0️⃣ One-time project setup (only once per machine)

```
# Go to where you want the project
cd /Users/aarushimahajan/Desktop

# Clone your agentic project
git clone https://github.com/AarushiMahajan001/agentic-doc-maintainer.git
cd agentic-doc-maintainer

# Create and activate virtual env
python3 -m venv .venv
source .venv/bin/activate

# Install deps
pip install -r requirements.txt

# Create .env with your Groq key
cat > .env << 'EOF'
GROQ_API_KEY=your_groq_api_key_here
GROQ_MODEL_NAME=llama-3.1-70b-versatile
EOF
```

1️⃣ Per-session setup (every new terminal)

```
cd /Users/aarushimahajan/Desktop/agentic-doc-maintainer
source .venv/bin/activate
```

2️⃣ Run on any GitHub repo (CLI pipeline)

2.1 (Optional) Clear old repos / index / docs

```
cd /Users/aarushimahajan/Desktop/agentic-doc-maintainer
source .venv/bin/activate

rm -rf data/repo/*
rm -rf data/index/*
rm -rf data/docs/*
```

2.2 Clone the target GitHub repo into `data/repo`

Template:

```
cd /Users/aarushimahajan/Desktop/agentic-doc-maintainer

git clone https://github.com/USERNAME/REPO_NAME.git \
  data/repo/REPO_NAME

ls data/repo
ls data/repo/REPO_NAME
```

2.3 If the code is in a notebook → convert to `.py` (optional)

```
cd /Users/aarushimahajan/Desktop/agentic-doc-maintainer/data/repo/REPO_NAME

# Convert notebook to script
jupyter nbconvert --to script "NOTEBOOK_NAME.ipynb" \
  --output module_name.py

# If it creates something odd like module_name.py.py:
mv module_name.py.py module_name.py
# Or if it creates a .txt:
# mv module_name.txt module_name.py
```

Now you have `data/repo/REPO_NAME/module_name.py` (or some path inside that repo).

2.4 Build / rebuild FAISS index

```
cd /Users/aarushimahajan/Desktop/agentic-doc-maintainer
source .venv/bin/activate

python scripts/ingest_repo.py

ls data/index      # should show: code.index  metadata.json
```

2.5 Run the agentic pipeline on a module

Very important mapping:

- Real file on disk:
  `data/repo/REPO_NAME/path/inside/repo/module.py`

- Argument to script:
  `REPO_NAME/path/inside/repo/module.py`

```
cd /Users/aarushimahajan/Desktop/agentic-doc-maintainer
source .venv/bin/activate

python scripts/run_cli_demo.py REPO_NAME/path/inside/repo/module.py
```

Example (your stock project):

```
python scripts/run_cli_demo.py \
  Event-Driven-Stock-Prediction-Using-NLP/stock_price_prediction.py
```

Inspect docs:

```
ls data/docs
cat data/docs/REPO_NAME_path_inside_repo_module.py.md
```

3️⃣ Run evaluation benchmark on multiple modules

1. Edit `eval/tasks.yaml` to list all tasks:

```
cd /Users/aarushimahajan/Desktop/agentic-doc-maintainer
source .venv/bin/activate

cat > eval/tasks.yaml << 'EOF'
- name: "Some task name"
  module_path: "REPO_NAME1/path/inside/repo/module1.py"
  query: "short natural-language description of what you care about"

- name: "Another task name"
  module_path: "REPO_NAME2/path/inside/repo/module2.py"
  query: "another focus query"
EOF
```

2. Run the benchmark:

```
python eval/run_benchmark.py
```

You'll see per-function scores and overall averages in the terminal, and docs in `data/docs/`.

4️⃣ Run the Streamlit frontend

Make sure you already ingested repos with `python scripts/ingest_repo.py` (step 2.4).

```
cd /Users/aarushimahajan/Desktop/agentic-doc-maintainer
source .venv/bin/activate

streamlit run frontend/app.py
```

Then in the browser (http://localhost:8501):

- For **module path**, enter either:
    - `REPO_NAME/path/inside/repo/module.py` (relative to `data/repo`), or
    - If you're using the GitHub-URL mode we added, paste the GitHub URL.
- Optionally type a focus query.
- Click **Run pipeline**.

Docs + scores will appear in the UI and also be written to `data/docs/`.

5️⃣ Run the pipeline on local code instead of GitHub

If you have a local project:

```
/Users/aarushimahajan/Desktop/my_local_project
```

with `.py` files:

```
cd /Users/aarushimahajan/Desktop/agentic-doc-maintainer
source .venv/bin/activate

# Copy local project into data/repo
cp -R /Users/aarushimahajan/Desktop/my_local_project data/repo/

# Rebuild index
python scripts/ingest_repo.py

# Run pipeline on a module
python scripts/run_cli_demo.py my_local_project/path/inside/project/module.py

# Check docs
ls data/docs
cat data/docs/my_local_project_path_inside_project_module.py.md
```