

Low-Code Modular RAG-based Knowledge Search Engine Prompts

In the Low-Code Modular RAG-based Knowledge Search Engine use case we are using a Generative AI Zero-shot Prompting technique to get the results. It consists of 1 prompt:

Important Note:

All examples and data presented in this article are purely fictive and for showcasing purposes only. Any resemblance to real persons, or actual companies is purely coincidental. Dummy users and entities have been created for this demo, and no real data or information about individuals or organizations is being disclosed. We prioritize privacy and ethical considerations in the presentation of information, and any similarities to real-world entities are unintentional.

Prompt 1:

RAG Response Synthesized Answer

To provide a synthesized answer in natural language based on the RAG Query Engine Response, we have created a prompt using a Zero-shot technique, where we explicitly mention the output we need.

Answer the <Question> based only on the provided <JSON documents>. If the <JSON documents> do not contain the answer to the <Question>, then respond with 'I do not find the answer in the documents'. Please do not try to search for information about the <Question>, and do not try to answer based on your knowledge base, instead, you must provide a very concise answer based only on the <JSON documents>, and if you do not find the answer in the <JSON documents>, then you answer with the phrase 'I do not find the answer in the documents'. The answer has the following json format: { "found": "Boolean response found in document", "document": "Document where the response is found", "response": "response of the question" } \n\n<Question>: "how much of CPU the framework consumes?" \n\n<JSON documents>: \n\n[{ "Document": "autonomous-database-self-repairing-5116047.pdf", \n"Context": "\", \n"Sentence": "machine learning to improve database runtime availability and performance with \nmaximum efficiency. The framework is lightweight and typically consumes less than 3% of CPU \nresources at runtime, so most processes run 24/7 as a daemon in the background." }, { "Document": "autonomous-database-self-repairing-5116047.pdf", \n"Context": "\", \n"Sentence": "corrective actions to prevent severe performance or availability issues. \nIt uses over 150 signals, synchronized every second, between the OS and the database to get to \nthe root cause and recommend appropriate action, such as increasing memory or CPU resources. \n•" }]