**1. Install dependencies**

In your project root (news-summarization-agent-langgraph/):

pip install -r requirements.txt

If you don’t have a requirements.txt, create one with at least:

langgraph

langchain

scikit-learn

requests

feedparser

openai

python-dotenv

**⚙️ 2. Configure environment**

Create a .env file in the project root:

OPENAI\_API\_KEY=your\_openai\_key\_here # if using OpenAI

NEWS\_API\_KEY=your\_newsapi\_key\_here # optional, if using NewsAPI

👉 If you only want RSS (free), you can skip NEWS\_API\_KEY.

**▶️ 3. Run the agent**

Inside project root:

python main.py

This will:

1. Run the **collector agent** → fetch news (via NewsAPI or RSS).
2. Pass them to **summarizer** (uses LLM).
3. **Categorizer** tags them (tech, politics, etc.).
4. **Filterer** applies your rules (you can set keywords, date, categories).
5. **Reporter** generates a Markdown/HTML report.

**📄 4. View the output**

After run, you’ll see something like:

=== Final News Report ===

# Tech

- \*\*Apple releases new iPhone\*\*

Summary: Apple unveiled its new iPhone featuring an upgraded camera system…

[Read more](https://link-to-news)

# Finance

- \*\*Markets rally after earnings\*\*

Summary: Stock prices surged as major companies beat expectations…

[Read more](https://link-to-news)

Depending on your reporter, it may also **save to file** (e.g., output/report.md).

**🧪 5. Custom run with parameters**

If you added CLI args in main.py, you can run like:

python main.py --categories tech finance --keywords "AI, climate" --days 1