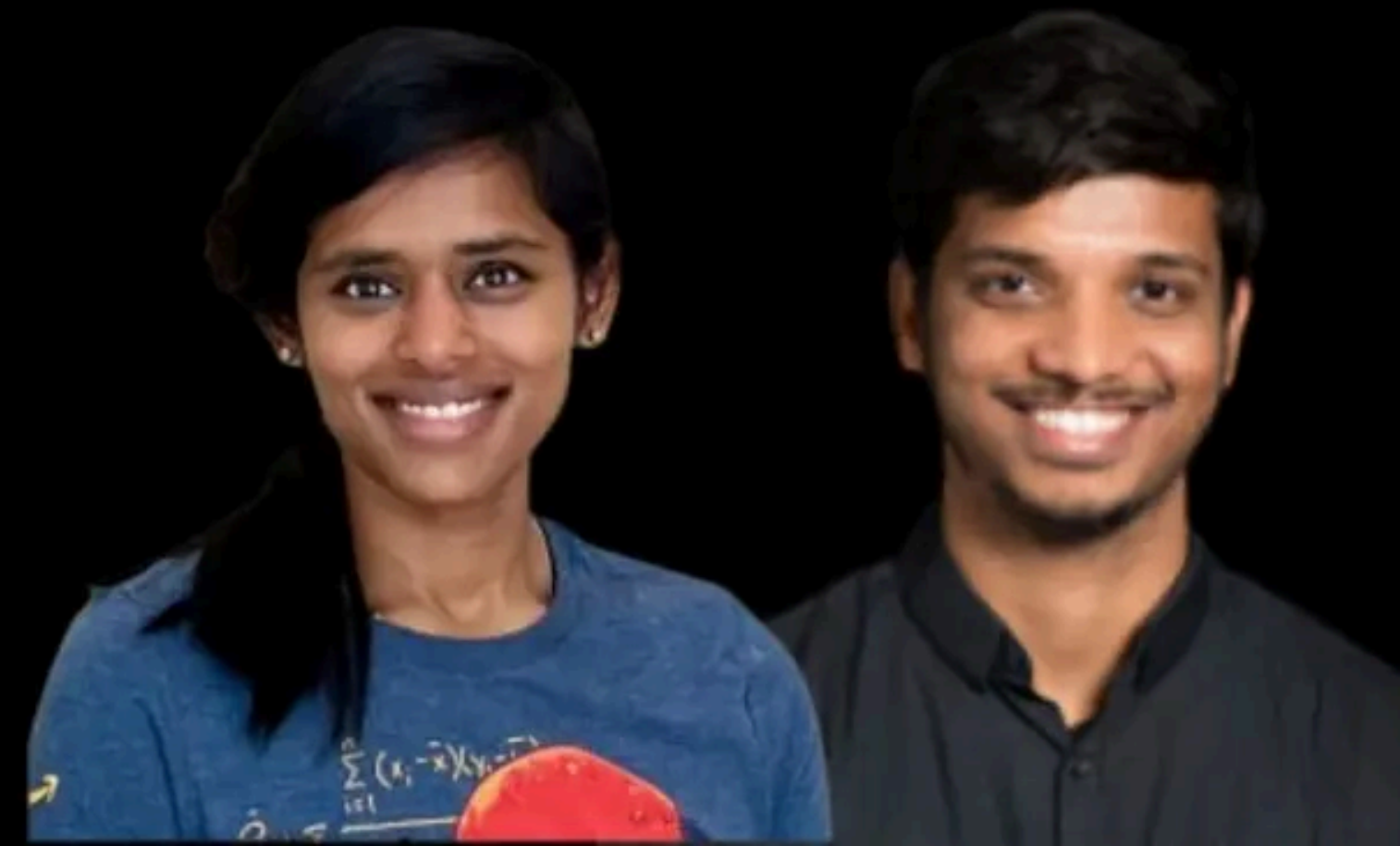


Building Agentic AI Applications with a **Problem- First** Approach

Aishwarya Naresh Reganti, Kiriti Badam

LangChain Setup and
Demo



LangChain - Full-Code Option for Assignments

Only pick one of LangFlow/LangChain

Environment Setup

- **Operating Systems:** MacOS / Linux / Windows
- **Tools & Frameworks:** Python, LangChain, LangGraph, OpenAI APIs.
- **AI Models:** OpenAI gpt-5-mini.
- Super simple setup, create Python virtual environment as shown.
- **Install required** libraries:
 - `uv pip install langchain==0.3.27 langgraph==0.6.7 langchain-openai==0.3.33 python-dotenv==1.1.1`
- Detailed instructions, specific for each platform are available in the Maven course page.

For macOS/Linux Users

1. Install uv

Open **Terminal** and run:

```
Bash ▾  
curl -LsSf https://astral.sh/uv/install.sh | sh
```

Required: Once this is done, you need to **restart the terminal** for uv to be available. You can either quit and open a new terminal, or open a new tab in the terminal.

2. Set up the course workspace or project directory

```
Bash ▾  
mkdir problem_first_ai  
cd problem_first_ai
```

3. Create and activate your virtual environment

```
Bash ▾  
uv venv --python 3.11.8  
uv python pin 3.11.8  
source .venv/bin/activate
```

Note: To deactivate a virtual environment on macOS and Windows, you can just run the command 'deactivate'

4. Confirm you're using the right Python

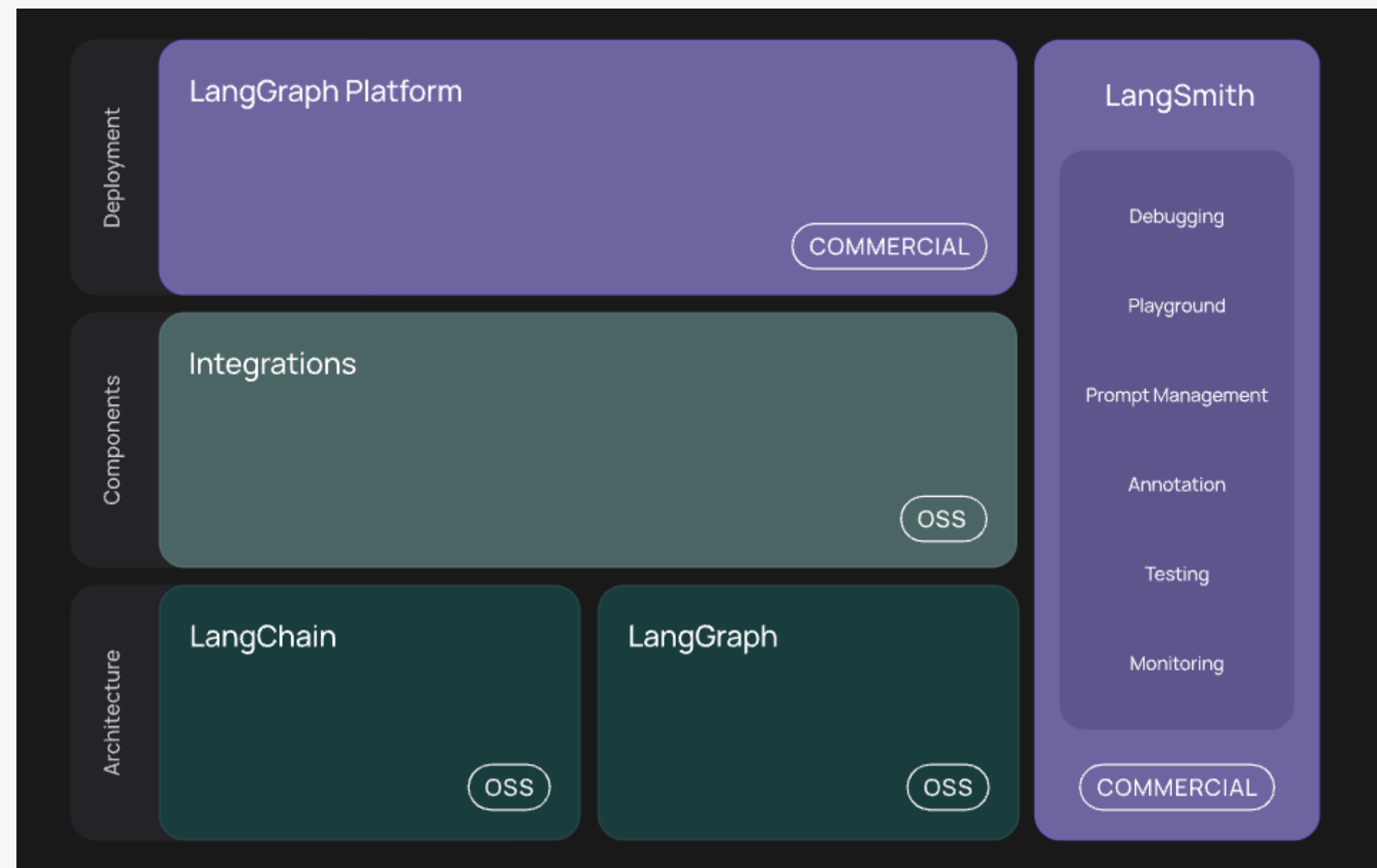
```
Bash ▾  
which python
```

Check that the output ends with `".venv/bin/python"`.

Screenshot from Maven Course Page

LangChain Ecosystem

- **LangChain**
 - One of the first and popular framework for building LLM applications.
 - Define components and chain them.
 - APIs evolved over time. **Use v0.3!!**
- **LangGraph**
 - Provides higher-level graph style constructs for building complex applications.
 - [Optional] LangGraph Platform – SaaS model for managed deployment.



Reference: <https://python.langchain.com/docs/introduction/>

LangChain

- Every component is implemented as a 'Runnable' i.e. it can be invoked/run and generates some output.
- Demo time
 - Chat Model
 - Message types (System, Human, AI)
 - Prompt templates
 - Chaining and invoke
- Read the docs for more info – https://python.langchain.com/docs/how_to/.