Open4Tech Summer School 2025

10 Skills Needed in the AI Era

Radu Capalb

Software Developer, Syncro Soft

Agenda

- The AI revolution
- 10 skills to thrive in the AI world
- Vibe coding
- What is Cursor AI?
- Building a tic tac toe app using Cursor AI – demo
- Key takeaways

The AI revolution

- You see 'AI' everywhere.
- Moving beyond the hype

- Will AI replace programmers?
- It's about augmentation and evolution
  - Code generation, debugging, learning

- The shift from assembly language to compilers
- AI is the next step in that evolution

10 skills to thrive in the AI world

AI fundamentals

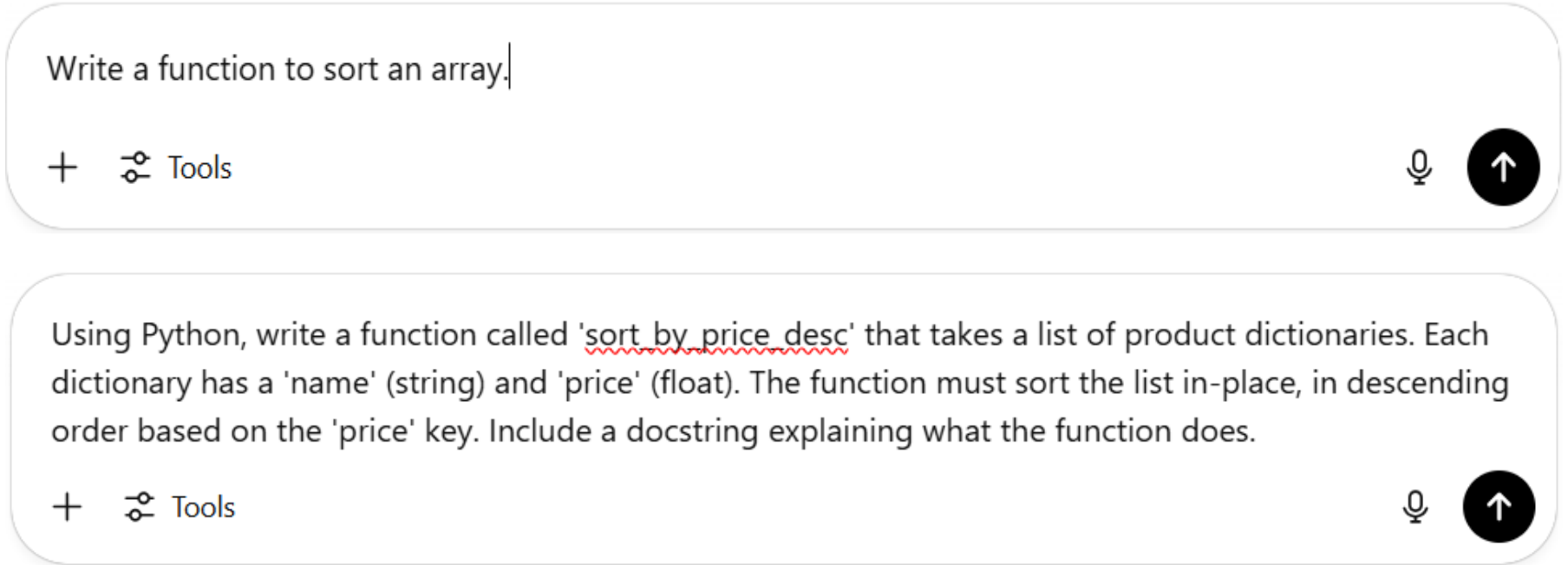
- You don't have to be a data scientist, but you need to speak the language
- Model, training, inference, APIs

- Not all models are created equal

- Text generation (Gemini, GPT-4)
- Image generation (Imagen, Midjourney, DALL-E)
- Embeddings models (text-embedding-ada-002)

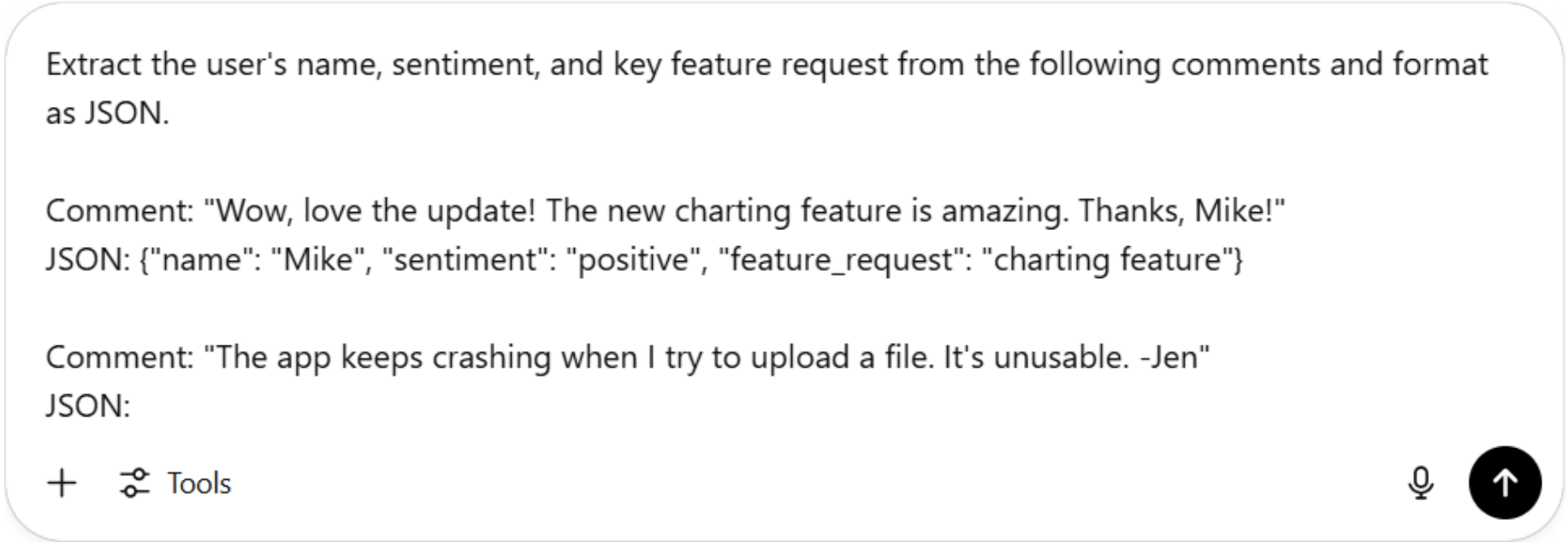
Prompt engineering

- Search engine ≠ Large Language Model (LLM)
- The quality of the input is key for the quality of the output



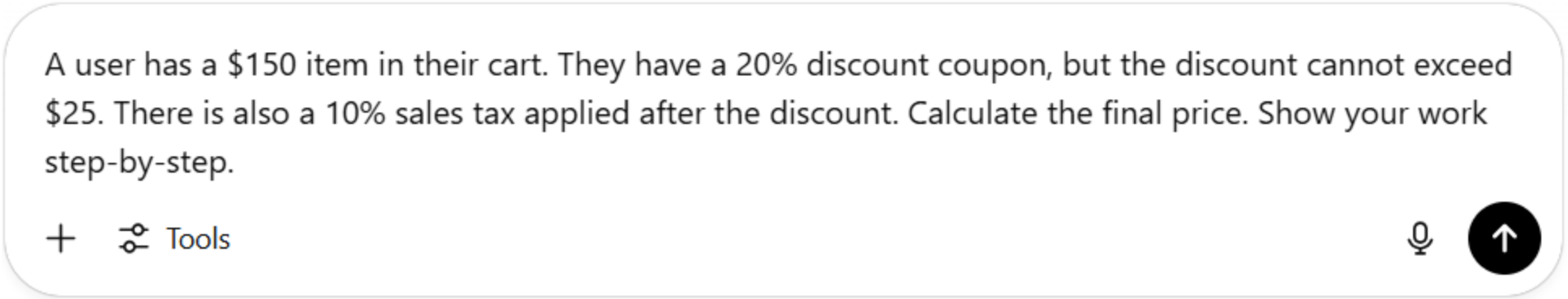
Prompt engineering (2)

- Few-shot prompting



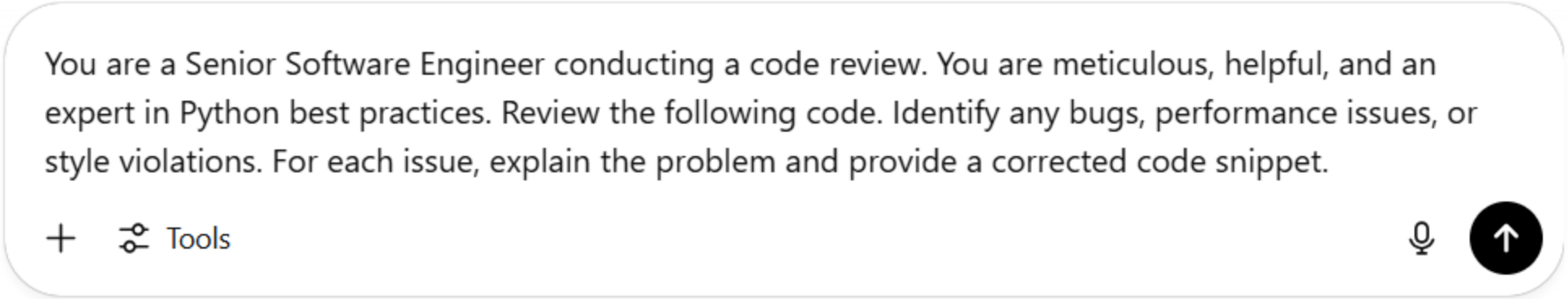
Prompt engineering (3)

- Chain-of-thought prompting



Prompt engineering (4)

- Persona setting



Full-stack development

- From frontend pixels to backend logic, full-stack developers bring ideas to life

- But why full-stack?
  - Faster prototyping – AI may help, but you still need to understand the intricacies
  - Independence – going from idea to MVP on your own
  - AI feature integration – integrate APIs, stream responses, manage state, secure data
  - Empowerment – you own the UX, from design to functionality to deployment

System design with AI components

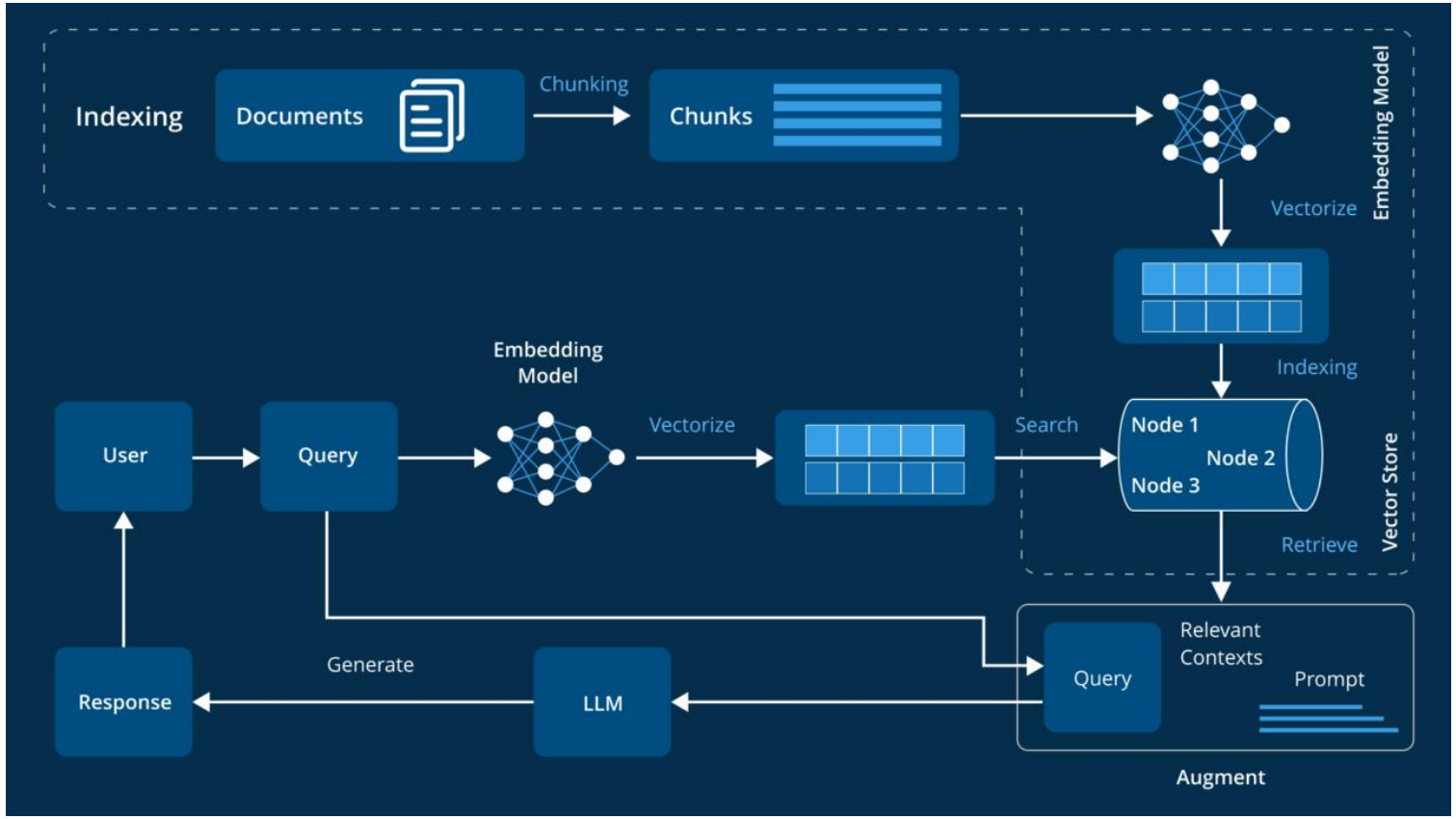
- Adding an AI feature ≠ importing a new library
- An AI component is different

- Latency – LLM API calls are slow
- Cost – every API call costs money
- Scalability

- AI as a microservice

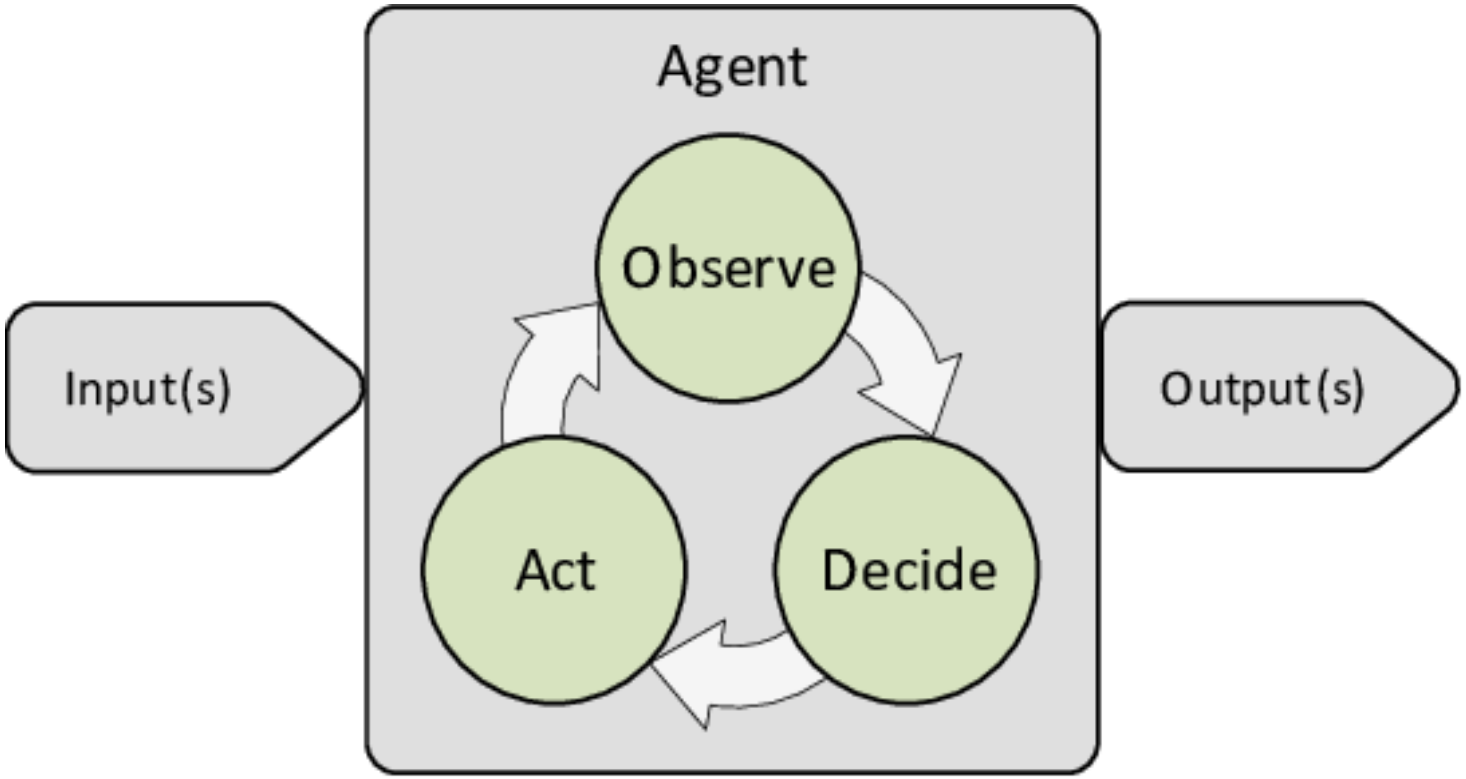
System design with AI components (2)

- Retrieval-augmented generation (RAG)



System design with AI components (3)

- Agentic workflows



Domain-specific knowledge

- AI is a powerful tool, but worthless without a problem
- True innovation ≠ try to apply AI to everything

- 'Can we use AI to improve our e-commerce site?' wrong question
- 'Can we use an embeddings model to create a semantic search for our product catalog, so when a user searches for 'beach party outfit,' it shows them sunglasses, shorts, and bright shirts instead of just products with the literal word 'beach' in the name?' right question

- How to build domain knowledge

Collaboration with AI tools

- Writing code
  - // Function that takes a user ID, fetches their profile and recent orders from two different APIs, and merges the results into a single JSON object
  - Learning on the fly – using a new library
- Debugging code
  - A more powerful rubber duck
  - 'What's wrong here?' or 'Explain this code to me step-by-step.'
- Refactoring code
  - 'Refactor this into smaller, single-responsibility functions.'

Empathy for building good UI&UX

- The AI does not know
  - The anxiety a user feels when a file upload seems stuck
  - The user might be colorblind

- ~~What?~~, **How?**, **Why?**

- Think like the user, not the coder
- Handle errors gracefully, plan for them
- Advocate for the user

Fact-checking and critical thinking

- AI is a brilliant, incredibly fast, and extremely confident intern
- Be the skeptical senior developer

- The problem of 'hallucinations'
  - LLMs are designed to generate plausible text, not to state the truth

- Trust, but verify
- Write tests for AI-generated code
- Never copy-paste blindly

Communication

- Problem engineering before prompt engineering
- Explain the 'Why' behind the 'What'
- Manage expectations and explain limitations

- How to level up your communication
  - Practice explaining technical concepts simply
  - Write more
  - Embrace presentations
  - Become an active listener

Lifelong learning and adaptability

- A degree is not a certificate of completion
- It is your entry ticket to the race

- Block out 2 – 3 hours every single week to read, watch, or build
- Make a learning stack – newsletters, key researchers and engineers
- Learn by building

- Your biggest asset

Vibe coding

- A casual, intuitive, and creative approach to programming, often focused more on feel and flow than strict structure or planning.

- Coding without a rigid plan – not following a detailed spec
- Creative exploration – experimenting with new technologies or ideas
- High-flow state – not worried about code quality, edge cases, etc.
- Emphasis on aesthetics or fun
- Minimal pressure – no deadlines, no code reviews

Vibe coding (2)

- The erosion of *understanding*
- Writing, debugging, and refactoring code – cognitive training
- Outsourcing the thinking to AI leads to losing the context of our code

- No shortcuts to mastery
  - Practice building ideas from scratch
  - Experiment, write code, break things, learn to debug

- Responsibility and trust – who is accountable for the written code?

What is Cursor AI?

- A fork of Visual Studio Code, that integrates powerful AI features

- Model selection: GPT-4, Claude, Gemini
- API-driven communication – including relevant context from codebase
- Prompt engineering

- Highly context-aware assistance through RAG

Building a tic tac toe app using Cursor AI

Key takeaways

1. AI is your new pair programmer, not your replacement
2. You are the senior developer, AI is the brilliant, but unreliable intern
3. The skill of learning is more important than any skill you've learned

THANK YOU!

Any questions?

Radu Capalb
radu_capalb@sync.ro