



# NTT DATA Türkiye

We Transform, the Future into the Present

# AI Engineer Case Study

Hello, and thank you for your interest!

This task represents the **technical evaluation phase** of your application to join our **AI team**.

Unless specific technologies are explicitly mentioned in the case requirements, you are free to choose the tools, libraries, and frameworks you deem appropriate. We encourage you to incorporate your own creativity and engineering perspective into the solution.

# Case Requirements

## 1. Data Collection

- Locate and download publicly available **Sustainability Reports** from **NTT DATA Business Solutions**. Here is the [link](#) of the reports.
- Include at least **five reports from different years**.
- Store them in the project repository with proper version control and naming conventions.

## 2. Preprocessing & Embedding

- Extract and chunk the text contents of the PDFs.
- Clean and preprocess the text.
- Embed the chunks using a suitable **embedding model**.

# Case Requirements

## 3. RAG Pipeline

- Build a Retrieval-Augmented Generation (RAG) pipeline combining a vector database and a large language model (LLM).
- For a given input question, retrieve relevant context and generate an answer.

## 4. API Layer

- Develop a FastAPI-based RESTful service with the following endpoints:
  - POST /ask → Accepts {question} and returns {answer, sources}.
  - GET /health → Returns service health status.

# Case Requirements

## 5. Containerization

- Package the solution in a **single Dockerfile**.
- If required, use **docker-compose** to orchestrate related services and dependencies.

## 6. Testing & Documentation

- Implement unit tests using **Pytest**.
- Provide a clear and complete **README** that includes:
  - Installation & usage instructions
  - An **architectural diagram** of your solution

# Additional Skills Considered **Nice-to-have**

**LLM implementation practices:** prompt engineering, advanced RAG concepts, agentic framework

**MLOps best practices:** CI/CD, model & embedding versioning, monitoring

**Clean code & software architecture:** type annotations, modularity, async programming, OOP

**Deployment experience:** Architectures in cloud environments

**Technical communication:** ability to explain complex concepts simply, and foster a code review culture

# Evaluation Criteria

Correctness & use of referenced sources	%30
Code quality (cleanliness, typing, testing)	%20
Architectural decisions & justification	%15
API design and ergonomics	%15
Docker/DevOps proficiency	%10
Documentation & presentation clarity	%10

# Submission

- Share all code in a version-controlled GitHub repository.
- Prepare a demo and a presentation no longer than 30 minutes.

We wish you the best of luck in your assignment and look forward to your submission 😊🚀!



