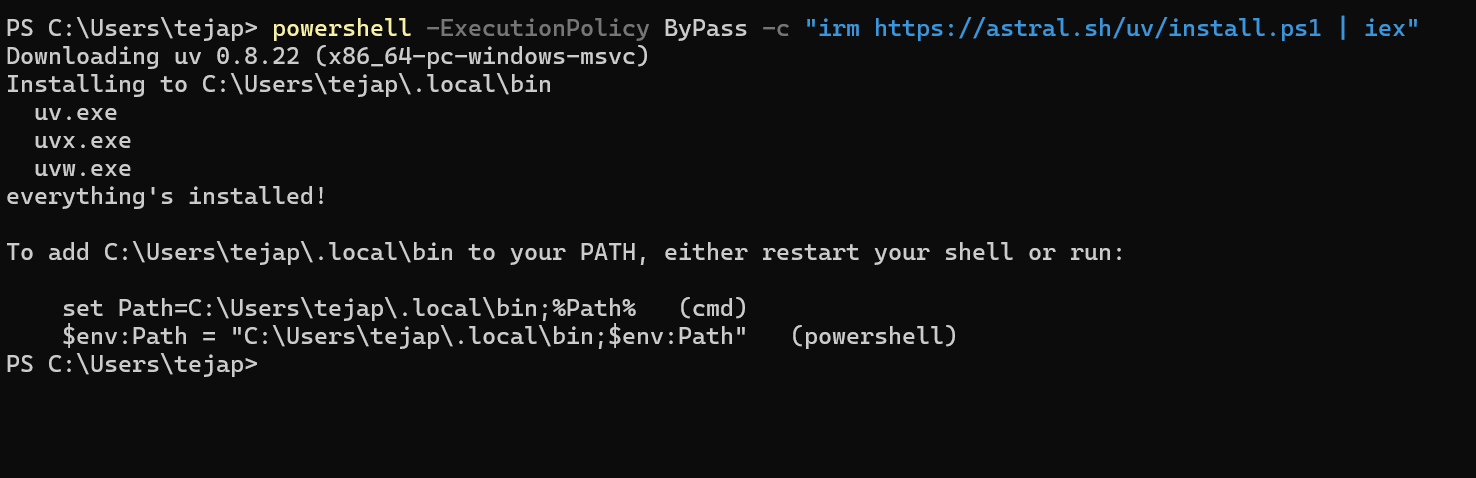
Step1: Clone my Repo – for notebook –

Step2: download cursor - <https://cursor.com/> and open cloned folder

Step3: install UV <https://docs.astral.sh/uv/getting-started/installation/>

Cmd: powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"



Step 4: uv sync (this creates isolated env, with python 3.12)

Run python file: uv run <filename>

Step 5: buy open ai key

<https://platform.openai.com/settings/organization/api-keys>

Step 6: add key in env variable

OPENAI\_API\_KEY=sk-proj-\*\*\*\*\*

**Step 7: Lab 1 – make an agentic workflow – only Open AI**

**Post lab discussion:**

Discussion1:

1. response = openai.chat.completions.create(

2.     model="gpt-4o-mini",

3.     messages=messages,

4. )

**vs**

1. const response = await openai.responses.create({

2. model: "gpt-5",

3. input: "Write a one-sentence bedtime story about a unicorn."

4. });

Discussion 2: what all models of GPT available for me: <https://platform.openai.com/docs/models>

**Step 8: Agentic AI Frameworks:**

****

**Complexity1:**

* **No Framework** using API’s like we did in lab1 / lab2 – connect directly to LLM’s - Custom-built agents from scratch using raw API calls and logic.
* **MCP** – Model Context Protocol by Anthrobic – it’s a way models can connect to each other – so we don’t need any glue code.

**Complexity2:**

* **OpenAI Agents SDK :** OpenAI’s official toolkit for building and deploying production agents.
* **Crew AI –** low code- config via yaml files : A lightweight agent-coordination library focused on simplicity and teamwork.

**Complexity 3: Heavy weight and learning curve, and great power**

* LangGraph: A graph-based extension of LangChain for orchestrating agent workflows.
* AutoGen : Microsoft’s decentralized multi-agent orchestration framework for collaboration.

**Step 9: Lab 2: Agentic Ai with Multi types of models**

**Model 1:** OpenAI: gpt-4o-mini (gpt-4o,o1,o3-mini) – key we have as part of Lab 1 – key working

Generate keys for below:

**Model 2:** Anthrophic: Claude-3-7-Sonnet - <https://console.anthropic.com/dashboard>.

Need 5$ minimum.

**Model 3:** Google: Gemini-2.0-flash – beta version

**Model 4:** Deepseek - <https://platform.deepseek.com/api_keys>. Free tier not available

**Model 5:** groq: open source LLMs including Llama3.3 - <https://console.groq.com/keys>

**Model 6:** Ollama – local service

Install it - <https://ollama.com/>, once installed and started it should be running on <http://localhost:11434/>

Recommended max model is llama3.2 or small variants of qwen, gemma, phi or deepseek

Available ollama model’s page: <https://ollama.com/search>

\*\*Restart your cursor for it to detect, ollama app running on our system.

**Post lab discussion:**

* Which pattern was it?
* Try completing the exercise.

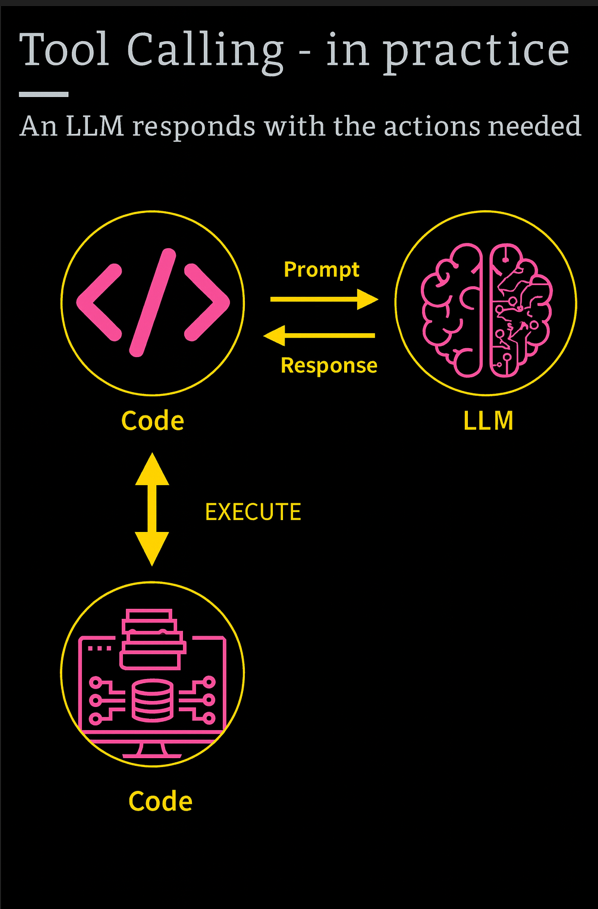
**Step 10: Terminology:**

**Resources:** provide LLM with resources to improve its expertise

i.e. shoving data relevant to the question into the prompt.

There are techniques like RAG to get really smart at picking **relevant context.**

**Tools:** give LLMs autonomy: give LLM power to carry out actions like “query a database” or “message other LLMs”.



**Step 11: Lab 3 – Feedback, Evaluator and optimizer methodology**

11.1: download your LinkedIn as pdf and save in the me folder

<https://www.linkedin.com/in/sai-teja-polisetty-92a7aa143/>

and write something about yourself into the summary.txt.

11.2: proceed with lab steps

\*\*Gradio : documentation - https://www.gradio.app/guides/sharing-your-app#hosting-on-hf-spaces

**Step 12: Lab 4 – First Project using tools (in previous lab we used resources) – Career Conversation**

12.1: Visit: <https://pushover.net/> ,

* Copy the user key, and keep in .env file with key **PUSHOVER\_USER**
* Create application: give some name
* Copy the token and keep in .env file with key **PUSHOVER\_TOKEN**

12.2: perform the steps in the lab, talk to tools

**Step 13: Deploy on hugging face**

Before you start: remember to update the files in the "me" directory - your LinkedIn profile and summary.txt - so that it talks about you! Also change `self.name = "Sai Teja"` in `app.py`..

**Delete the readme if any**

1. Visit https://huggingface.co and set up an account

2. From the Avatar menu on the top right, choose Access Tokens. Choose "Create New Token". Give it WRITE permissions - it needs to have WRITE permissions! Keep a record of your new key.

3. In the Terminal, run: `uv tool install 'huggingface\_hub'` to install the HuggingFace tool, then `hf auth login` to login at the command line with your key. Afterwards, run `hf auth whoami` to check you're logged in

4. Take your new token and add it to your .env file: `HF\_TOKEN=hf\_xxx` for the future

5. From folder where you have your app.py, enter: `uv run gradio deploy`

6. Follow its instructions: name it "career\_conversation", specify app.py, choose cpu-basic as the hardware, say Yes to needing to supply secrets, provide your openai api key, your pushover user and token, and say "no" to github actions.

**To delete your Space in the future:**

1. Log in to HuggingFace

2. From the Avatar menu, select your profile

3. Click on the Space itself and select the settings wheel on the top right

4. Scroll to the Delete section at the bottom

5. ALSO: delete the README file that Gradio may have created inside this 1\_foundations folder (otherwise it won't ask you the questions the next time you do a gradio deploy)