TECH TIP — BEGINNER

What Is GitHub and Why Does It Matter?

The version control platform that runs most of the internet's code

From Nolan Law Firm — nemolegal.com/tools

Start With Git, Then GitHub

Git is a version control system — software that tracks every change you make to files over time. GitHub is a website that hosts your Git repositories (more on that word in the next guide) and adds collaboration tools on top. Think of Git as the engine and GitHub as the garage where you park and share your cars.

You can use Git without GitHub. You can't use GitHub without Git. Most people use them together, and that's what this guide assumes.

Why It Matters — Even If You Don't Write Code

GitHub is where the world's software lives. Every open-source tool you use — n8n, Docker, Hermes Agent, Ollama, VS Code — has its source code on GitHub. When you search for solutions to technical problems, GitHub is often where the answer lives. When you build your own tools and want to back them up, collaborate, or share them, GitHub is the standard place to do it.

GitHub Does This	Which Means
Stores your code with full history	You can roll back any change, ever
Hosts your projects publicly or privately	Share with the world or keep it locked down
Lets others contribute via pull requests	Collaboration without overwriting each other's work
Runs automated tasks on code changes	Tests, deployments, notifications triggered automatically
Issues and project boards	Bug tracking and task management built in
GitHub Pages	Free static website hosting directly from a repo

The Overlooked Parts Most People Miss

Most introductions to GitHub focus on code collaboration. That misses a lot of what makes it valuable for non-developers:

Search. GitHub's search finds code, files, and projects across millions of repositories. If you're looking for an example of how to do something — a working n8n workflow, a Jinja2 template, a docker-compose.yml for a specific app — GitHub search often finds a working example faster than any tutorial.

Stars and watching. You can star repos you find useful (like bookmarks) and watch repos to get notified when they update. If you're following Hermes Agent, autoresearch, or any other active project, watching the repo keeps you current.

Issues as documentation. The Issues tab on any repo is where bugs and questions live. If something isn't working, search the Issues first — someone has probably already asked and gotten an answer.

Releases. The Releases tab shows stable, tagged versions of a project. This is where you download production-ready builds rather than cutting-edge development code.

GitHub vs. the Alternatives

Platform	Who Uses It	Key Difference
GitHub	Most open-source, most companies	Largest community, most integrations
GitLab	Enterprises, self-hosters	Can run entirely on your own server
Bitbucket	Atlassian/Jira shops	Tight Jira integration
Codeberg	Privacy-focused developers	Non-profit, no tracking

For most solo practitioners and small firms, GitHub is the right choice. It has the largest community, the best search, and free private repos. GitLab is worth knowing about if client data means you need self-hosted infrastructure.

> **Using AI to Help With This** You don’t have to fully understand this to use it. Here are prompts that work:

```
"I found a GitHub repository for [tool]. Explain what I’m looking at and how to figure out if it’s actively maintained and worth using."
```

```
"Search GitHub for examples of [specific thing — n8n webhook workflow, Jinja2 legal template, etc.] and explain what the best results are doing."
```

```
"I want to follow the development of [project] on GitHub. What should I watch, star, or subscribe to in order to stay current?"
```