Lecture 12: Running Model Locally with Ollama

Why Run Locally?

- Until now, we used **OpenAI as a service** → requests went to their server.
- Needed an account and credits (paid).
- But you may want to run models **on your own laptop/desktop** for free or customisation.

Ollama – The Local Runner

- Ollama lets you run Large Language Models (LLMs) on your machine.
- Supports many open-source models: Llama, DeepSeek, Phi, Mistral, Gemma, and more.
- Simple setup:
    - Download (~1 GB).
    - Install (Next → Next → Finish → Agree).
    - Open terminal → type 'ollama' to verify.

Working with Models

- Check installed models → ollama list.
- By default, no models are included (each must be downloaded separately).
- Example:
    - **Cmd**: ollama run mistral
- If the model exists, it runs immediately.
- If not, it downloads and installs automatically.

Example Usage

- ollama run mistral → starts the Mistral model.
- Ask: "What is Spring AI?" → may give vague/old answers (since local models are trained on older data).
- Ask: "Tell me a joke." → responds with programming jokes.
- Follow-up "more" → conversation continues (local memory works).

Things to Know

- **Local models are heavy**:
    - Mistral 7B (~4 GB) → manageable.
    - DeepSeek 32B (~20 GB) → needs high RAM + GPU.
- Start with smaller models.
- Running bigger ones may strain your system.

Integration with Code

- Cloud models (like OpenAI) require API keys.
- **Local Ollama models don't need a key** → they run on your system.
- To connect with your Spring AI project, you'll create a **separate controller** for Ollama integration.
- Current controller is OpenAI-specific.

Key Takeaways

- Ollama = Local LLM runner (free, customizable).
- Supports multiple models, downloaded on demand.
- Local models may be slower and need strong hardware.
- Spring AI will require a new controller setup for Ollama integration.