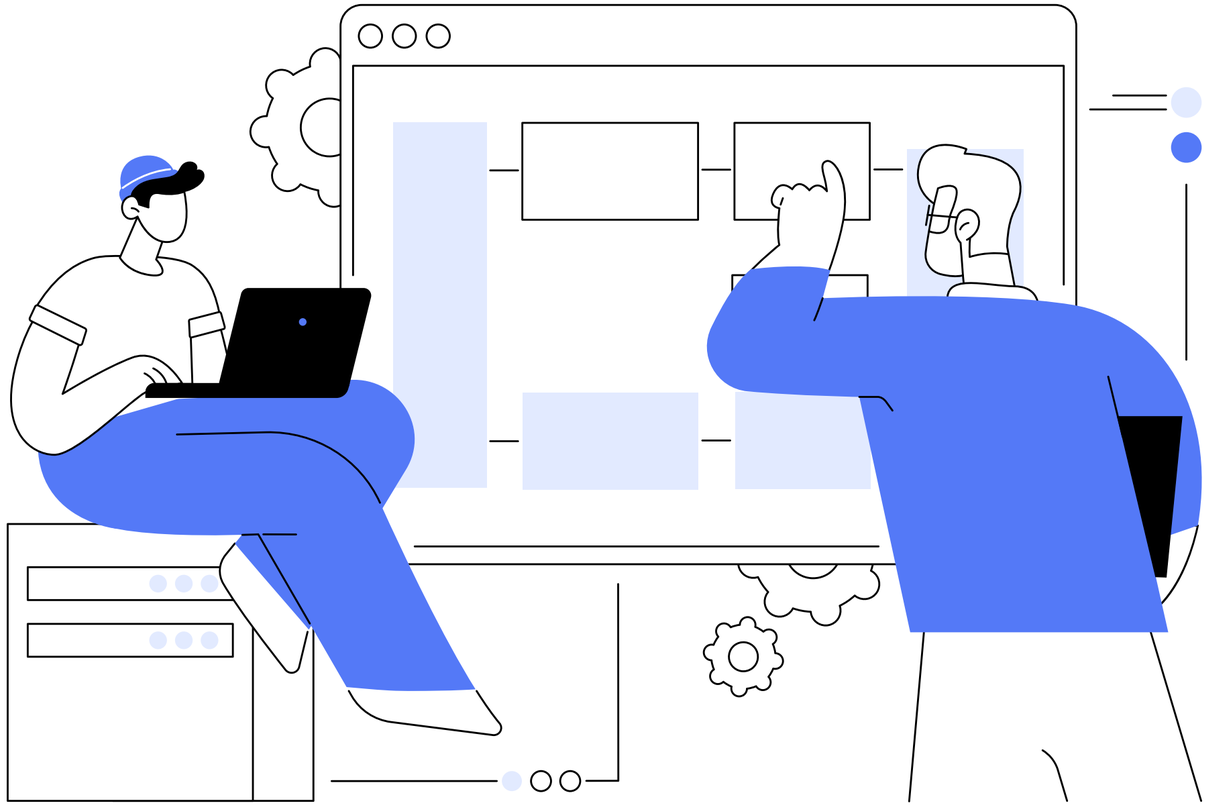
HOW TO BUILD A CHATBOT

Hands-On Workshop

WELCOME

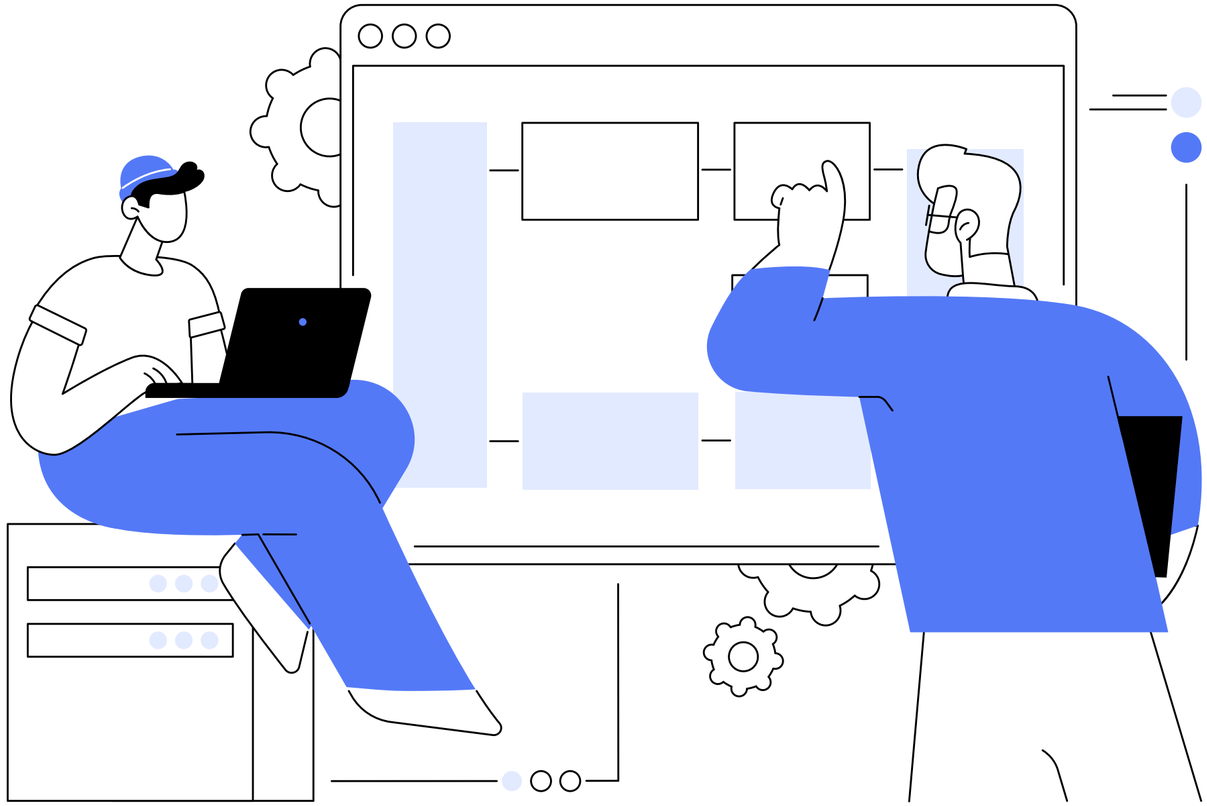
Martin Kovacs

- **AI Research Engineer @ Festo**
- **Lecturer Machine Learning @ HS Esslingen**
- **M.Sc. in Applied Informatics**
- **Research Field: Generative AI, LLM Agents, LLM Multi Agent Systems**



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

1. Session 1: Introduction to LLMs
2. Session 2: Introduction to LangChain
3. Session 3: Retrieval-Augmented Generation
4. Session 4: Building a RAG-Chain
5. Session 5: Building the Chatbot

WORKSHOP AGENDA

Session 1

Theory (20 min):
Introduction to Large Language Models (LLMs)

Practise (45 min):
Deploy and use LLMs

Session 2

Theory (20 min):
Introduction to LangChain

Practise (45 min):
Use langchain for accessing LLMs

Session 3

Theory (20 min):
Introduction to Retrieval-Augmented Generation

Practise (45 min):
Deploy vector database, data integration & search

Session 4

Theory (20 min):
Introduction in Chains and Agents

Practise (45 min):
Implement a RAG-Chat- Chain/Agent

Session 5

Theory (20 min):
Building the Chat Application

Practise (45 min):
Implement a RAG-Chat- Chain/Agent

-> STEP BY STEP TO YOUR OWN CHATBOT

WORKSHOP GOAL

Personalized Learning Assistant:

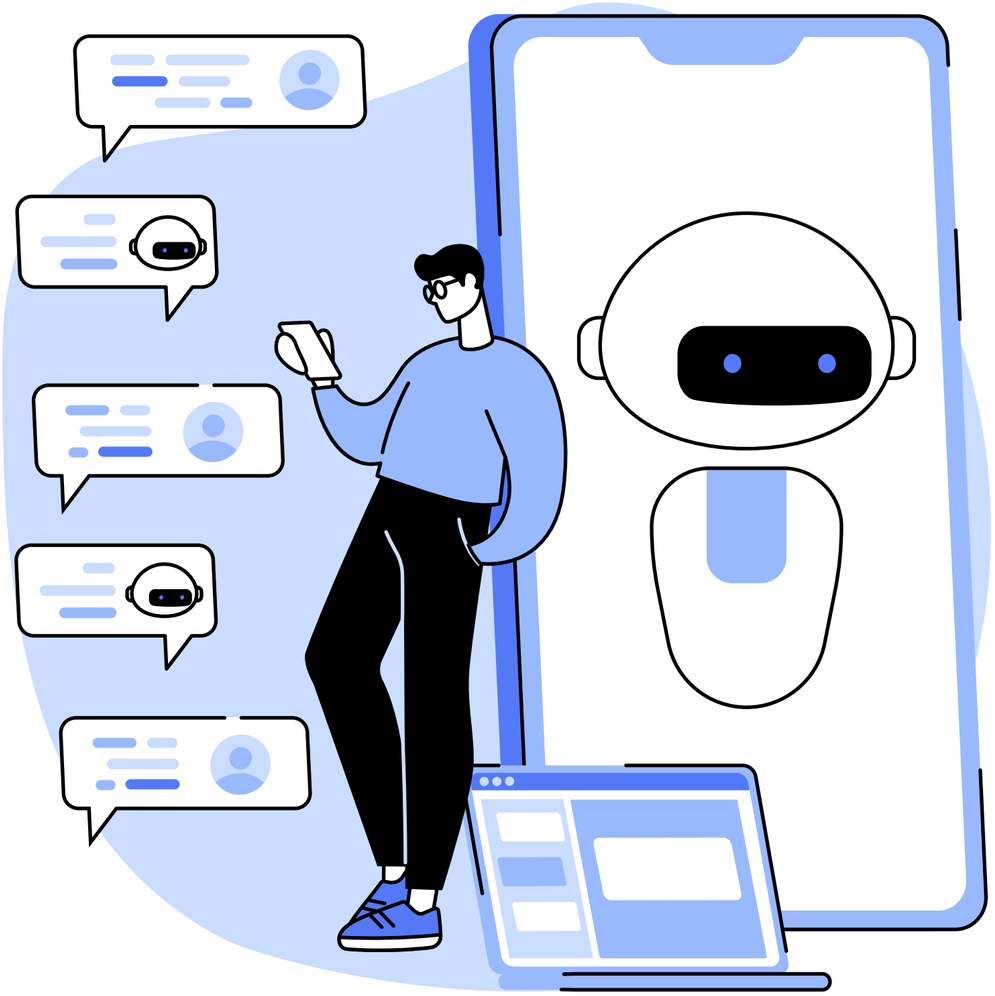
- **Create a chatbot that acts as a learning tutor.**

Interactive Study Tool:

- **Upload lecture scripts, ask questions about the content, or generate exam-related questions.**

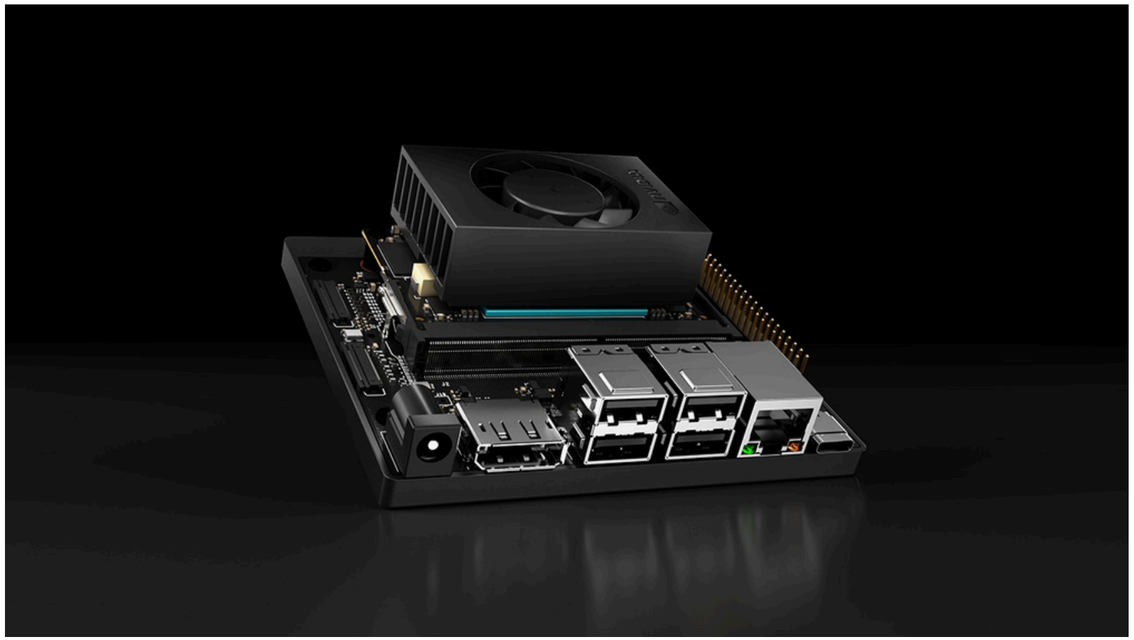
Exam Preparation Support:

- **Use the chatbot to reinforce your understanding of key topics.**



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.**

Application Layer:

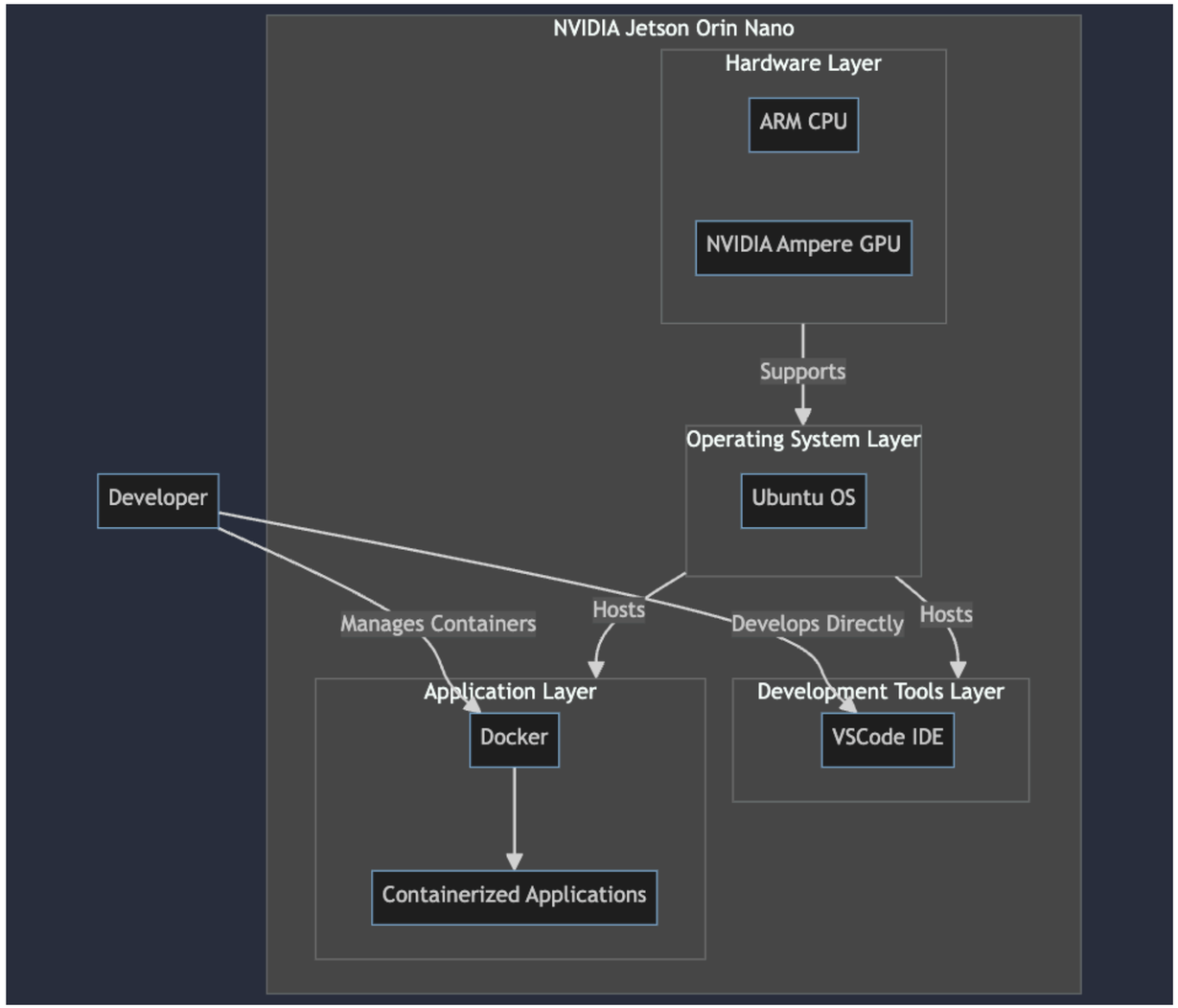
- **Docker runs containerized AI applications.**

Development Tools Layer:

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

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:

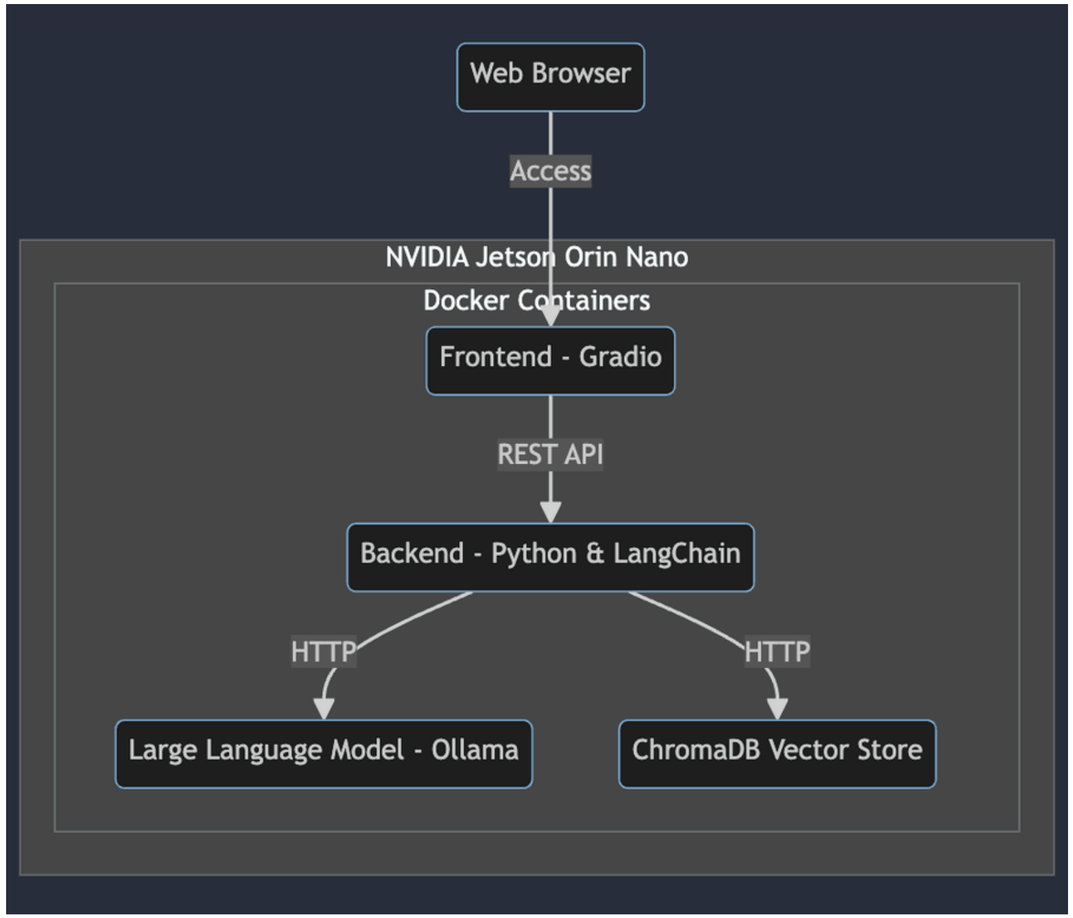
- **Ollama for managing large language models.**

Knowledge Storage:

- **Vector database for knowledge management.**

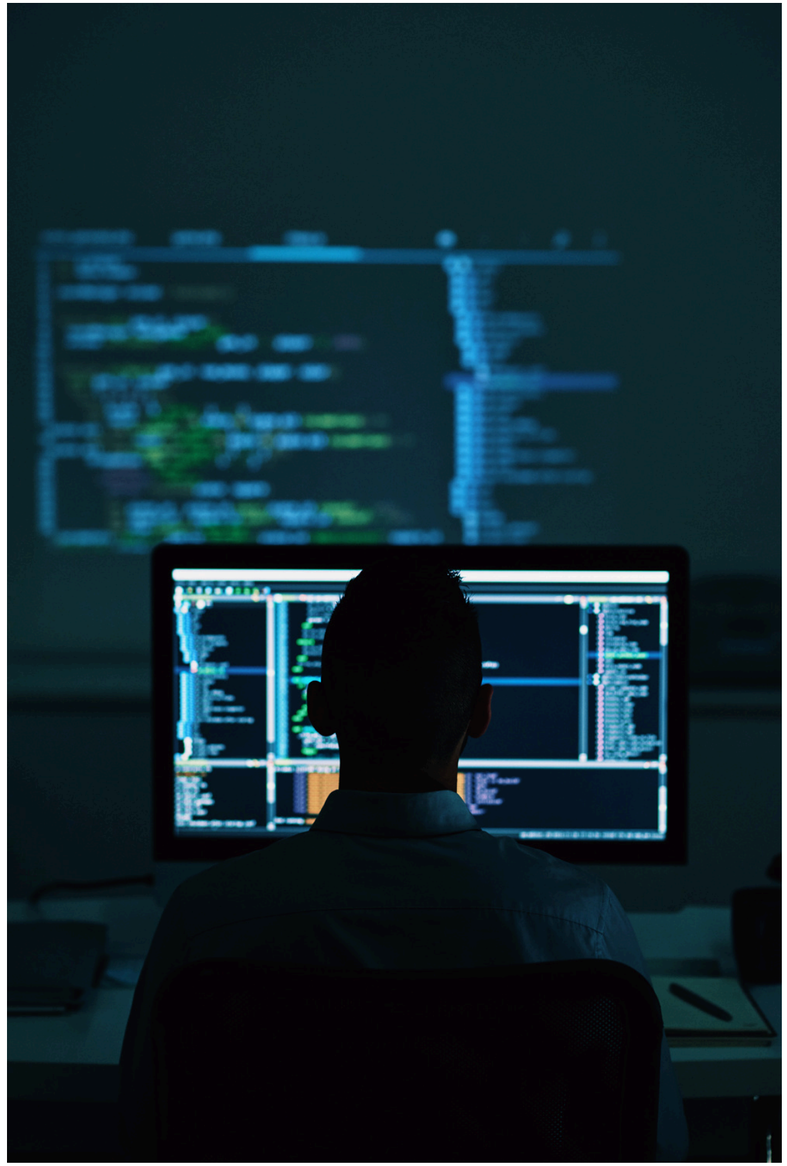
Deployment:

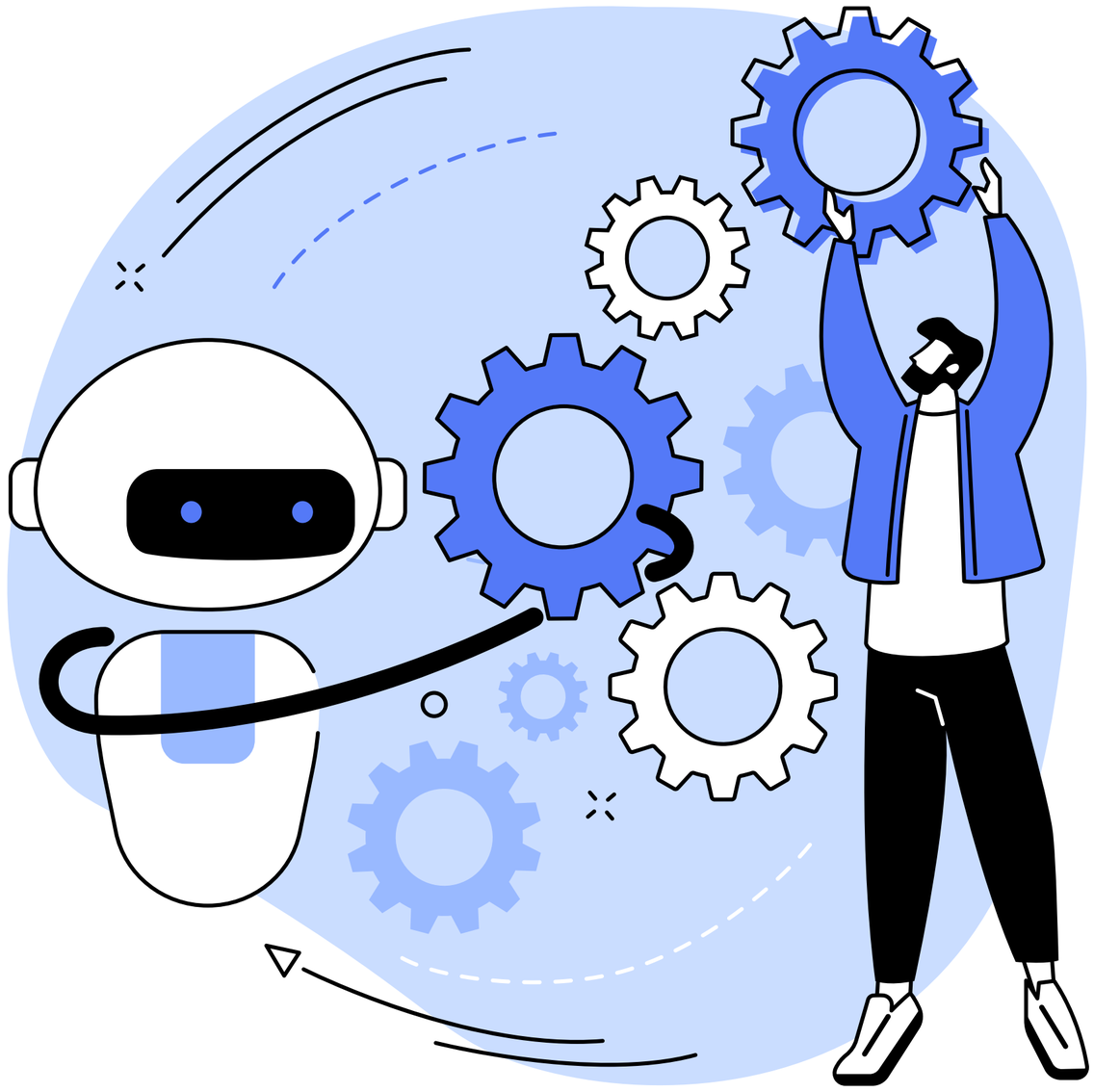
- **Docker containers for application deployment.**



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 from Git.**
- **Verify Docker: Ensure Docker is running.**





IT’S YOUR TURN