New member guide

Everything you need before your first project. Plan on about an hour. Do the first section before your first meeting; the rest can happen at working time.

Before your first meeting

1. **Join the Slack.** Meeting links, help, and announcements live there. isenbergailab.netlify.app/slack
2. **Read the guardrails.** Then confirm on the signup form. Code of conduct and guardrails (`isenbergailab.netlify.app/docs/guardrails.html`)
3. **Subscribe to the Substack.** Monthly reports on what members built. isenbergailab.substack.com
4. **Bring a laptop.** Any operating system. A Chromebook works if you use the browser option below.

Accounts

1. **GitHub.** Create a free account at github.com. Add your UMass email to it.
2. **Student Developer Pack.** Apply with your UMass email at education.github.com. It adds free tools, including the free Copilot Student plan.
3. **Say hello.** Post your GitHub username in the Slack, so officers recognize your first pull request.

Pick where you build

Lab code never runs directly on your laptop's main file system. You have two sandboxed options.

Your laptop, in Docker	Recommended. Install the tools below once. Every project runs in a container you can delete.
GitHub Codespaces	Nothing to install. A disposable cloud workspace in your browser. Personal accounts get about 60 free hours a month on the smallest machine; stop it when you are done. Open the Lab repo on GitHub, choose Code, then Codespaces. Skip to Tell Git who you are.

Install the tools

Pick your system. The commands install Git, Python, VS Code, and Docker Desktop.

WINDOWS · POWERSHELL

```
# Docker on Windows needs WSL. Restart after this one.
wsl --install

winget install --id Git.Git -e
winget install --id Python.Python.3.12 -e
winget install --id Microsoft.VisualStudioCode -e
winget install --id Docker.DockerDesktop -e
```

MACOS · TERMINAL

```
# Install Homebrew first if you don't have it: https://brew.sh
xcode-select --install
brew install git python@3.12
brew install --cask visual-studio-code docker-desktop
```

Open Docker Desktop once after installing, and leave it running while you work. On a Mac, Docker Desktop needs macOS 14 or newer.

Tell Git who you are

Use your GitHub noreply address to keep your personal email out of public commits. Find it under GitHub, Settings, Emails.

TERMINAL

```
git config --global user.name "Your Name"
git config --global user.email "12345678+username@users.noreply.github.com"
git config --global init.defaultBranch main
```

Get the repo and run the checks

This proves your setup works. The last line should say the guardrail checks passed.

WINDOWS · POWERSHELL

```
git clone https://github.com/isenbergailab/ISENBERG-AI-LAB-REPO.git
cd ISENBERG-AI-LAB-REPO
python -m venv .venv
.venv\Scripts\Activate.ps1
pip install -r requirements.txt
python scripts/check_repo.py
```

MACOS · TERMINAL

```
git clone https://github.com/isenbergailab/ISENBERG-AI-LAB-REPO.git
cd ISENBERG-AI-LAB-REPO
python3 -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
python scripts/check_repo.py
```

On Windows, if PowerShell blocks the activate script, run `Set-ExecutionPolicy -Scope CurrentUser RemoteSigned` once.

Test Docker

If this prints a hello message, containers work on your machine.

TERMINAL

```
docker run --rm hello-world
```

API keys and spending

You pay for your own model usage. The guardrails keep that small and safe.

- **One key per project.** Never reuse a key across projects.
- **Hard cap at the provider.** $10 a month by default. Set it on the provider's site, not in your code.
- **Keys live in `.env`.** It is gitignored. Commit a `.env.example` with placeholders instead.
- **Never paste a key** into Slack, a chat assistant, a screenshot, or a commit.
- **Leaked a key?** Revoke it the same day, then tell an officer. No one is penalized for reporting fast.

Many members use OpenRouter: one account reaches many models, and each key can carry its own credit limit.

Optional: run models locally

Local models keep data on your machine and cost nothing per call. They are the best choice for anything sensitive. Install Ollama, then pull a small model to start.

WINDOWS · POWERSHELL

```
winget install --id Ollama.Ollama -e
ollama run qwen3.5:4b
```

MACOS · TERMINAL

```
brew install ollama
ollama serve   # leave running in one window
ollama run qwen3.5:4b
```

Set up your AI assistant

Whatever you use, Claude, Copilot, Cursor, or a chat window, point it at the Lab's rules before it writes anything. Details: Rules for AI assistants (isenbergailab.netlify.app/docs/ai-assistants.html).

SAY THIS FIRST

```
Read AGENTS.md at the root of the Isenberg AI Lab repo and follow it for every change you su
ggest or make.
```

You are ready when

1. You're in the Slack and your GitHub username is posted.
2. You confirmed the guardrails on the signup form.
3. `python scripts/check_repo.py` passes on your machine or in Codespaces.
4. `docker run --rm hello-world` works, or you're using Codespaces.
5. You know where your keys live and what your cap is.

Next: Your first project (isenbergailab.netlify.app/docs/first-project.html).