[Challenge Title]

Build and Evaluate the Next-Generation of RAG System for AEC Knowledge

[Problem Statement]

AEC/O companies work with extremely large and complex technical documents, BIM standards, construction manuals, regulations, specifications, safety documents, and product documentation. These documents are often hundreds of pages long, written in dense technical language, and updated frequently.

Professionals who need fast answers — architects, engineers, planners, and site managers — waste significant time searching for the right information manually. As AI becomes increasingly adopted, Retrieval Augmented Generation (RAG) is emerging as the leading approach to enable trustworthy, context-grounded answers based on internal documents.

However:

- RAG systems behave very differently depending on retrieval strategy (TF-IDF, semantic vector search, hybrid)

- Evaluating RAG quality is difficult and requires specialized metrics for hallucinations, relevancy, and recall

- Many teams lack tools and datasets to properly measure how well their RAG system performs

Your challenge: build a RAG prototype and reliably measure its performance — all within 48 hours.

This problem matters because trustworthy AI assistance could reduce costly mistakes, speed up decision-making, and help every AEC/O professional interact with technical content more effectively.

[The Goal]

Design, implement, and evaluate an innovative RAG prototype using the provided AEC/O document corpus.

Teams should:

1. Create or modify a RAG model
    - Choose your retrieval strategy (TF-IDF, FAISS semantic search, etc.)
    - Use any LangChain-compatible LLM (OpenAI, Vertex AI, local models)

2. Generate or use the provided Q&A dataset
    - Use synthetic ground-truth dataset, or create your own improved evaluation samples

3. Run automated evaluation
    - Use integrated DeepEval “LLM-as-judge” pipeline
    - Measure:
        - Answer Relevancy
        - Faithfulness
        - Contextual Recall
        - Contextual Precision

4. Improve RAG system
    - Tune chunking, embeddings, prompt templates, retrieval settings, or hybrid retrieval
    - Compare performance across strategies
    - Present insights and visualized results

[Expected hackathon deliverable]

- A working RAG prototype
- Evaluation dataset (ground truth + answers)
- Evaluation results
- A short pitch explaining:
    - What you changed
    - Why your RAG is better
    - What future improvements would unlock

This requires creativity, experimentation, and applying AI tools to real AEC/O documents.

[Resources and Support]

All materials, instructions, datasets, and technical guidance are available in challenge GitHub repository:

👉 **GitHub:** https://github.com/Nem-AI-RnD/tum-hackathon/tree/main

Participants will find everything they need there, including:

- AEC document(s)
- Starter framework
- Examples and instructions

Mentors from Nemetschek will be available during the hackathon for technical questions and support.