Version Control Foundations & GitHub Setup

AiCE Software Engineering Foundations — The Developer Environment

Every developer eventually loses work they wish they hadn't — a good change overwritten by a bad one, with no way back. Version control exists to make that scenario nearly impossible. This lesson introduces what version control actually solves, the relationship between Git and GitHub, and how to get both properly set up on your machine and account.

1. What Is Version Control?

Imagine working on a codebase for three months. One morning, a change you made breaks the entire application, and you need to revert to the version from two days ago — but you've been overwriting the same files every day, with no record of what they looked like before. That scenario is a nightmare, and it happens routinely to developers who don't use version control.

Version control is a system that records the history of changes to your files over time. With it, you can:

See exactly what changed, when it changed, and who changed it.

Revert any file, or the entire project, back to a previous state.

Work on different features simultaneously without breaking the main codebase.

Collaborate with other developers without overwriting each other's work.

> **Git and GitHub**
> *Git is the most widely used version control system in the world, created by Linus Torvalds — the same engineer behind the Linux kernel. GitHub is a platform that hosts Git repositories online, making it easy to back up code, collaborate, and share projects publicly.*

2. Git vs. GitHub — Understanding the Difference

These two names get confused constantly. They're related, but distinct:

	Git	GitHub
What it is	Software installed on your computer	A website (github.com)
Where it runs	In your terminal	In a web browser
What it does	Tracks changes to your code locally	Hosts your repositories online
Internet required?	No (works offline)	Yes
Who made it?	Linus Torvalds (2005)	Founded 2008, acquired by Microsoft 2018

The relationship in one sentence: Git handles the versioning logic on your own machine, and GitHub provides the cloud storage where you send that versioned code. Together, they form the complete version control workflow that every developer relies on.

3. GitHub — Setting Up Your Account

Go to github.com in any browser and create an account if you don't already have one. Use a professional email address and choose a username that represents you well — your GitHub profile becomes part of your developer portfolio from day one.

Once logged in, take a moment to explore three key areas of the interface:

Repositories — containers where your code projects live.

Profile page — your public developer identity.

Explore page — where you can discover open source projects.

What Is a Repository?

A repository (often shortened to "repo") is like a special folder managed by Git. It stores all your project files and tracks the complete history of every change ever made to those files.

> **A Useful Mental Model**
> *Think of a repository the way you'd think of a file cabinet in an office. Each drawer holds the files for one project. But unlike an ordinary file cabinet, a Git repository remembers every version of every document that was ever placed inside it.*

Creating Your First Repository

On GitHub, click the green New button (or find it under the Repositories tab), then fill in:

Repository name — a descriptive name using hyphens instead of spaces, e.g. african-tech-directory.

Description — a brief sentence describing what the project is for.

Visibility — Public (anyone can view) or Private (only you and invited collaborators).

Click Create Repository, and your new, empty repository is live on GitHub.

> **Repository Naming Convention**
> *Use lowercase letters and hyphens; avoid spaces and special characters. Good: nairobi-bus-tracker. Bad: Nairobi Bus Tracker.*

4. Installing and Configuring Git

Check whether Git is already installed on your machine:

```
git --version
```

If installed, you'll see output resembling:

```
git version 2.43.0
```

If it's not installed, on Ubuntu:

```
sudo apt install git
```

Before using Git for the first time, you need to configure it with your identity — this information gets attached to every commit you make from this point forward:

```
git config --global user.email "you@example.com"
git config --global user.name "YourGitHubUsername"
```

> **Match Your GitHub Account**
> *Use the same email address you registered on GitHub, and your exact GitHub username, case-sensitive. This is what ties your local commits back to your GitHub identity.*

Key Takeaways

Version control gives you a permanent, searchable history of every change to your project — the safety net that makes fearless experimentation possible. Git is the local engine that tracks that history on your machine; GitHub is the cloud home where that history is stored, shared, and backed up. With your GitHub account created, your first repository live, and Git installed and configured with your identity, you're ready for the daily workflow covered in the next lesson: cloning, adding, committing, pushing, and pulling.