them to different contributors, and set due dates.

Integration

Projects can be linked with issues and pull requests, providing a comprehensive view of ongoing work related to a repository.

Aspects of Git : Add, Commit and Rebase

Add

The git add command is used to add changes (modified or newly created files) to the staging area. Files in the staging area are not yet committed to the repository, but they are marked to be included in the next commit.

Commit

A commit in Git represents a snapshot of the repository at a particular point in time. Each commit has a unique ID (hash) and stores the changes made to the files. It also includes a commit message that describes the purpose of the commit.

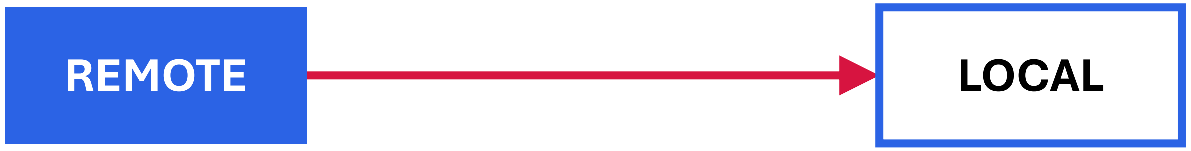
Rebase

Rebasing is the process of moving or combining a sequence of commits to a new base commit. It is often used to keep a linear project history by integrating changes from one branch into another without creating merge commits.

Aspects of Git : Pull & Push

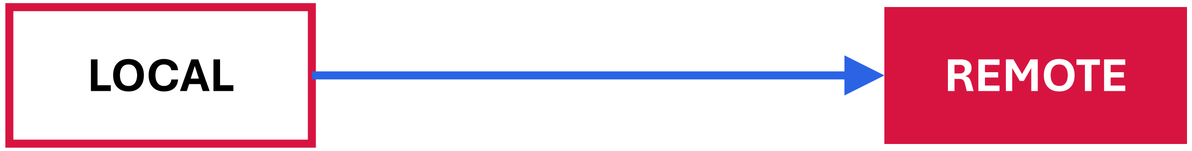
Pull Command

The git pull command is used to fetch and integrate changes from a remote repository (like GitHub) into your local repository. It is a combination of two commands: git fetch (which retrieves changes) and git merge (which integrates those changes).


REMOTE → LOCAL

Push Command

The git push command is used to upload your local commits to a remote repository. After making changes locally (commits), you push them to the remote repository (e.g., GitHub) so that others can access and collaborate on those changes.


LOCAL → REMOTE

Aspects of Git : Branch & Merge

Branching

A branch in Git is essentially a separate line of development. It allows you to isolate your work (e.g., for developing a feature or fixing a bug) without affecting the main project. By default, every Git project starts with the main branch, but you can create new branches to work independently.

The git merge command is used to integrate changes from one branch into another. Once work in a branch is complete, you can merge it back into the main branch or another branch.

Merging

Aspects of Git : Fork & Contribute

Fork

Forking a repository is a GitHub-specific feature that creates a copy of a repository in your GitHub account.

This is typically done when you want to contribute to someone else’s project. Forking allows you to experiment and make changes to the code without affecting the original project.

Contribute

Once you’ve made changes in your forked repository, you can contribute those changes back to the original repository by submitting a pull request (PR).

This asks the repository owner to review and possibly merge your changes into the original project.

Thank You

Any Questions?