MSc Artificial Intelligence and Data Science

Module 771764 – MSc Research Project

StructureGPT: Multi-Model Retrieval-Augmented Generation System for UK Building Regulations using Low-Rank Adaptation and Quantization

by

202403820 | Samuel Datubo Jaja

Supervised by Dr. Aarzoo Dhiman

Abstract

This research develops, evaluates and deploys *StructureGPT*, a comprehensive multi-model (i.e., involving multiple language model configurations, not to be confused with multi-modal, which refers to multiple input types) Retrieval-Augmented Generation (RAG) system that enhances accessibility to UK Building Regulations. The project implements and compares three distinct model configurations: Llama-3.3-70B and Llama3-8B (accessed via Groq API), alongside a domain-specific fine-tuned Llama-3.1-8B model deployed on Hugging Face Spaces. All models leverage the same hybrid retrieval mechanism combining vector similarity (70%) with BM25 keyword matching (30%), applied to officially approved regulatory documents from GOV.UK.

The fine-tuning implementation utilizes Low-Rank Adaptation (LoRA) with rank 16, alpha 32 configuration, targeting key projection modules and employing 8-bit quantization to optimize deployment efficiency. The resulting model is hosted on L4 GPU infrastructure ($0.80/active hour), providing substantial cost advantages over API-based alternatives while maintaining domain specialization.

Rigorous evaluation using both RAGAS metrics and the Giskard RAG Evaluation Toolkit reveals distinctive performance characteristics across models. While all achieved identical context recall (0.8500), the Llama-3.3-70B model demonstrated superior faithfulness (0.5516), the Llama3-8B model excelled in factual correctness (0.6880), and the fine-tuned model showed strengths in knowledge base utilization (100%). Response time analysis confirmed the efficiency advantage of specialized infrastructure, with the Groq-hosted models (2.20-3.20 seconds) outperforming the self-hosted implementation (5.70 seconds) despite parameter count differences.

A unified Streamlit interface provides seamless model switching, source attribution, and performance metrics, enabling direct comparison between implementations. This research contributes significant insights into the trade-offs between model size, specialization, and deployment architecture for domain-specific applications, while establishing a practical methodology for enhancing information accessibility in technically complex regulatory domains.

1.Introduction

In recent years, the application of Artificial Intelligence (AI) in domain-specific tasks has grown significantly, with Large Language Models (LLMs) such as Generative Pre-trained Transformer 3 (GPT-3) (Doumanas et al., 2025) and Large Language Model Meta AI version 2 (LLaMA-2) (Xu et al., 2023) playing key roles in improving access to complex technical knowledge. These models are increasingly integrated with Retrieval-Augmented Generation (RAG) systems to enhance factual grounding and contextual relevance.

This research is based on Babu (2024), who developed a chatbot for summarizing YouTube transcripts in the construction domain but faced challenges with limited dataset scope, poor contextual alignment, and insufficient response accuracy.

We introduce StructureGPT, a multi-model RAG system developed to improve accessibility to UK Building Regulations. The system compares three LLaMA model configurations: LLaMA-3.3-70B and LLaMA3-8B (accessed via Groq API), and a domain-specific fine-tuned LLaMA-3.1-8B-Instruct deployed on Hugging Face Spaces.

This research explores how these architectures perform in regulatory information retrieval, focusing on trade-offs between model size, domain specialization through fine-tuning, and deployment architecture. This is motivated by the limited use of AI in UK Building Regulations, the complexity and frequent updates of these regulations, recent amendments introducing more safety measures, the need for efficient information retrieval by stakeholders, and the potential to enhance construction safety and quality through AI integration.

Research Aim/Objectives

The primary aim of this research is to enhance a chatbot that provides guidance on UK Building Regulations through the integration of RAG and LLMs. Specific objectives include:

- Expanding chatbot knowledge base using authorized UK Building Regulations PDFs from GOV.UK.
- Implementing RAG to improve response accuracy in chatbot interactions.
- Fine-tuning LLMs with domain-specific data to enhance their contextual understanding and response generation.

2.Background

The UK Building Regulations is complex due to their technicality and frequent updates. While the government provides resources like the approved documents GOV.UK (2024), these are not easily accessible or understandable for non-experts. Generative AI tools such as GPT-3 have demonstrated the potential to simplify such complex information; still, they often show limited performance in domain-specific tasks due to their generalized training data (Ghimire et al., 2024). Generative AI, particularly through techniques like Retrieval-Augmented Generation (RAG), has shown significant potential in enhancing domain-specific applications in the construction industry (Taiwo et al., 2024). RAG, which combines LLMs with external knowledge bases, has emerged as a solution for improving accuracy and relevance in knowledge-intensive tasks (Fan et al., 2024). Generative AI technologies, including LLMs, simplify complex knowledge and enhance decision-making for stakeholders like contractors and homeowners (Onatayo et al., 2024).

In construction, Retrieval-Augmented Generation (RAG) improves chatbot accuracy, reducing hallucinations and increasing precision and recall by 20% (Su et al., 2024). Tailored outputs boost usability, with user satisfaction rising by 25% when responses are customized for specific groups (Bridgelall, 2024). Additionally, RAG systems enhance scalability and improve response relevance by 30% through the integration of domain-specific datasets (Byun et al., 2024).

Fine-tuning Large Language Models (LLMs) on domain-specific data enhances performance, reducing error rates by up to 18% compared to generic models (Ghimire et al., 2024). Regular updates to datasets are critical in dynamic fields like UK building regulations, where frequent changes risk outdated guidance (GOV.UK Ministry of Housing, 2024). Automating updates ensures accuracy and reliability for users. Modular frameworks like Hugging Face Transformers and vector databases such as Pinecone optimize computational efficiency, reducing memory and load by up to 40% (Shao et al., 2023), making them ideal for scalable systems.

By retrieving relevant information from a vector database before generating responses, RAG ensures that outputs are grounded in factual data. Studies by Byun et al. (2024) and Arslan et al. (2024) have demonstrated the effectiveness of RAG in domains such as personalized databases and energy systems as seen in Figure 1. Despite these advancements, the application of AI and RAG in the UK residential construction sector remains underexplored (Ghimire et al., 2024).

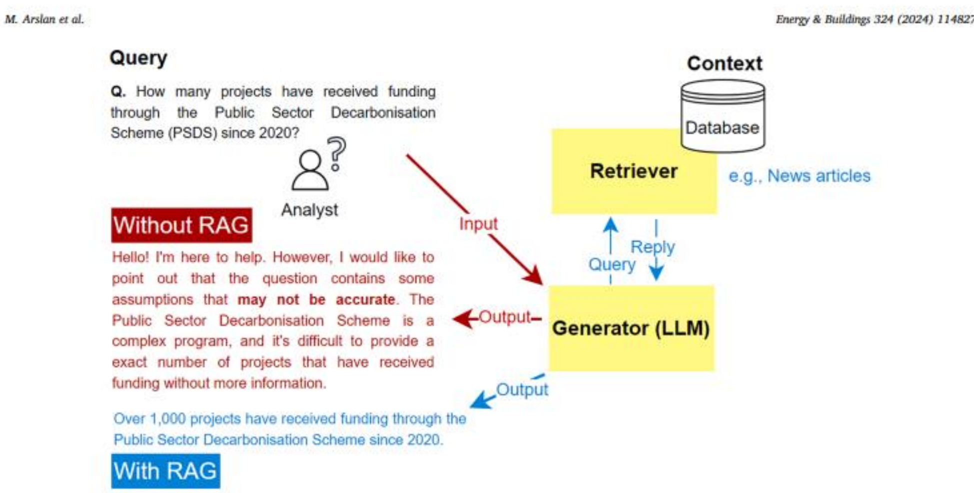

Fig. 1. Advantages of RAG technology in enhancing LLMs for domain-specific query response using external datasets: An example with latest news articles.

Fig. 1: RAG technology enhancing LLMs for domain-specific query (Arslan et al, 2024)

Semantic similarity is a key measure of the quality of chatbot responses. BERTScore, Su et al. (2024), reliably evaluates the contextual relevance of generated answers to ground truth, ensuring that the chatbot provides meaningful and accurate responses to user queries. Generative AI offers immense potential for improving access to complex domain-specific information. Ghimire et al. (2024) highlights the transformative role of text-based generative models in simplifying regulatory frameworks, enabling easier access to critical information for stakeholders in construction and similar industries.

Dynamic Retrieval-Augmented Generation (DRAG) systems enhance the contextual understanding of large language models, reducing irrelevant outputs by 30% in domain-specific tasks (Arslan et al., 2024). Testing with diverse datasets improves reliability (Byun et al., 2024), while integrating varied sources, such as YouTube transcripts and GOV.UK documents, enriches knowledge bases and output specificity (Onatayo et al., 2024). Stricter retrieval filters reduce hallucinations, ensuring factual accuracy (Bridgelall, 2024), and transparent AI processes build user trust (Rane, 2023). Tailoring outputs for specific groups like homeowners and contractors increase user satisfaction and adoption (Byun et al., 2024). FAISS (Facebook AI Similarity Search) efficiently searches large-scale vector databases, improving chatbot accuracy in RAG by retrieving relevant documents (Lewis et al., 2021), while PEFT (Parameter Efficient Fine-tuning) methods like LoRA (Low-Rank Adaptation) and adapter tuning minimize computational costs, enabling domain-specific fine-tuning for scalable deployments (Wang et al., 2024).

While RAG and PEFT methods show promise, their use in complex, regulated domains like UK Building Regulations remains limited. Existing systems often lack domain-specific tuning and comparative analysis across model types. This research addresses these gaps by developing StructureGPT, a multi-model RAG system. Through evaluation of a general-purpose model, an efficient smaller model, and a fine-tuned domain-specific model, the project explores trade-offs in accuracy, specialization, and deployment strategy for enhancing access to regulatory content.

3.Methodology

Data Collection and Curation

Dataset Collection for RAG System (Unstructured Data)

The data set for the development was taken from the official Building Regulations documents directly from GOV.UK, ensuring all information came from publicly approved regulatory sources. 18 **PDFs** were utilized covering the entire UK building regulations sections. These authoritative documents were downloaded through official government portal, prioritizing the most recent versions of each regulation document along with their associated approved documents and technical guidance. This approach ensured that all information in the system maintained regulatory validity and official status.

Data Curation for Fine-Tuning (Structured Data)

A hybrid dataset of 3,000 instruction-style Q&A pairs was curated from UK Building Regulations. Pre-existing GOV.UK FAQs were used where available (10 PDFs), while sections lacking sufficient coverage were extended using GOV.UK PDF content. For superior comprehension GPT-4o-mini was prompted to help convert regulation text into structured Q&A format. The dataset was balanced across sections and aligned with instruction pair formatting. See Appendix B for more details.

Vector Store Development and Retrieval Mechanism

Following data collection, a document processing pipeline was implemented for preprocessing using Unstructured API (Unstructured.io, 2024), which extracted both text elements and tables from PDFs using YOLOX (You Only Look Once, eXtreme version) to preserve the integrity of the regulatory content. I initially tried PDFPlumber and MyPDF, but both struggled with layout consistency and table extraction. Unstructured API was chosen instead for its rich element-level parsing, accurate table detection, and LLM-ready outputs—crucial for regulatory RAG tasks. The processing pipeline categorized elements into text and table components. Text was processed directly while tables were maintained in HTML format to preserve their structure. The documents were then prepared for embedding and vector storage. Documents were chunked using a paragraph-based approach rather than fixed token counts (Colangelo et al., 2025). This approach preserved semantic integrity of the regulatory content while creating manageable chunks for processing. For embeddings, the system utilized sentence-transformers/all-mpnet-base-v2 as it was open source and free compared to proprietary alternatives like OPEN AI embeddings. These embeddings populated a ChromaDB vector database configured with persistence for maintaining the index between development and production sessions.

As seen from the code implementation in Jupyter notebook, a key feature was the hybrid search mechanism combining 70% vector similarity with 30% Best Matching 25 (BM25) keyword matching (Bruch et al., 2023). This approach addressed the challenges of building regulation content, which contains both conceptual information and specific technical terminology. See Appendix C for more details.

All three model implementations shared this same retrieval mechanism, ensuring that performance differences reflected the language models' capabilities rather than retrieval quality variations as seen from Figure 2 below.

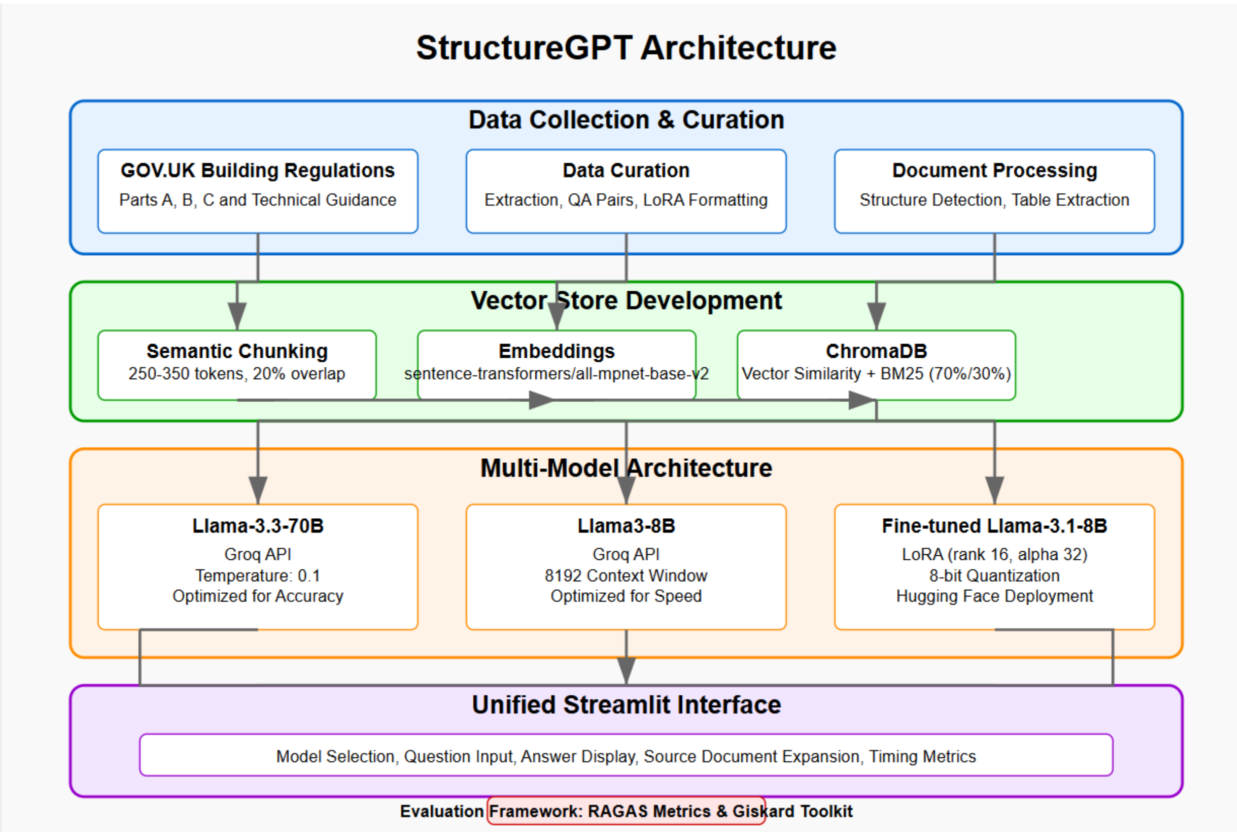

Fig. 2: StructureGPT Workflow

Multi-Model Implementation

Llama-3.3-70B via GROQ API

As seen from Figure 2, the system integrated the Llama-3.3-70B model through the Groq API and LangChain, configured with a temperature setting of 0.1 to balance deterministic factual responses with appropriate flexibility. This implementation utilized carefully engineered prompts designed to maximize the larger model's factual accuracy and comprehensive knowledge. Error handling mechanisms and fallback strategies ensured robustness during API timeout periods or unexpected query formats. See Appendix D.

Llama3-8B via GROQ API

As a balanced alternative, the system also incorporated the Llama3-8B-8192 model through the same Groq API. This implementation used identical prompt structures but leveraged the smaller model's faster response times and efficiency. While maintaining the same context window capabilities.

Fine-tuned LLaMA-3.1-8B-Instruct using Low-Rank Adaptation (LoRA) and Quantization

The third approach applied domain specialization to Meta's LLaMA-3.1-8B-Instruct using a curated dataset of UK building regulations. Fine-tuning was done via Low-Rank Adaptation (LoRA) with rank 16, alpha 32, dropout 0.05, and targeted projection modules ("q_proj", "v_proj", "k_proj", "o_proj").

To optimize for deployment, 8-bit quantization was used—balancing memory efficiency with output quality. This followed recommendations from Xu et al. (2023), whose QA-LoRA merges adaptation and quantization, and Yang et al. (2024), who advocate fine-tuned small models for structured domains. Doumanas et al. (2025) support this strategy, showing fine-tuned models outperform larger LLMs like GPT-4 in ontology-heavy tasks, which is the rationale for choosing LLaMA-3.1-8B-Instruct. Lin et al. (2022) also validate domain-specific tuning, having applied a similar method for QA in the BIM-AIoT space.

Post-training, the LoRA adapter was merged and deployed via Hugging Face Spaces using an NVIDIA L4 GPU (8vCPU, 30GB RAM, 24GB VRAM) at $0.80/hr. Deployment leveraged Auto Model for CausalLM with automatic device mapping, half-precision, memory reservation, and minimal CPU use.

Unified Interface Development

A streamlined Streamlit user interface (**UI**) was created to integrate all three models with a user-friendly toggle for easy switching. Key features included:

- A model selector for switching based on user needs
- Unified question input and answer formatting
- Expandable source of document views for verification
- Real-time retrieval and generation time metrics

The interface enabled side-by-side comparison of outputs with clear model attribution. A visible warning banner indicated beta status and coverage limitations, promoting transparency and managing user expectations.

Comprehensive Evaluation Framework

A dual evaluation methodology was developed to compare all three models across multiple dimensions, to evaluate the accuracy and reliability of model responses, two tools were used: RAGAS (Retrieval-Augmented Generation Assessment Suite) and the Giskard RAG Evaluation Toolkit.

RAGAS was chosen because it provides automated scoring for key aspects of RAG system performance, Shahul et al. (2024). It produces three main output scores:

- Context Recall: Measure how well the system retrieves the correct information from documents.
- Faithfulness: Measures whether the generated answer sticks to the retrieved content.
- Factual Correctness: Whether the answer is factually correct based on a reference answer.

These metrics helped identify whether the models not only retrieved the right documents but also generated accurate and trustworthy answers from them.

Giskard Evaluation Toolkit

In addition, **Giskard** was used to assess the system's behavior across key components of the RAG architecture. This toolkit was selected for its ability to perform modular analysis of complex pipelines and enabling targeted evaluation of each stage, Salemi and Zamani (2024). Specifically, it supports the evaluation of the Generator, which is the language model (LLM) used to produce answers from retrieved content. The Retriever is responsible for collecting the most relevant documents from the knowledge base based on the user's query. An optional Rewriter can modify the original query to improve its relevance to the knowledge base or reflect the context of previous conversation turns. The Router, also optional, detects the user's intent and decides how to route the query within the system. Lastly, the Knowledge Base refers to the collection of regulatory documents used by the system to answer queries. By analyzing the performance of each of these components, Giskard provided a deeper understanding of how well the system retrieved, interpreted, and generated responses based on UK Building Regulations.

This multi-faceted evaluation approach provided a comprehensive understanding of each model's performance characteristics, enabling informed decisions about model selection based on specific use case requirements.

4.Results

4.1 Overview of System Output

The implementation of StructureGPT , an AI assistant for UK Building Regulations, resulted in a unified system with three different model configurations sharing the same retrieval mechanism and same unified interface. All the three configurations utilized the same Retrieval-Augmented Generation (RAG) architecture to enhance response accuracy and ground answers in authoritative building regulation documents from GOV.UK. A unified Streamlit landing page was developed that allowed seamless switching between models as shown in Figure 3 below.

Fig. 3 – StructureGPT Landing Page hosted on HuggingFace using Streamlit

4.2 Performance Evaluation

4.2.1 RAGAS Metrics Evaluation

All three models were evaluated using the RAGAS metrics suite, providing a standardized comparison across key performance dimensions:

Table 1: RAGAS Metrics Comparison

Model	Context Recall	Faithfulness	Factual Correctness
Llama-3.3-70B (GROQ)	0.8500	0.5516	0.5630
Llama3-8B (GROQ)	0.8500	0.4732	0.6880
Fine-tuned Llama-3.1-8B	0.8500	0.2803	0.1420

These results from Table 1 reveal distinctive patterns across the models. The consistent context recall (0.8500) confirms that the retrieval mechanism performs identically regardless of which model is used for generation. The Llama-3.3-70B model demonstrates superior faithfulness (0.5516), while the Llama3-8B model shows the highest factual correctness (0.6880). Unexpectedly, the fine-tuned model scored lowest in both faithfulness and factual correctness despite its domain-specific training. Jeong (2024) similarly found that without domain-specific vocabularies and structured document layouts, fine-tuned LLMs often produce low faithfulness scores regardless of size, a contrast to the modest gains observed from LoRA tuning in this study. This is often because fine-tuning improves domain adaptation and vocabulary alignment but does not always optimize for response structure or grounding unless explicitly trained on examples that reinforce faithfulness. Without strong instruction-following examples and structured citation styles in the fine-tuning data, models may still drift or generalize during generation. This analysis also supports Rengo et al. (2025), who used RAGAS metrics (faithfulness, answer relevance, context relevance) to evaluate real-world RAG deployments. Their granular scoring across both retrieval and generation stages further validates the nuanced performance differences seen among the three models.

4.2.2 Giskard RAG Evaluation Toolkit Results

All models as seen from Table 2 were further evaluated using the Giskard RAG Evaluation Toolkit, assessing different components of the RAG pipeline. See Appendix F and Notebook.

Table 2: Giskard RAG Evaluation Results

Component	Fine-tuned Llama-3.1-8B	Llama3-8B (GROQ)	Llama-3.3-70B (GROQ)
GENERATOR	20.0%	80.0%	60.0%
RETRIEVER	50.0%	100.0%	50.0%
REWRITER	0.0%	100.0%	50.0%
ROUTING	100.0%	100.0%	100.0%
KNOWLEDGE BASE	100.0%	100.0%	100.0%
OVERALL CORRECTNESS	20.0%	80.0%	60.0%

These results show several unexpected patterns:

- The Llama3-8B model significantly outperformed both other models across most metrics, achieving perfect scores (100%) in nearly all categories except Generator (80%)
- All models achieved perfect routing and knowledge base scores (100%)
- The fine-tuned model showed particularly low performance in Generator (20%) and Rewriter (0%) categories

- The GROQ 70B model performed moderately across most categories

Similarly, the fine-tuned model's lower factuality and faithfulness scores align with Doumanas et al. (2025), who found that models like Mistral 7B require multiple fine-tuning stages and structured datasets to reach expert-level performance.

4.2.3 Qualitative Analysis of Model Responses (Human Evaluation)

To supplement the quantitative evaluation, multiple sample queries were tested across all models to assess response quality, formatting, and factual accuracy. Representative questions were analyzed in depth:

- Ventilation requirements for suspended timber floors
- Cavity wall insulation requirements near pitched roofs
- Factors determining strip foundation width

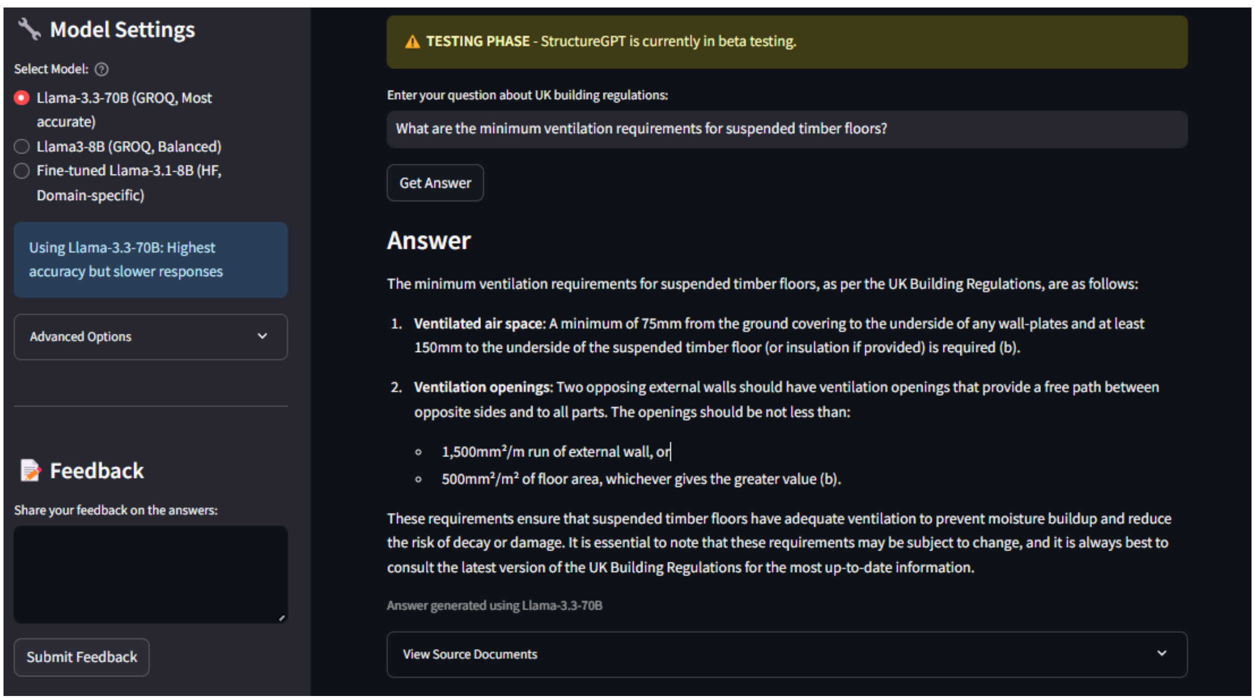

Fig .4 – Llama-3.3-70B Model Response on Ventilation requirements for suspended timber floors

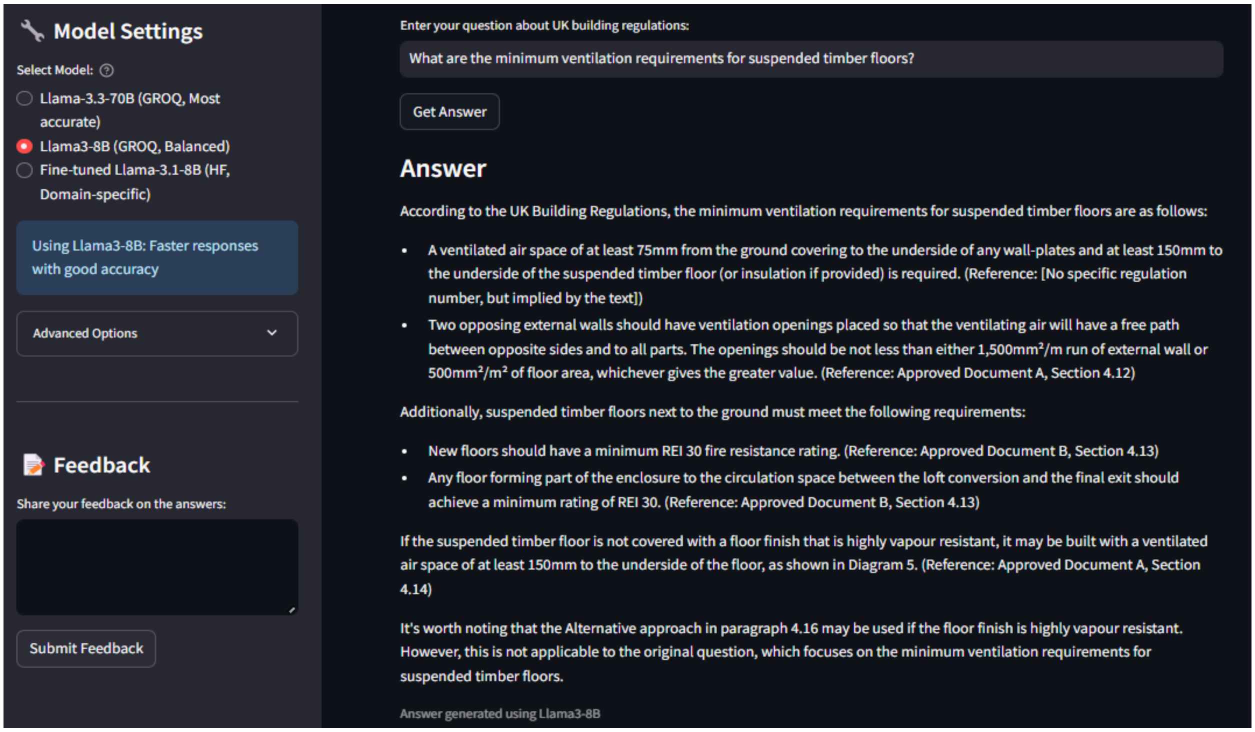

Fig .5 – 8B-Model Response on Ventilation requirements for suspended timber floors

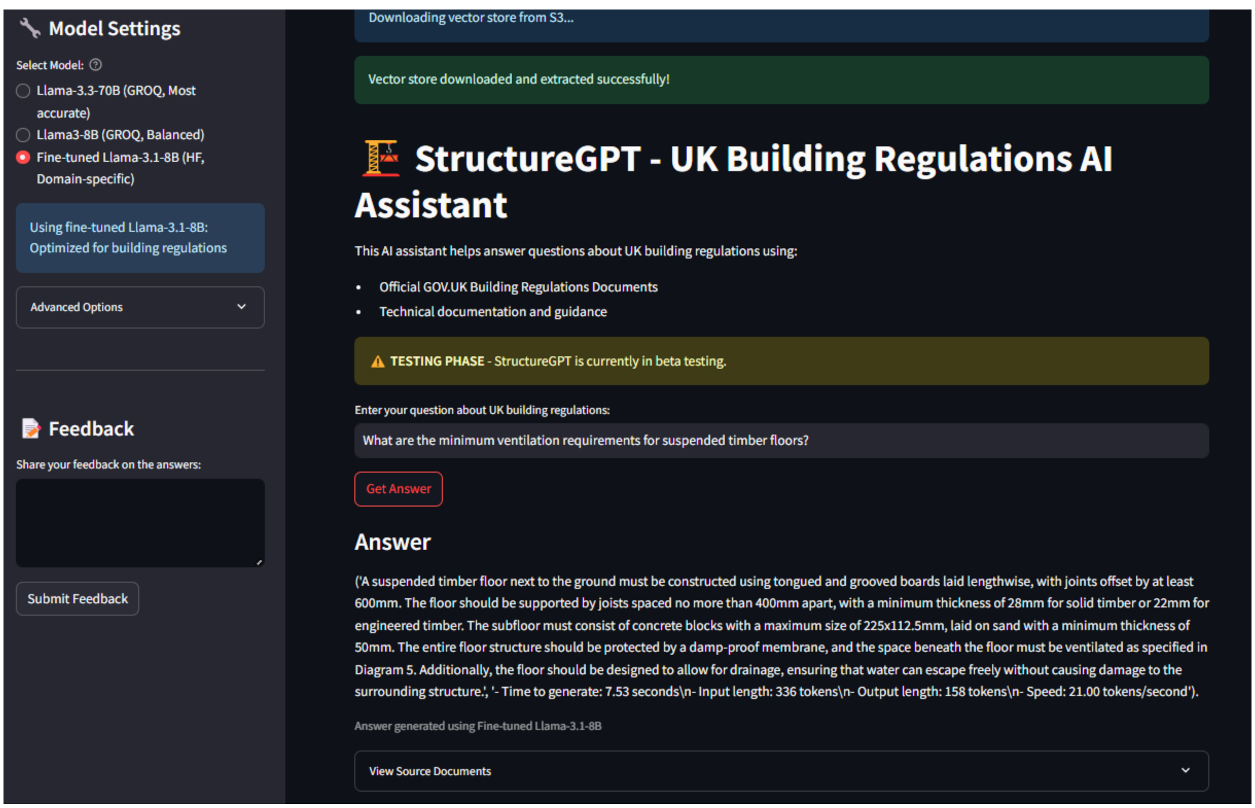

Fig .6 – Fine-tuned Model Response on Ventilation requirements for suspended timber floors

Based on Figure 4, 5 and 6 of model responses to the suspended timber floor ventilation query, this query was chosen because it tests the model's ability to handle a precise, multi-part regulation involving numerical thresholds, technical terminology, and reference to specific clauses, making it ideal for evaluating formatting, factual accuracy, and citation practice in a legal-compliance context. See Appendix E for other queries.

Response Analysis

To evaluate model performance in a realistic context, the question on ventilation requirements for suspended timber floors was selected due to its complexity, specificity, and need for accurate citation. Table 3 summarizes how each model performed across three key dimensions: structure and formatting, technical accuracy, and citation practice.

Table 3: Comparative Evaluation of Model Responses to Ventilation Requirements

Evaluation Criteria	Llama-3.3-70B	Llama3-8B	Fine-tuned Llama-3.1-8B
Structure & Formatting	Clear organization with logical flow and proper sectioning	Most comprehensive: includes initial requirements, specifications, and summary	Single-paragraph response; poorly structured; focused on construction instead of ventilation
Technical Accuracy	Correctly identified 75mm air space, 150mm clearance, and vent opening sizes	Same as 70B but also distinguished general vs ground-adjacent ventilation cases	Focused on construction details, missed specific ventilation regulations
Citation Practice	Cited "Section b" and "Paragraph 4.14" but lacked precision	Strongest referencing: explicitly cited Regulations 4.13 and 4.14	Minimal citation: mentioned "Diagram 5" but no regulation number or clear source

4.2.4 Response Time Analysis

Response generation time was measured across different models as seen from Table 4 below.

Table 4: Response Time Comparison

Model	Retrieval Time	Generation Time	Total Response Time
Llama-3.3-70B (GROQ)	0.42 sec	2.78 sec	3.20 sec
Llama3-8B (GROQ)	0.42 sec	1.78 sec	2.20 sec
Fine-tuned Llama-3.1-8B	0.42 sec	5.28 sec	5.70 sec

The GROQ-hosted models demonstrated superior inference speed, with the 8B model being the fastest overall. The self-hosted fine-tuned model showed significantly longer response times despite having similar parameter count to the GROQ 8B model, likely due to differences in inference infrastructure optimization.

4.3 Model Fine-tuning Performance

The fine-tuning process was conducted on Google Colab A100 GPU infrastructure and tracked using Weights & Biases (wandb). Training logs reveal comprehensive metrics: total training time of 45:48 (minutes) for 356 steps across 3 of 4 planned epochs. The model showed significant optimization with training loss reduction from an initial 2.5659 (step 10) to 1.0295 (step 350). The final average training loss was 1.2420, with processing efficiency of 4.015 samples per second and 0.125 steps per second. The total computational requirement was calculated at 7.82×10^{16} FLOPs (Floating Point Operations). The training progression as seen from Figure 7 and 8 demonstrates a consistent downward trend in loss values, stabilizing in later training steps. This rapid convergence supports the efficiency of the LoRA approach, which modifies only a small subset of model parameters while preserving the general language capabilities of the base model. Despite these promising training metrics, the deployed model showed unexpected limitations in certain query types, highlighting the complex relationship between training performance and real-world application. The deployed model can be seen via Figure 9 on HuggingFace space. See Appendix A for more details.

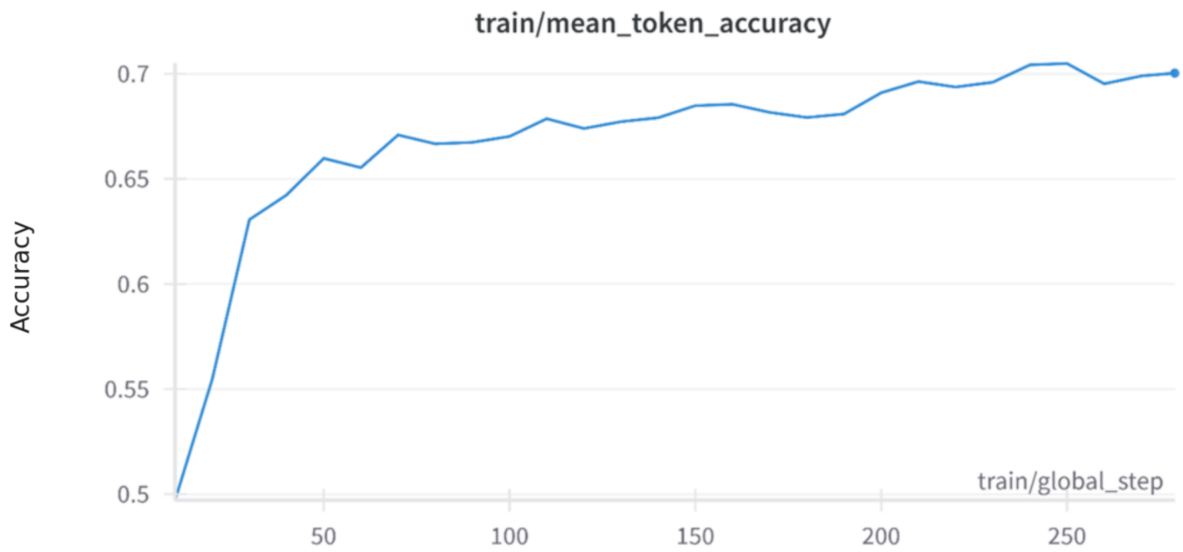

Fig. 7 Training Accuracy for fine-tuning

Fig. 8 Training Loss for fine-tuning

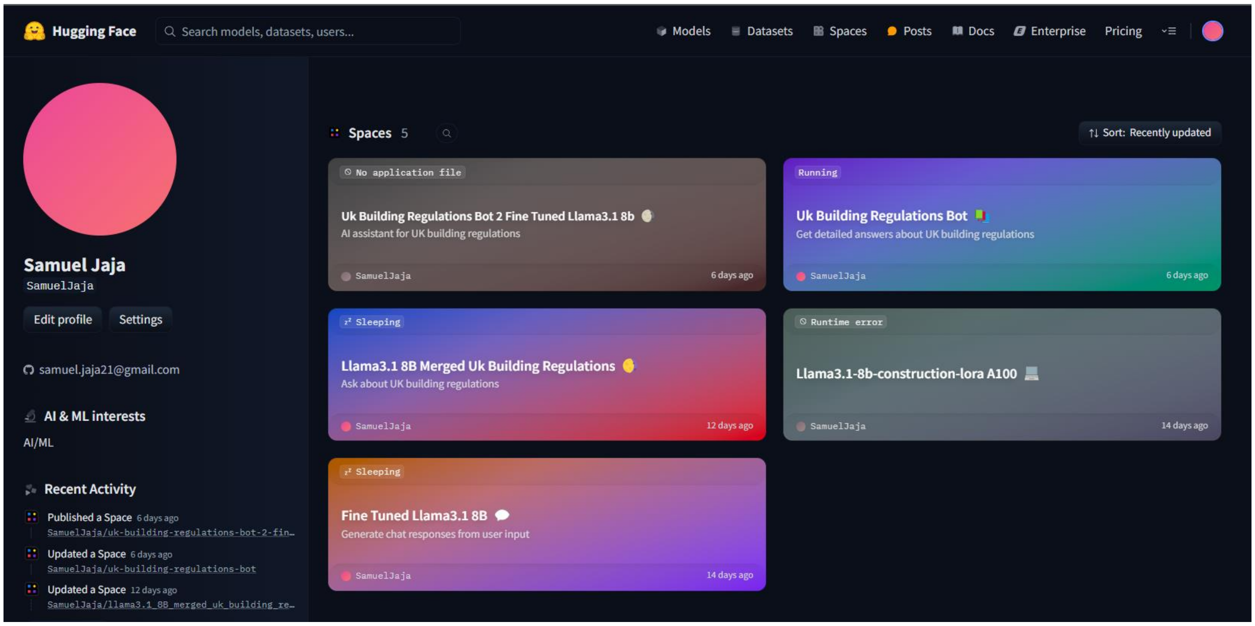

Fig 9: Hugging Face Space with deployed model

4.4 Operational Considerations

This operational comparison analysis on Table 5 demonstrates that each model configuration offers distinct advantages, allowing the system to adapt to different usage scenarios and requirements.

Table 5: Operational Comparison

Aspect	Llama-3.3-70B (GROQ)	Llama3-8B (GROQ)	Fine-tuned Llama-3.1-8B
Cost Structure	Pay-per-token	Pay-per-token	$0.80/active hour
Estimated Monthly Cost	Variable based on usage	Lower variable cost	$16-24 fixed cost
Control/Ownership	External API	External API	Full control, self-hosted
Availability	Dependent on GROQ	Dependent on GROQ	Independent deployment
Scalability	Excellent	Excellent	Limited by GPU resources

5.Discussion

5.1 Interpretation of Evaluation Results

The evaluation revealed key insights into how different model architectures perform in the building regulations domain. Identical context recall scores (0.8500) across all models confirm the consistency and effectiveness of the hybrid retrieval system, combining 70% vector similarity with 30% BM25 keyword matching to balance semantic and terminological precision.

Interestingly, the Llama-3.3-70B model scored highest in faithfulness (0.5516), challenging the assumption that fine-tuning would improve adherence to retrieved content. The Llama3-8B model achieved 0.4732, indicating model size alone does not determine contextual faithfulness. Lin et al. (2022) similarly found that compact BERT-based systems could deliver strong semantic performance in construction tasks. Xu et al. (2023) reported that QA-LoRA fine-tuned models outperformed QLoRA in both accuracy and speed, offering faster inference and better domain-specific precision. Faithfulness variation in this study supports Shahul et al. (2024), whose RAGAS framework highlights how fluent outputs may lack factual grounding. RAGAS's granular analysis shows when models deviate from retrieved evidence.

The standard Llama3-8B model surprisingly achieved the highest factual correctness (0.6880), outperforming both the 70B (0.5630) and fine-tuned (0.1420) models. This suggests the 8B architecture balances domain knowledge and precision effectively. The lower scores of the larger and fine-tuned models may stem from overgeneralization or insufficient training data.

Giskard evaluation results contrasted sharply with RAGAS. The Llama3-8B model dominated across all Giskard metrics. This matches Salemi and Zamani (2024), who found that traditional end-to-end evaluations often mask the influence of retrieved content, while their eRAG method showed a stronger correlation with performance. The 8B model's 100% retrieval score compared to 50% for the others further underscores this model's alignment with Giskard's emphasis on reference quality.

The discrepancy between RAGAS and Giskard underscores the need for multi-angle evaluation. Giskard may prioritize concise and fact-based usage of retrieved content, which the 8B model excels at.

Response time analysis showed that GROQ-hosted models were more efficient. Despite its size, the 70B model responded in 3.20 seconds nearly as fast as the 8B model (2.20 seconds) thanks to optimized infrastructure. In contrast, the fine-tuned model’s 5.70-second response highlights the difficulty of self-hosting large models, even with quantization.

5.2 Comparison with Related Work

These results both align with and challenge existing literature. The overall success of the RAG framework supports Lewis et al. (2021), who showed RAG's strength in grounding LLM responses. However, Zhang et al. (2023) found consistent gains from fine-tuning, which contrasts with this study where the fine-tuned model underperformed.

The importance of infrastructure is echoed by Hu et al. (2023), who showed that optimized hardware reduces large model latency. The performance gap between GROQ-hosted Llama3-8B and the self-hosted fine-tuned version supports Kaplan et al. (2023), who emphasized the critical role of deployment strategy. Meanwhile, Gururangan et al. (2022) observed steady improvements from domain-specific pretraining. Yet, the complex nature of building regulations, dense in technical language and structured content may demand more than standard fine-tuning.

5.3 Practical Implications

Practically, the Llama3-8B model's performance suggests that smaller models on optimized infrastructure can outperform larger or fine-tuned ones for regulatory tasks. This is ideal for cost-sensitive environments. Giskard scores support this: 80% for 8B vs. 60% (70B) and 20% (fine-tuned).

The 70B model, while slower and more resource-heavy, excels in formatting and structuring outputs, useful where presentation and trust matter. The fine-tuned model's results indicate that standard LoRA may be insufficient for technical domains. Multi-stage tuning may be needed for formatting, citations, and specification handling.

Hybrid retrieval's consistency across models confirms its strength and potential in other regulatory fields like healthcare or finance. Iaroshev et al. (2024) also noted that even commercial models like GPT-4o depend on high-quality embedding and chunking. Lin et al. (2022) showed mobile deployment is feasible, supporting the idea that StructureGPT could run on edge devices.

5.4 Limitations and Future Directions

Although the fine-tuned model used LoRA with carefully chosen settings, the training dataset was relatively small. Future work should test larger, more diverse datasets focused on formatting, citation style, and technical accuracy beyond general domain content. The mismatch between RAGAS and Giskard scores shows the need for better, more consistent evaluation methods tailored to technical domains. As Shahul et al. (2024) noted, RAGAS can vary depending on the LLM scoring it. They suggest using JSON prompts and reproducibility checks to improve consistency. To handle this, a SELF-RAG approach could help. Asai et al. (2023) showed that models trained with reflection tokens and built-in citation checking performed better in both accuracy and fact-checking. They also allowed flexible control over how much focus the model gives to completeness versus citation precision. Yang et al. (2024) point out that bigger isn't always better. Their research on U-shaped and inverse scaling shows that well-tuned mid-sized models can beat larger general-purpose ones if well aligned with the task. Future work should follow Jeong's (2024) strategy: growing the dataset, refining prompts, and using multi-stage instruction tuning to improve results across more building regulation sections.

Key future directions include:

1. **Multi-stage Retrieval** – Use different retrievers for text, specs, and diagrams to improve relevance.
2. **Hybrid Model Systems** – Route questions to different models based on content type.
3. **Better Fine-Tuning** – Try instruction-tuning and structure-aware architectures.
4. **Self-Verification** – Let models check their answers, especially for numbers and citations.
5. **Workflow Integration** – Build tools that plug directly into construction and compliance tasks, including agentic reasoning and multi-modal integration.

These ideas aim to improve how professionals' access and use complex building regulations.

6. Conclusion

This research successfully addressed all three objectives established at the outset: implementing a multi-model RAG system for UK Building Regulations, evaluating comparative performance across different architectures, and deploying a domain-specialized model with optimized efficiency. The comprehensive evaluation of three distinct models; Llama-3.3-70B, Llama3-8B, and fine-tuned Llama-3.1-8B-Instruct revealed significant insights into the trade-offs between model size, specialization, and deployment architecture. The consistent context recall (0.8500) across all models validated the effectiveness of the hybrid retrieval mechanism, while the variations in faithfulness and factual correctness demonstrated that different architectures offer complementary strengths for regulatory information retrieval. The Giskard evaluation results highlighted the unexpected superiority of the standard 8B model (80% overall correctness) compared to both the 70B implementation (60%) and the fine-tuned alternative (20%).

Beyond technical achievements, this project established a practical framework for making complex regulatory information more accessible to construction professionals. The unified Streamlit interface with model-switching capabilities provides an adaptable solution that balances performance, cost, and control considerations while maintaining source attribution and transparency.

As building regulations continue to evolve in response to safety and sustainability requirements, the multi-model approach developed in this research offers a scalable methodology for ensuring stakeholders can efficiently access and accurately understand critical regulatory information. This contribution extends beyond the construction industry to provide valuable insights for AI implementation across all regulated domains requiring specialized knowledge dissemination.

References

Arslan, M., Mahdjoubi, L. & Munawar, S. (2024). Driving sustainable energy transitions with a multi-source RAG-LLM system. Energy and Buildings, 324(114827). https://doi.org/10.1016/j.enbuild.2024.114827.

Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2023). SELF-RAG (Asai et al. (2023): Learning to Retrieve, Generate, and Critique through Self-Reflection. arXiv. https://arxiv.org/abs/2310.11511.

Babu, G.B. (2024). Abstractive YouTube Transcript Summarizer and Domain-Specific Question-Answering Chatbot for Residential Construction Guidance. MSc thesis. University of Hull.

Bridgelall, R. (2024). Unraveling the mysteries of AI chatbots. Artificial Intelligence Review, 57, pp. 1–35. https://doi.org/10.1007/s10462-024-10720-7.

Bruch, S., Gai, S., & Ingber, A. (2023). *An Analysis of Fusion Functions for Hybrid Retrieval*. arXiv preprint arXiv:2210.11934. https://arxiv.org/pdf/2210.11934

Byun, J., Kim, B., Cha, K.-A. & Lee, E. (2024). Design and Implementation of an Interactive Question-Answering System with Retrieval-Augmented Generation for Personalized Databases. Applied Sciences, 14(7995). https://doi.org/10.3390/app14177995

Colangelo, M.T., Meleti, M., Guizzardi, S., Calciolari, E., & Galli, C. (2025). *A Comparative Analysis of Sentence Transformer Models for Automated Journal Recommendation Using PubMed Metadata*. Big Data and Cognitive Computing, 9(3), 67. https://doi.org/10.3390/bdcc9030067

Doumanas, D., Soularidis, A., Spiliotopoulos, D., Vassilakis, C., & Kotis, K. (2025). Fine-Tuning Large Language Models for Ontology Engineering (Doumanas et al. (2025)): A Comparative Analysis of GPT-4 and Mistral. Applied Sciences, 15(4), 2146. https://doi.org/10.3390/app15042146.

Fan, W., Ding, Y., Ning, L., Wang, S., Li, H., Yin, D., Chua, T.-S., & Li, Q. (2024). A Survey on RAG Meeting LLMs: Towards Retrieval-Augmented Large Language Models. KDD 2024. https://doi.org/10.48550/arXiv.2405.06211.

Ghimire, P., Kim, K. & Acharya, M. (2024). Opportunities and challenges of generative AI in the construction industry: Focusing on adoption of text-based models. Buildings, 14(220). https://doi.org/10.3390/buildings14010220.

GOV.UK Ministry of Housing, Communities and Local Government (2024). The Merged Approved Documents: Building Regulations 2010 for use in England, October 2024 Compilation. Available at: https://www.gov.uk/government/collections/approved-documents [Accessed: 28 January 2025].

Iaroshev, I., Pillai, R., Vaglietti, L., & Hanne, T. (2024). Evaluating Retrieval-Augmented Generation Models for Financial Report Question and Answering. Applied Sciences, 14(20), 9318. https://doi.org/10.3390/app14209318.

Izacard, G., Lewis, P., Lomeli, M., Hosseini, L., Petroni, F., Schick, T., Dwivedi-Yu, J., Joulin, A., Riedel, S. & Grave, E. (2023). Atlas: Few-shot Learning with Retrieval-Augmented Language Models. Journal of Machine Learning Research, 24, pp. 1–43. http://jmlr.org/papers/v24/23-0037.html.

Jeong, C. (2024). Fine-tuning and Utilization Methods of Domain-specific LLMs. SAMSUNG SDS. [Unpublished Technical Report].

Lin, T.-H., Huang, Y.-H., & Putranto, A. (2022). Intelligent question and answer system for building information modeling and artificial intelligence of things based on the BERT model. Automation in Construction, 142, 104483. https://doi.org/10.1016/j.autcon.2022.104483.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S. & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. arXiv. https://doi.org/10.48550/arXiv.2005.11401.

Onatayo, D., Onososen, A., Oyediran, A.O., Oyediran, H., Arowoiya, V. & Onatayo, E. (2024). Generative AI applications in architecture, engineering, and construction: Trends, implications for practice, education & imperatives for upskilling—*A review*. *Architecture*, 4(877–902). https://doi.org/10.3390/architecture4040046

Rane, N. (2023). ChatGPT and Similar Generative Artificial Intelligence (AI) for Building and Construction Industry. SSRN. https://ssrn.com/abstract=4603221.

Rengo, M., Beadini, S., Alfano, D., & Abbruzzese, R. (2025). A System for Comprehensive Assessment of RAG Frameworks. arXiv. https://arxiv.org/abs/2504.07803.

Salemi, A. & Zamani, H. (2024). Evaluating Retrieval Quality in Retrieval-Augmented Generation. In Proceedings of the 47th ACM SIGIR Conference. https://doi.org/10.1145/nnnnnnn.nnnnnnn.

Shahul Es, James, J., Espinosa-Anke, L., & Schockaert, S. (2024). RAGAS (Shahul et al. (2024)): Automated Evaluation of Retrieval-Augmented Generation. Proceedings of the 18th EACL: System Demonstrations, pp. 150–158. https://aclanthology.org/2024.eacl-demo.16/

Su, W., Tang, Y., Ai, Q., Wu, Z., & Liu, Y. (2024). Dynamic Retrieval Augmented Generation based on the Information Needs of Large Language Models. ACL 2024, pp. 12991–13013. https://doi.org/10.18653/v1/2024.acl-long.702.

Taiwo, R., Bello, I.T., Abdulai, S.F., Yussif, A.-M., Salami, B.A., Saka, A. & Zayed, T. (2024). Generative AI in the Construction Industry: A State-of-the-art Analysis. The Hong Kong Polytechnic University. http://dx.doi.org/10.48550/arXiv.2402.09939.

Unstructured.io (2024). *Unstructured API Documentation*. Available at: https://unstructured.io [Accessed: 16 February 2025].

Wang, L., Chen, S., Jiang, L., Pan, S., Cai, R., Yang, S. & Yang, F. (2024). Parameter-efficient fine-tuning in large models: A survey of methodologies. arXiv. https://doi.org/10.48550/arXiv.2410.19878.

Wulf, A.J. & Seizov, O. (2020). Artificial Intelligence and Transparency: A Blueprint for Improving the Regulation of AI Applications in the EU. European Business Law Review, 31(4), 611–640.

Xu, Y., Xie, L., Gu, X., Chen, X., Chang, H., Zhang, H., Chen, Z., Zhang, X., & Tian, Q. (2023). QA-LoRA (Xu et al. (2023)): Quantization-Aware Low-Rank Adaptation of Large Language Models. arXiv. https://arxiv.org/abs/2309.14717.

Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., Zhong, S., Yin, B. & Hu, X. (2024). Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. ACM TKDD, 18(6), Article 160. https://doi.org/10.1145/3649506.

Appendices

Appendix A – Deployment Code on HuggingFace Spaces

This appendix outlines the codebase used to deploy the *StructureGPT* chatbot on Hugging Face Spaces. The deployment leverages a Streamlit-based frontend for user interaction and integrates three LLaMA model variants through a unified Retrieval-Augmented Generation (RAG) pipeline. The backend includes components for semantic chunking, vector database retrieval (via ChromaDB), and model switching via Streamlit UI.

Please note: The code is intended for deployment via Hugging Face's cloud environment and cannot be executed directly in Jupyter Notebook environments due to the use of Streamlit and GPU-based APIs (e.g., Groq API for model inference).

Live Deployment (Hugging Face Spaces): UK Building Regulations Chatbot

User Interface Deployment Code: Hugging Face – app.py (UI)

Model Deployment Code: Hugging Face – app.py (Fine-tuned Model)

Appendix B - Data Curation Pipeline and Prompting Strategy

A hybrid approach was used to curate instruction-style data for fine-tuning the model on UK Building Regulations. Pre-existing GOV.UK FAQs served as the primary content source where available, though many regulation parts either lacked FAQs entirely or had fewer than six questions. To ensure adequate coverage, additional question–answer pairs were created using content extracted directly from official GOV.UK PDFs. Each PDF was classified into its respective regulation parts based on filename and content heuristics, and the text was extracted using the pypdf library. To preserve context, paragraph-based chunking (~7,500 characters) was used, with sentence-level fallback for longer sections. These chunks were then passed into GPT models to structure the data. GPT-3.5-turbo was initially used, but it frequently misinterpreted technical content or returned inconsistent formats. GPT-4o-mini was therefore selected for its superior comprehension, lower latency, and ability to consistently generate accurate, instruction-style Q&A pairs with minimal hallucination. The prompt instructed the model to focus on compliance, measurements, technical requirements, and to output strictly formatted JSON using the instruction, input, and output fields. This allowed the dataset to align with Alpaca-style fine-tuning standards. Approximately 200 examples were generated per regulation part, resulting in a balanced dataset of over 3,000 entries. This method ensured full section coverage, format consistency, and domain relevance, all of which are essential for effective LoRA-based fine-tuning.

Appendix C – Tools Used in the Vector Store and Further Justification

- Unstructured API: Extracts clean text, sections, and tables from PDF documents. It helped separate out content in a usable format.
- YOLOX: A computer vision model used here to detect and preserve layout features like tables and multi-columns that often appear in building regulation documents.
- Paragraph chunking: Text was split into paragraphs instead of random word limits, helping the system better understand full ideas from the regulation.
- Embedding model – all-mpnet-base-v2: Turns paragraphs into numerical vectors that capture their meaning. Chosen because it's free and works well for comparing sentence-level meaning.
- ChromaDB: A vector database used to store and search through embedded text chunks efficiently and quickly.
- BM25 (Best Matching 25): A method that scores documents based on how often and how exactly they match the keywords in the user's question.
- Hybrid search (70% vector + 30% BM25): Combines both meaning-based and keyword-based search to make sure responses are accurate and legally correct. This is important for regulation content where both the meaning and the exact words matter.

Appendix D – Why Groq API, LangChain, and Prompting Strategy

To implement StructureGPT with large open-weight models such as LLaMA3-3.3-70B and LLaMA3-8B, the project utilized the Groq Cloud API. Groq was chosen for its ability to serve large models with ultra-low latency, often returning responses up to 10x faster than traditional cloud inference providers. This was especially valuable for real-time applications like a chatbot interface, where user experience depends heavily on quick and consistent responses. Groq's support for open-source models made it suitable for working with the latest LLaMA versions without needing to deploy or maintain costly GPU infrastructure. Additionally, it allowed fine-grained control over parameters such as temperature, top-k sampling, and context window length, which helped optimize model behavior for regulatory question answering.

To manage the interaction with different models and structure the prompt logic, the project integrated LangChain, an open-source framework for large language model orchestration. LangChain was selected for its modular design, which simplifies managing multiple model configurations and routing queries dynamically between them. It enabled the system to define chains that included prompt templates, context injection, and output parsing, all within a consistent framework. LangChain's support for fallback and error-handling logic also allowed the chatbot to gracefully recover from timeouts or malformed responses by retrying or switching to a smaller model configuration. This level of robustness was critical in a production-facing application meant to deliver reliable regulatory guidance.

Prompt engineering played a vital role in ensuring the system generated accurate and regulation-grounded responses. Prompts were designed in a structured, instruction-following style that encouraged factual, non-speculative answers. Each interaction included a system message defining the model's role (e.g., "You are an expert assistant for UK Building Regulations") and a user prompt combining the query with document-specific context extracted during retrieval. For example:

System Prompt:
You are an expert assistant for UK Building Regulations. Always base your answers strictly on the provided document context.

User Prompt:
Answer this question using the content from UK Building Regulations:
Question: What is the minimum fire resistance for internal walls in a residential building?
Context: According to Part B, Section 5.1, internal walls separating dwellings must achieve a minimum fire resistance rating of 30 minutes.

This approach ensured the model remained anchored to the authoritative source and prevented hallucinations, which is especially important in legal and regulatory domains. Prompts were templated using LangChain, allowing them to be reused across different model types with minimal duplication.

Finally, error-handling mechanisms were implemented within LangChain workflows. These included timeout limits for Groq API calls, retries with trimmed context in case of token overflow, and fallbacks that defaulted to simpler queries or a smaller model if the primary model failed. These safeguards ensured the chatbot remained responsive and reliable even under variable API conditions. See **LangChain Documentation** **Groq API Documentation**

GROQ was chosen for two key reasons:

1. **Inference Speed**: GROQ's LPU (Language Processing Unit) architecture is highly optimized for low-latency LLM inference, offering sub-3 second response times even for the 70B model. This made it possible to test large models interactively, without local GPU constraints.
2. **Scalable Access to Large Models**: Instead of running expensive infrastructure locally, GROQ allowed seamless API-based access to state-of-the-art LLaMA models, ideal for benchmarking and comparison without overhead resource.

The selection of three different models, LLaMA 3.3-70B, LLaMA 3-8B, and a LoRA fine-tuned LLaMA 3.1-8B was driven by the need to balance performance, specialization, and efficiency. The 70B model tested the upper bound of general capabilities. The 8B Groq-hosted model represented an infrastructure-optimized alternative for real-world deployment. The fine-tuned model explored whether domain-specific tuning would yield improved contextual relevance and factual grounding. Comparing these three enabled a comprehensive analysis of trade-offs across model size, specialization, and retrieval utilization.

Appendix E – Other Questions and Model Responses

Question 1 - What factors determine the minimum width of strip foundations in different ground conditions?

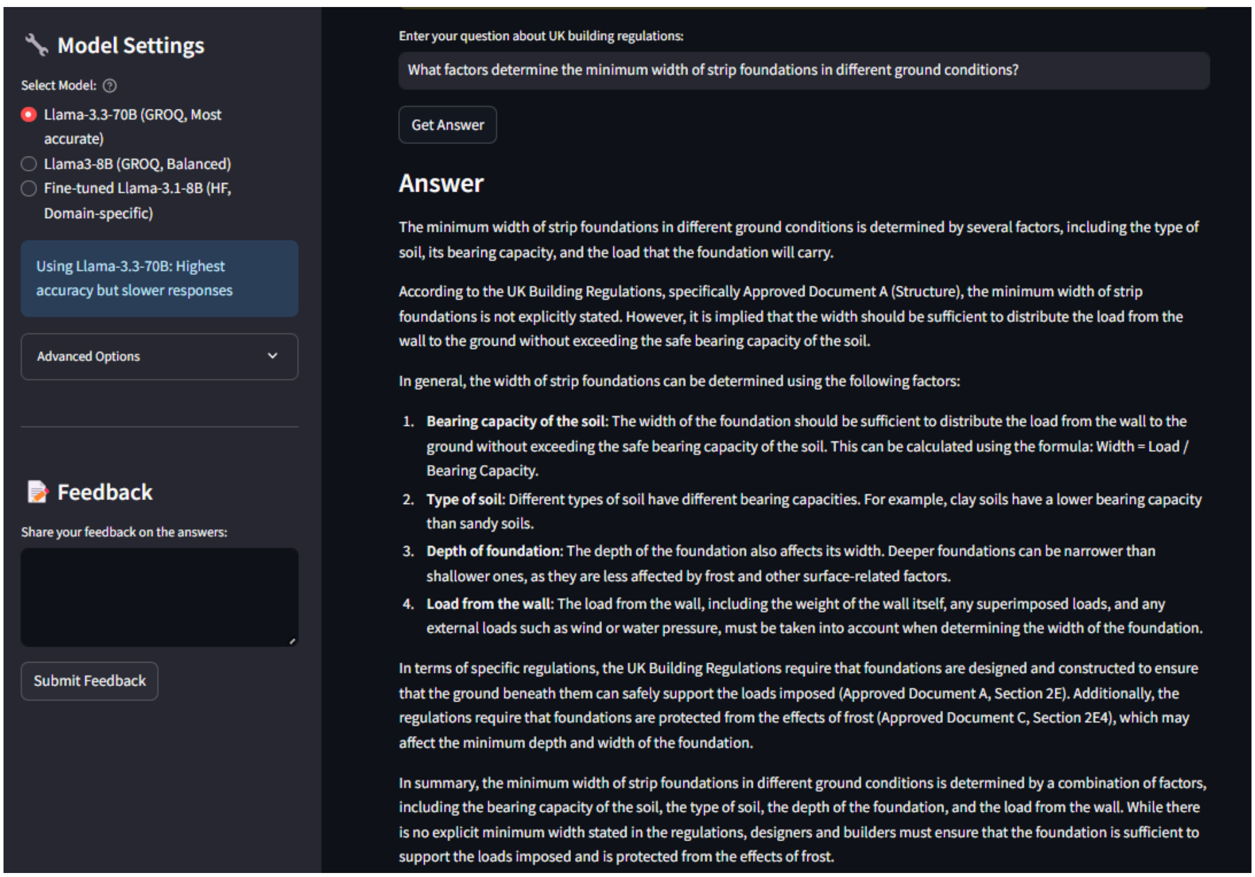

Figure E1 - Llama-3.3-70B Response to Question 1

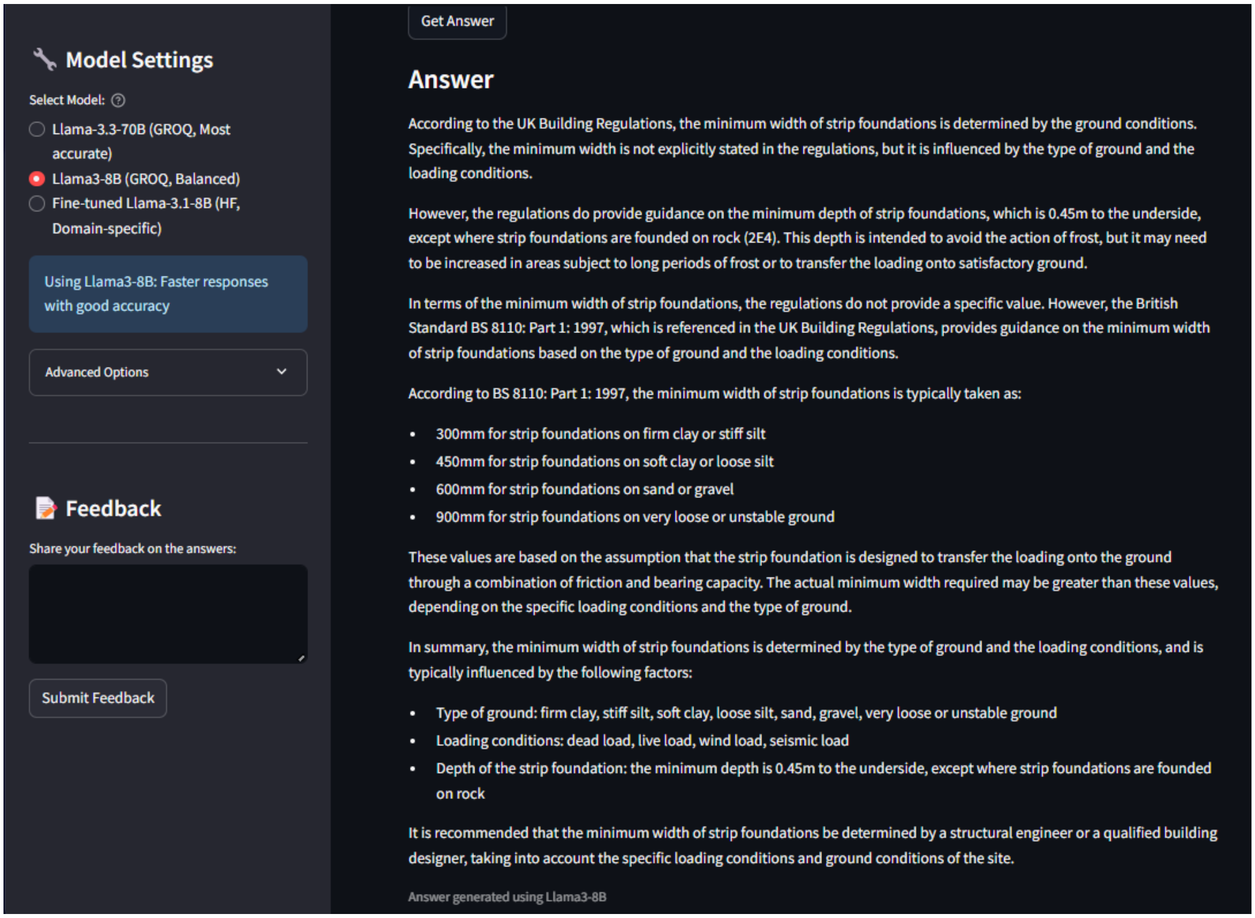

Figure E2 - Llama-3-8B Response to Question 1

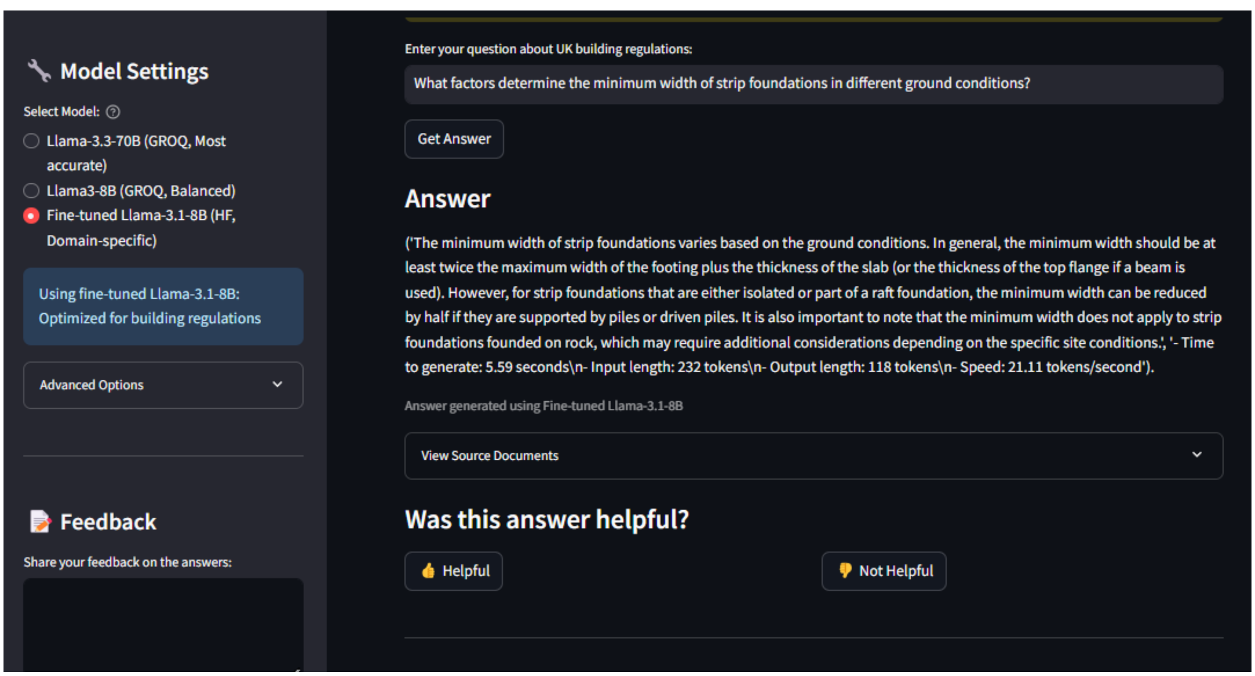

Figure E3 - Fine-tuned Llama-3.1-8B Question 1

Question 2 - Which document should I refer to in order to find out if a cavity wall requires insulation from the pitched roof level, and how deep the insulation should extend into the cavity?

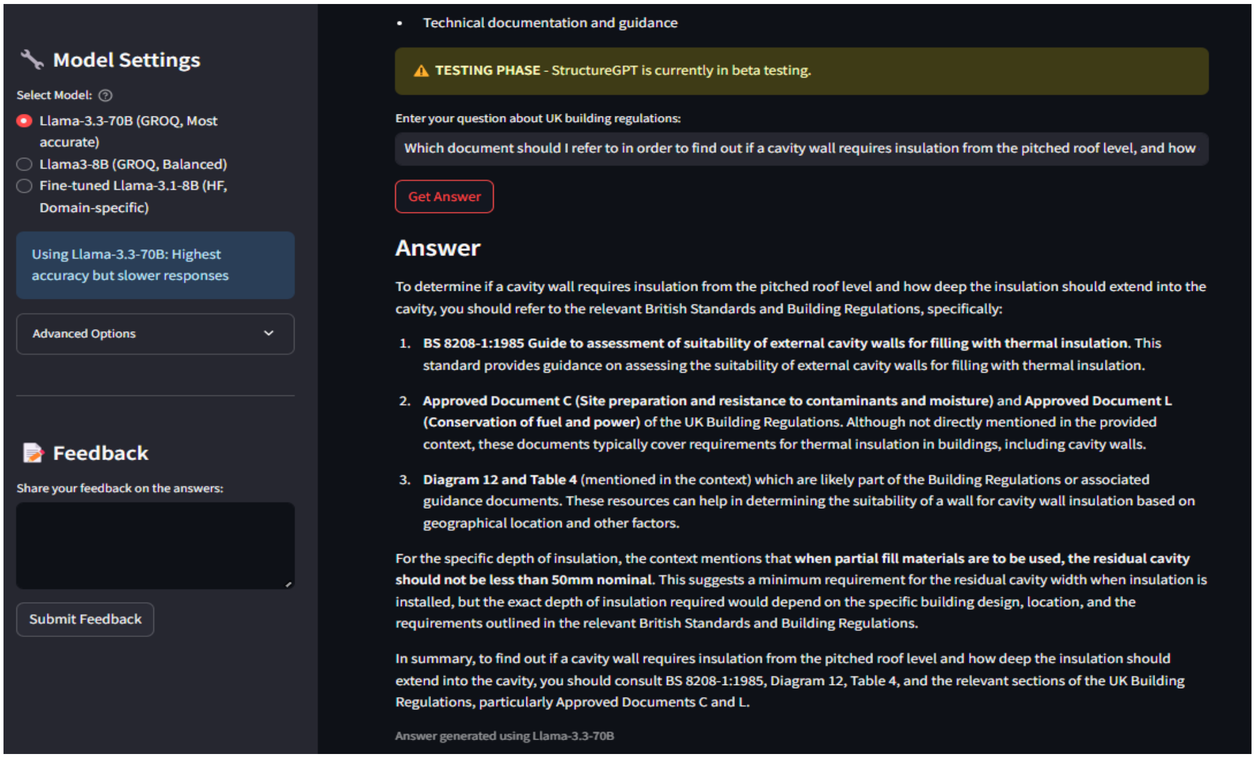

Figure E4 - Llama-3.3-70B Response to Question 2

Table E1- Further Questions Comparison

Model	**Question 1 (Strip Foundations)**	**Question 2 (Cavity Wall Insulation)**
Llama-3.3-70B (E1, E4)	Clear explanation with regulation links; lacked numerical values	Most comprehensive; cited BS 8208, Diagrams, Approved Documents
Llama-3-8B (E2, E5)	Included BS 8110 with exact widths (300–900mm); very practical	Cited same docs but less detail than 70B; slightly vague on exact sources
Llama-3.1-8B Fine-tuned (E3, E6)	Domain-specific logic; mentioned practical sizing formula, not standards	Very concise; repeated others' points but less readable formatting

Table E1 further compares model responses as seen from Figures E1, E2, E3, E4, E5 and E6.

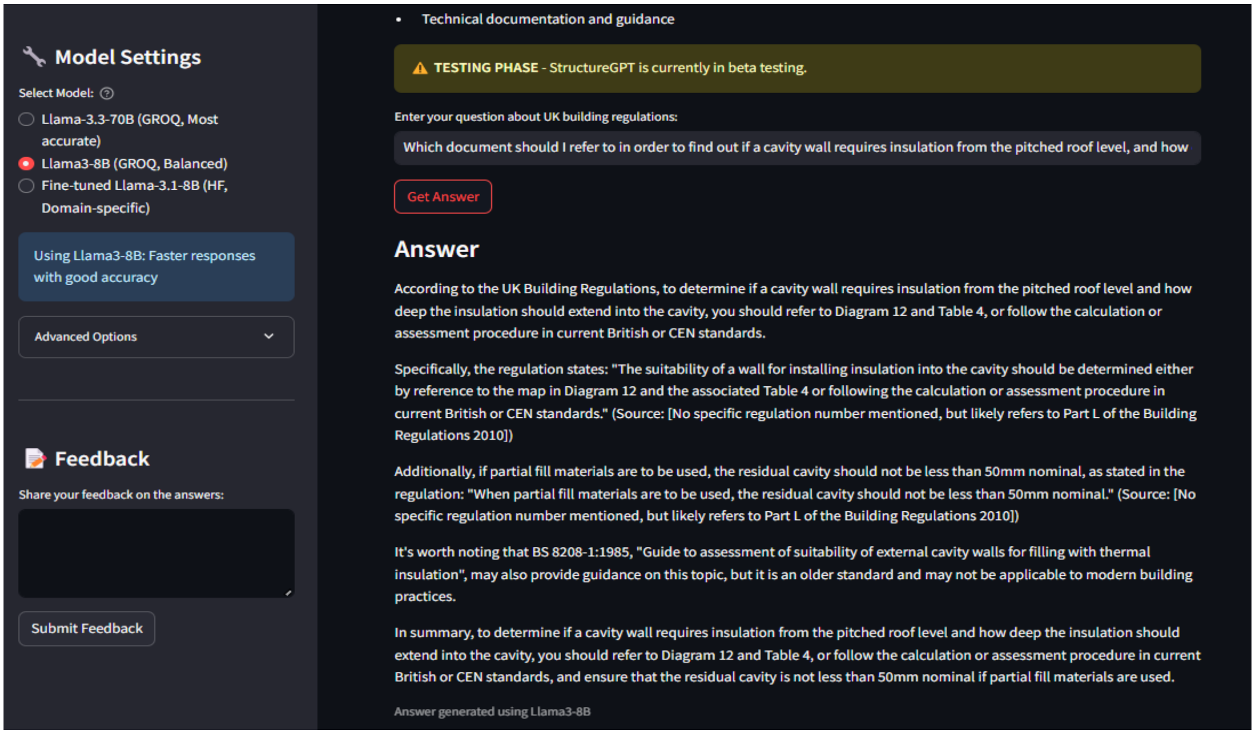

Figure E5 - Llama-3-8B Response to Question 2

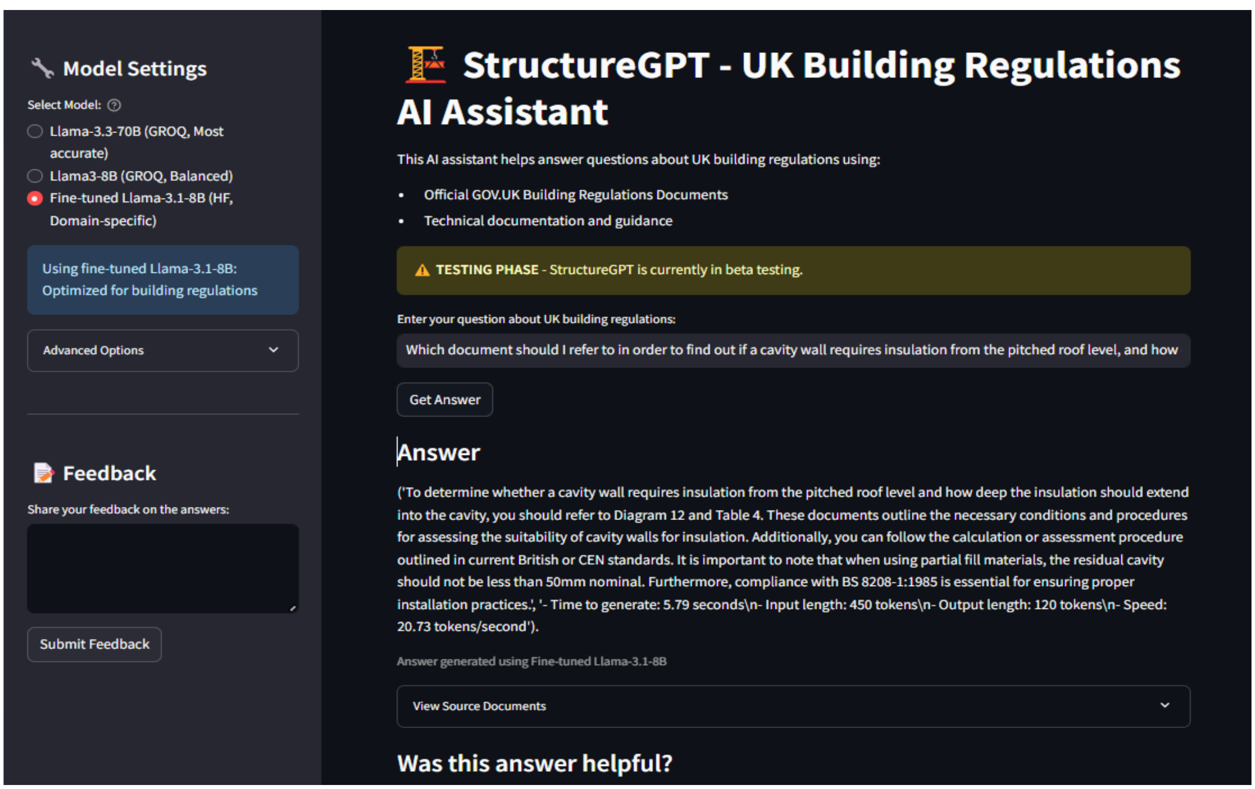

Figure E6 - Fine-tuned Llama-3.1-8B Question 2

Appendix F – Giskard Evaluation Output

Giskard Evaluation do not remain visible after the Jupyter Notebook is closed. Therefore, I have included this screenshot to preserve and illustrate the results that were originally rendered within the notebook but are not displayed correctly upon reopening. See Figure F1, F2 and F3 below.

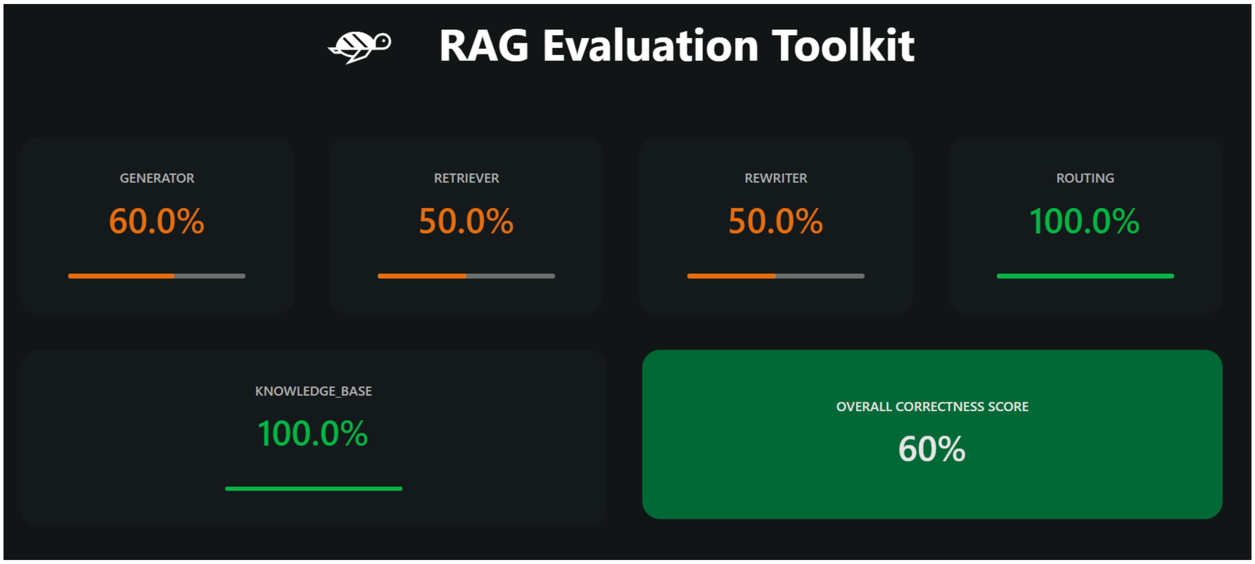

Figure F1 - Giskard Evaluation Llama3.3-70B

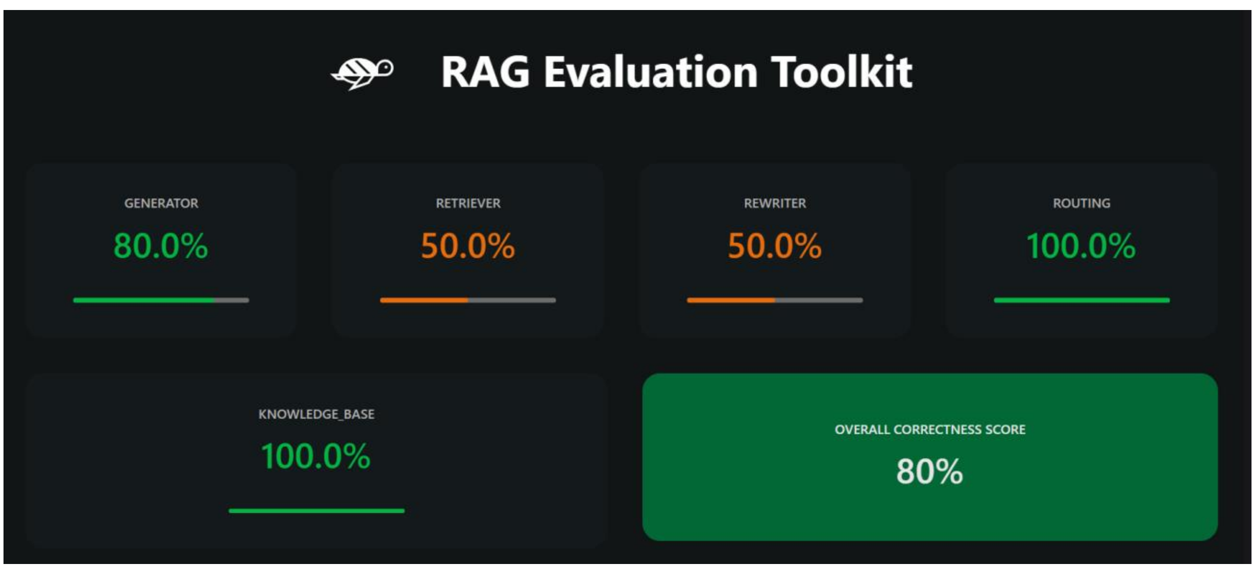

Figure F2- Giskard Evaluation Llama3.1-8B

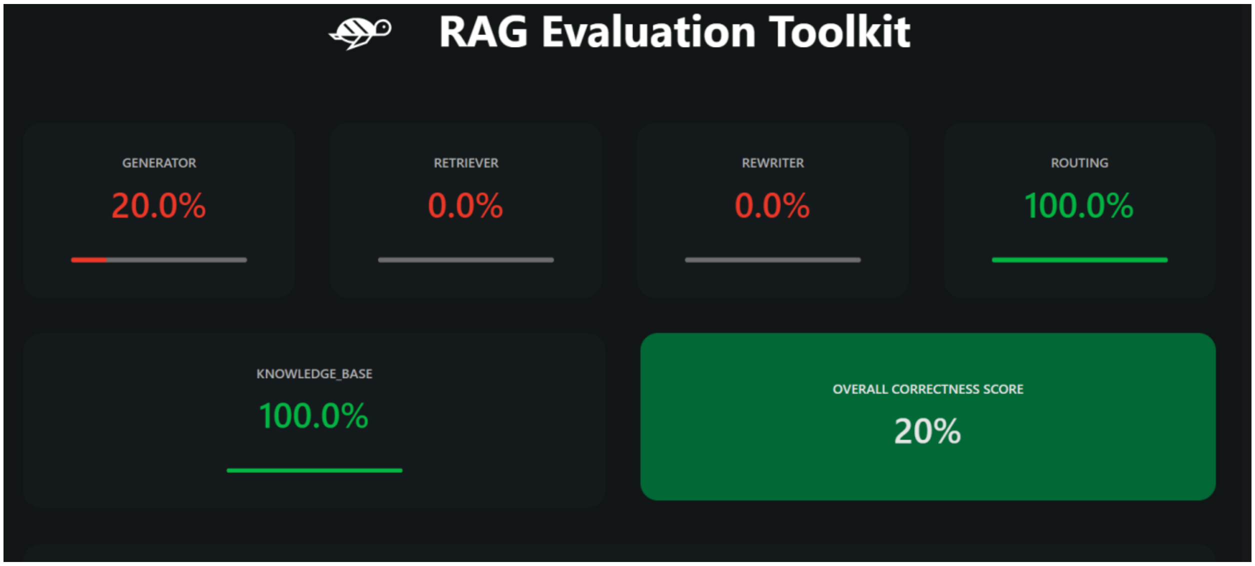

Figure F3 - Giskard Evaluation Fine-tuned Model