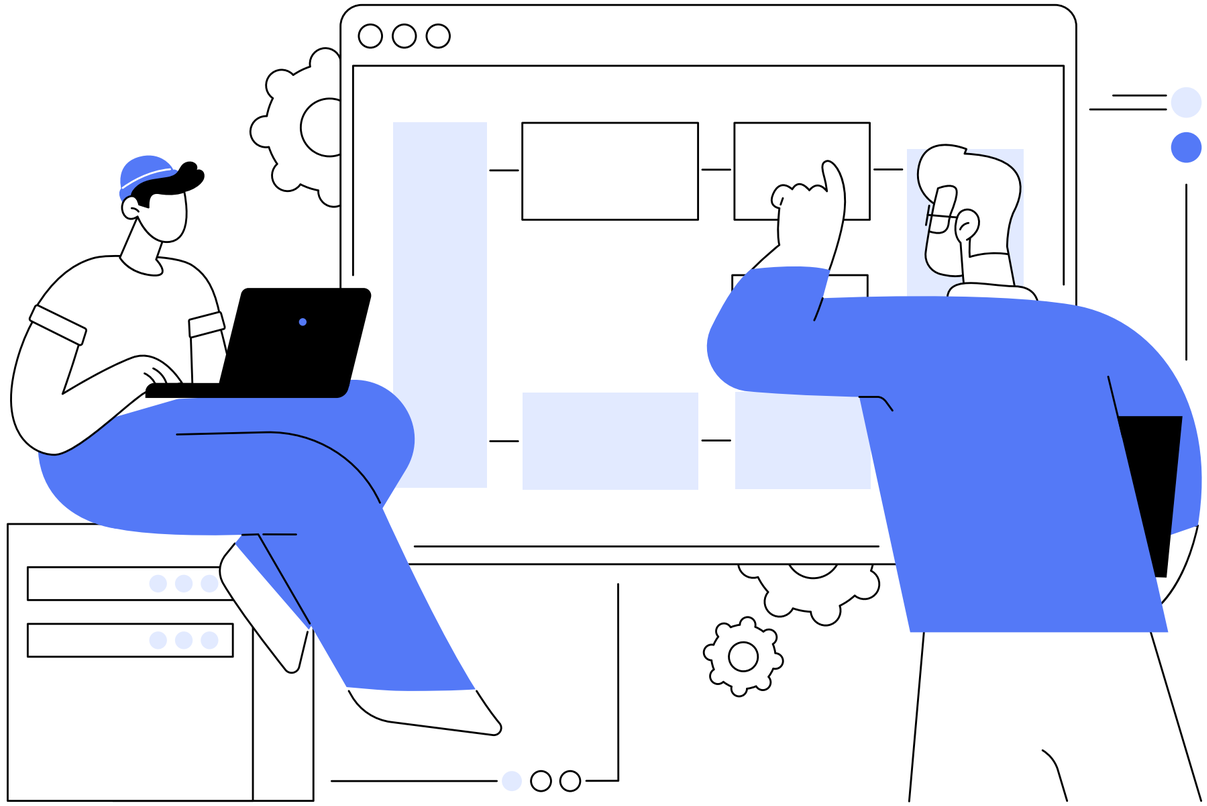
HOW TO BUILD A CHATBOT

Hands-On Workshop

INTRODUCTION

- Overview of the day’s agenda and workshop goals
- Introduction to workshop hardware NVIDIA Jetson Orin Nano
- Setting up the development environment



WORKSHOP AGENDA

Session 1

Theory:

Introduction to Large Language Models (LLMs)

Practise:

Deploy and use LLMs

Session 2

Theory:

Introduction to LangChain

Practise:

Use LangChain with LLMs

Session 3

Theory:

Introduction to Retrieval-Augmented Generation

Practise:

Deploy vector database, data integration & search

Session 4

Theory:

Introduction to RAG Chains in LangChain

Practise:

Implement a Q/A-RAG Chain

Session 5

Theory:

How to build a RAG-Chatbot

Practise:

Implement a RAG-Chatbot App

-> STEP BY STEP TO YOUR OWN CHATBOT

WORKSHOP TIMELINE

Time	Topic	Duration
09:00 AM	Start of Workshop	-
09:00 AM - 09:15 AM	Welcome and Introduction	15 min
09:15 AM - 09:30 AM	Session 1: Introduction to LLMs - Theory	15 min
09:30 AM - 10:30 AM	Session 1: Introduction to LLMs - Practical	60 min
10:30 AM - 10:45 AM	Session 2: Introduction to LangChain - Theory	15 min
10:45 AM - 11:45 AM	Session 2: Introduction to LangChain - Practical	60 min
11:45 AM - 12:00 PM	Session 3: Retrieval-Augmented Generation (RAG) - Theory	15 min
12:00 PM - 12:30 PM	Lunch Break	30 min
12:30 PM - 1:30 PM	Session 3: Retrieval-Augmented Generation (RAG) - Practical	60 min
1:30 PM - 1:45 PM	Session 4: Building a RAG-Chain - Theory	15 min
1:45 PM - 2:45 PM	Session 4: Building a RAG-Chain - Practical	60 min
2:45 PM - 3:00 PM	Session 5: Building the Chat Application - Theory	15 min
3:00 PM - 4:00 PM	Session 5: Building the Chat Application - Practical	60 min

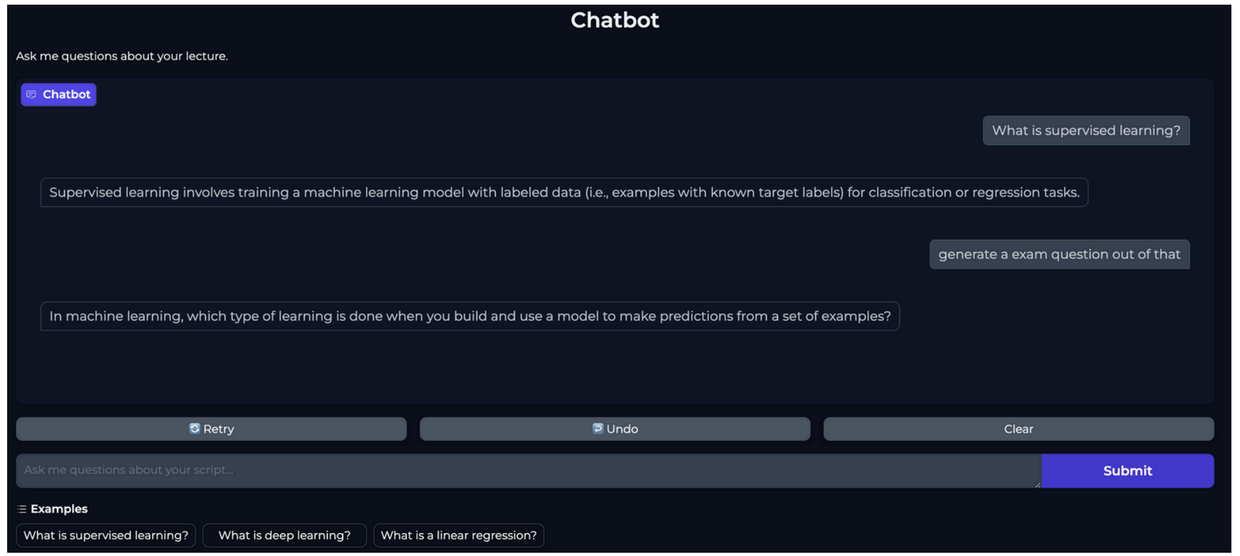
WORKSHOP GOAL

Personalized Learning Assistant:

- Create a chatbot that acts as a learning tutor.

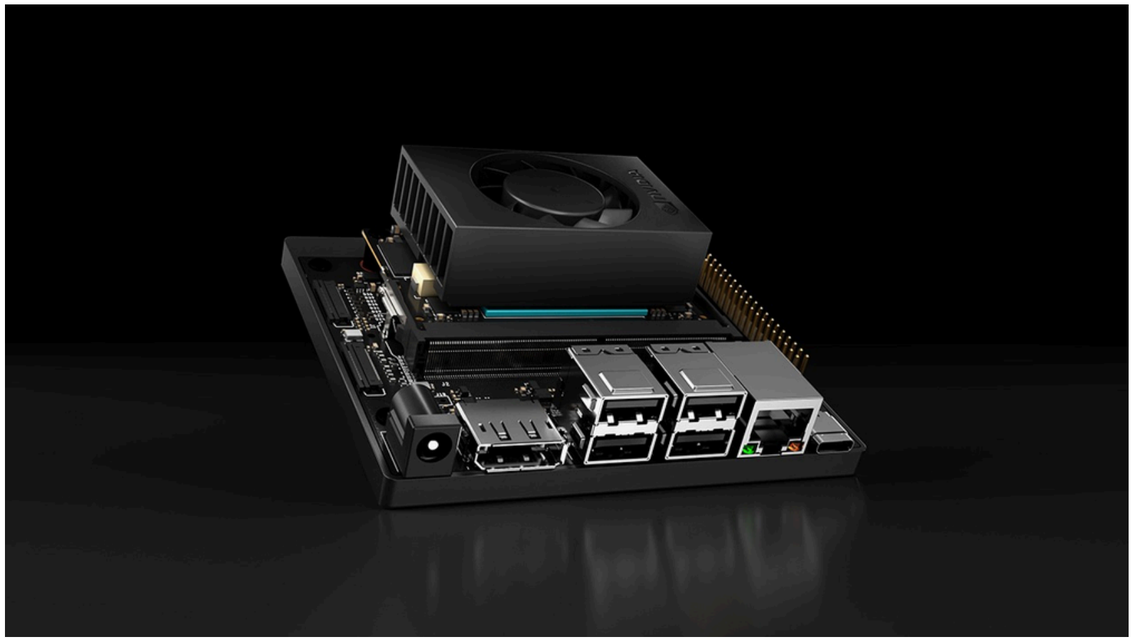
Interactive Study Tool:

- Use your own lecture script
- Ask questions about the content



NVIDIA JETSON ORIN NANO

- Edge AI platform
- ARM-based CPU with NVIDIA Ampere GPU
- Supports NVIDIA JetPack SDK and AI frameworks
- Ideal for on-device AI applications and models



DEVELOPMENT ENV

Hardware Layer:

- ARM CPU and NVIDIA Ampere GPU handle computing.

Operating System Layer:

- Ubuntu OS provides the base environment.

Development Tools Layer:

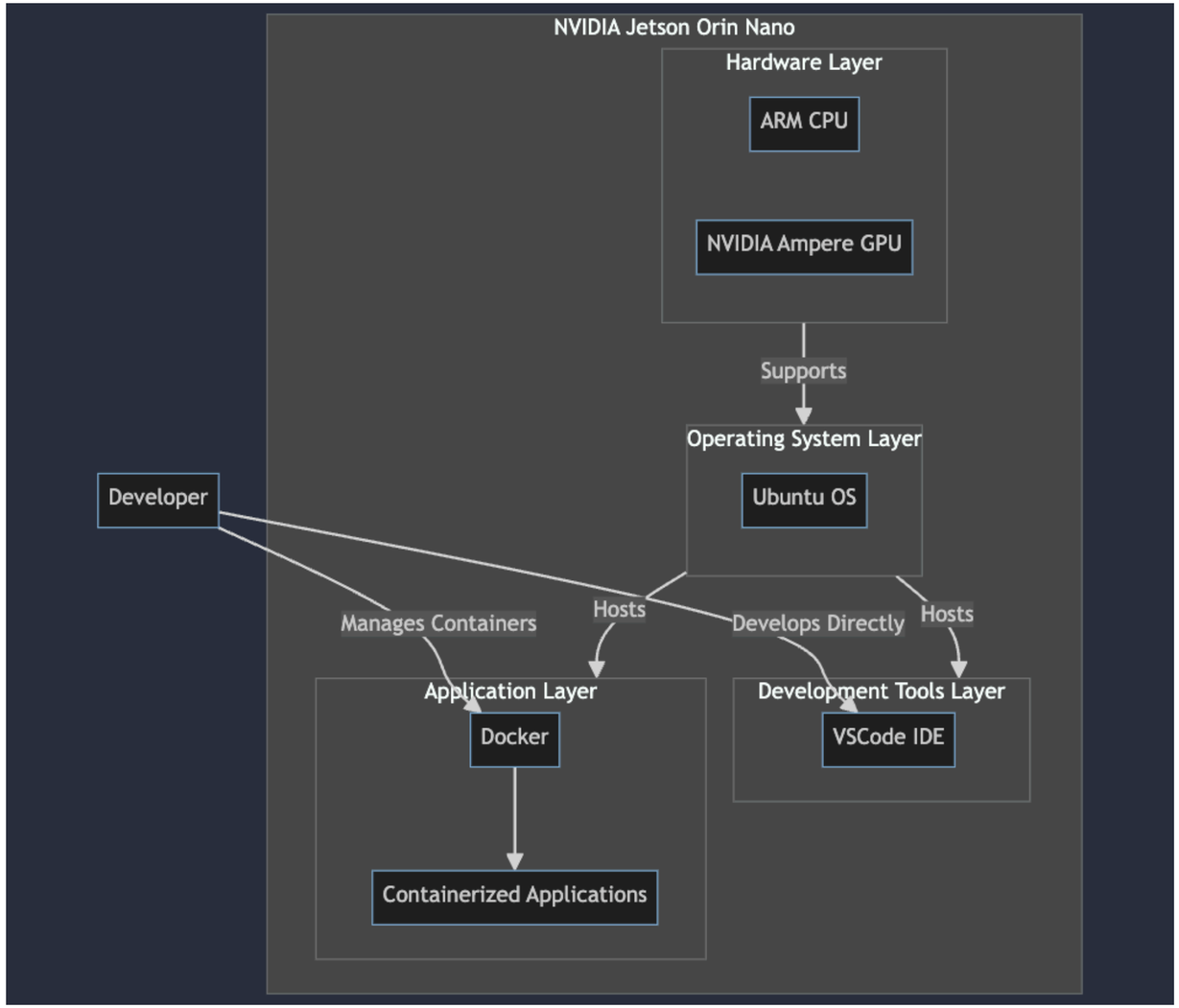
- VSCode IDE is used for direct development on the device.

Application Layer:

- Docker runs containerized AI applications.

Developer Interaction:

- Developers code and manage containers directly on the Orin Nano.



GOAL ARCHITECTURE

Frontend:

- Web app built with Gradio, accessible via browser.

Backend:

- Python-based with FastAPI and LangChain.

LLM Serving:

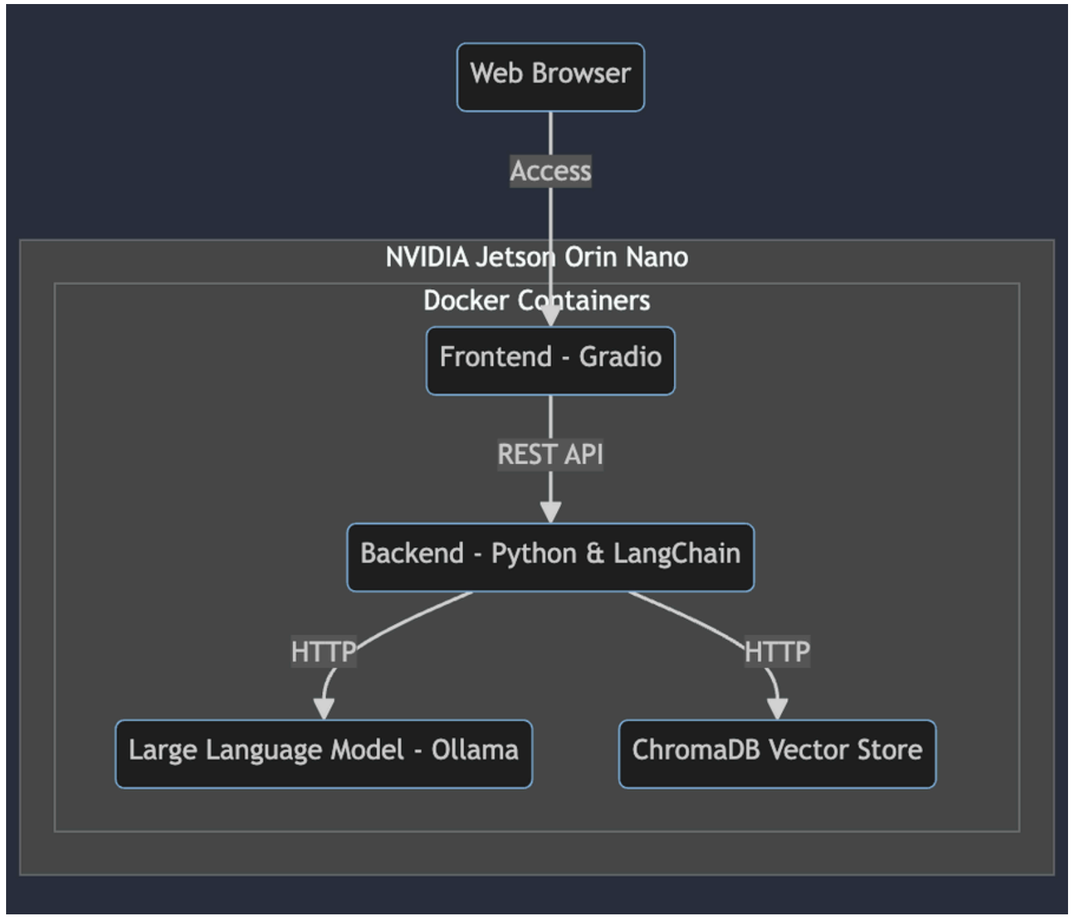
- Ollama for managing large language models.

Knowledge Storage:

- Vector database for knowledge management.

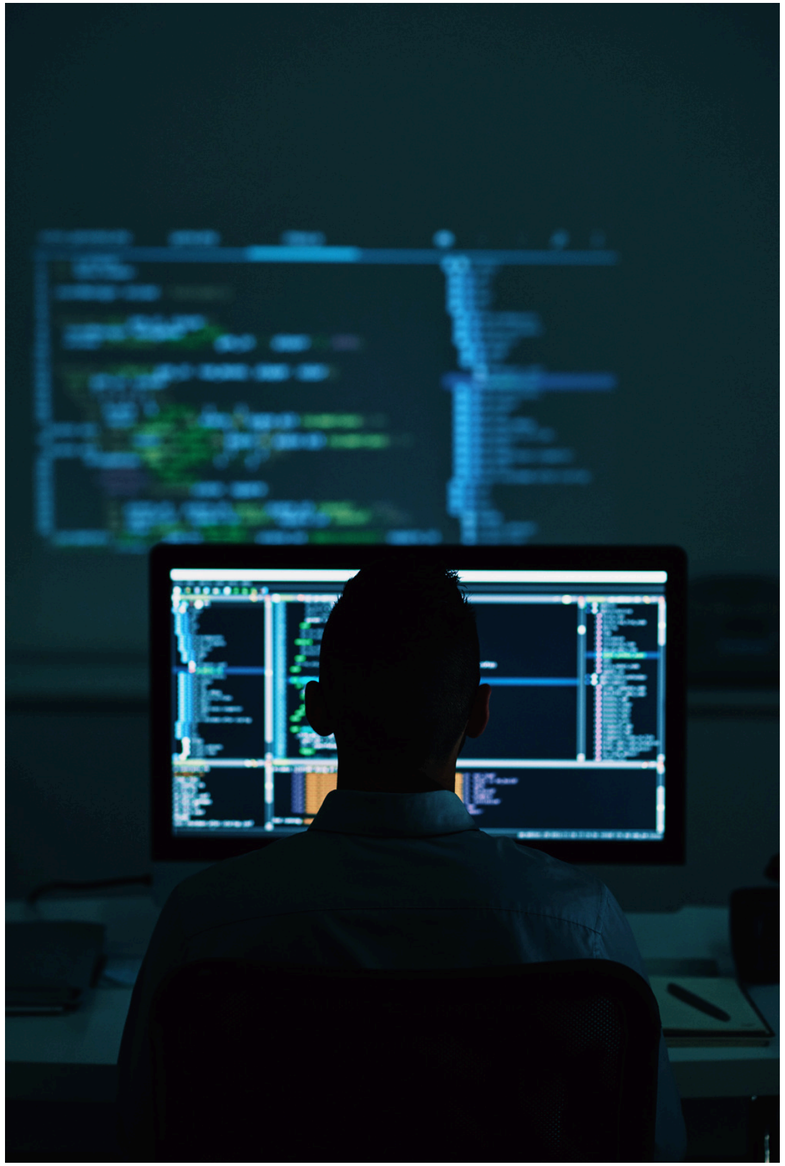
Deployment:

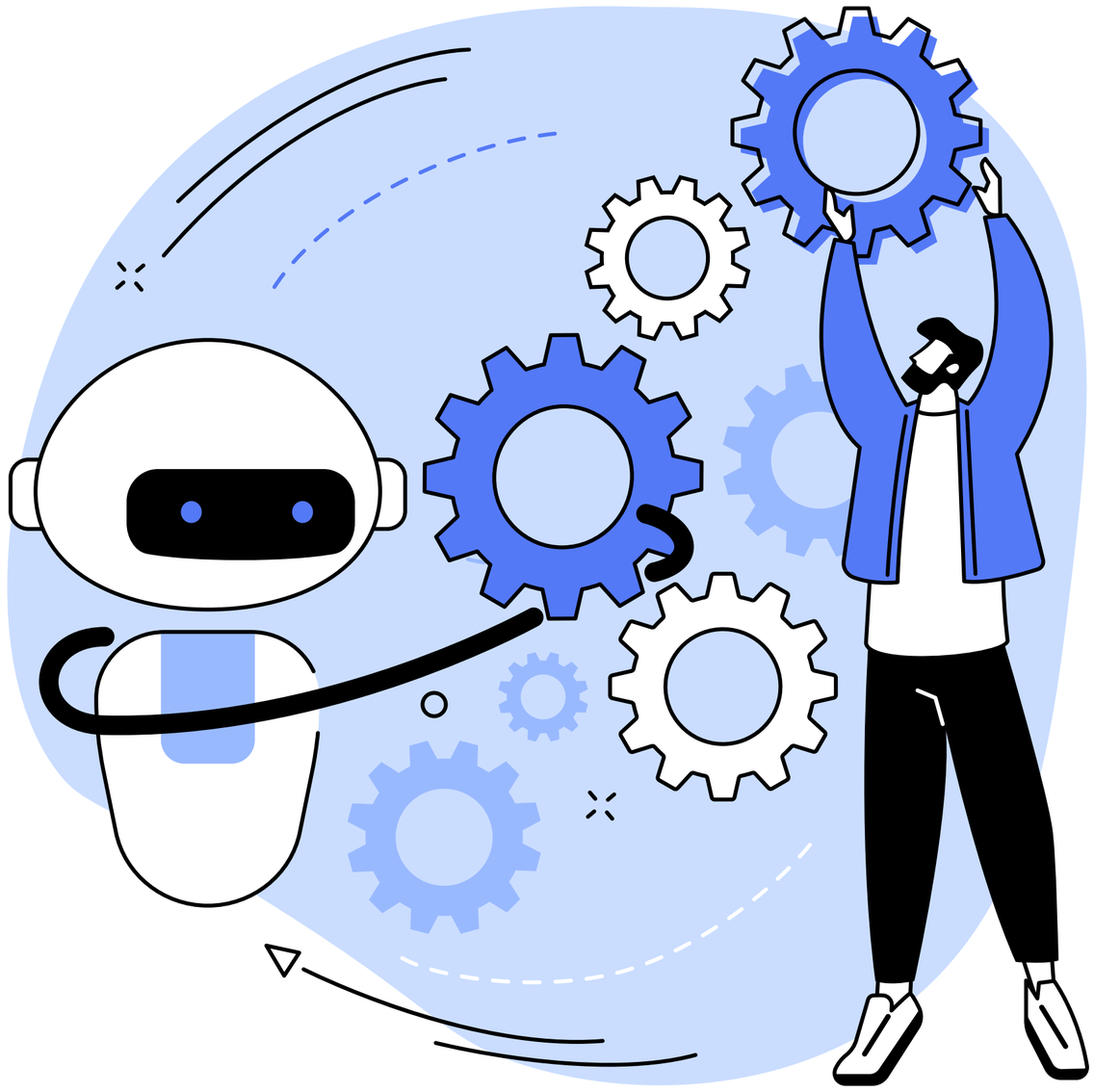
- Docker containers for application deployment.



STARTUP DEV ENV

- **Power On:**
  - Start NVIDIA Jetson Orin device.
- **Login:**
  - Authenticate with user credentials.
- **Launch VSCode:**
  - Open the development environment.
- **Open Repository:**
  - Access template project.
- **Verify Docker:**
  - Ensure Docker is running
- **Follow instructions** > “startup_dev_env.md”





IT’S YOUR TURN