Authentic Databricks Databricks-Generative-AI-Engineer-Associate Exam Questions with Answers

P.S. Free & New Databricks-Generative-AI-Engineer-Associate dumps are available on Google Drive shared by Prep4sures: https://drive.google.com/open?id=1jmHHJjJxDYYr1m0pE-9Vzp3rZ88VHrLd

It is known to us that our Databricks-Generative-AI-Engineer-Associate study materials are enjoying a good reputation all over the world. Our study materials have been approved by thousands of candidates. You may have some doubts about our product or you may suspect the pass rate of it, but we will tell you clearly, it is totally unnecessary. If you still do not trust us, you can choose to download demo of our Databricks-Generative-AI-Engineer-Associate Test Torrent. The high quality and the perfect service system after sale of our Databricks-Generative-AI-Engineer-Associate exam questions have been approbated by our local and international customers. So you can rest assured to buy.

Databricks Databricks-Generative-AI-Engineer-Associate Exam Syllabus Topics:

Topic	Details
Topic 1	• Governance: Generative AI Engineers who take the exam get knowledge about masking techniques, guardrail techniques, and legal • licensing requirements in this topic.

Topic 2	• Assembling and Deploying Applications: In this topic, Generative AI Engineers get knowledge about coding a chain using a pyfunc mode, coding a simple chain using langchain, and coding a simple chain according to requirements. Additionally, the topic focuses on basic elements needed to create a RAG application. Lastly, the topic addresses sub-topics about registering the model to Unity Catalog using MLflow.
Topic 3	• Evaluation and Monitoring: This topic is all about selecting an LLM choice and key metrics. Moreover, Generative AI Engineers learn about evaluating model performance. Lastly, the topic includes sub-topics about inference logging and usage of Databricks features.
Topic 4	• Application Development: In this topic, Generative AI Engineers learn about tools needed to extract data, Langchain • similar tools, and assessing responses to identify common issues. Moreover, the topic includes questions about adjusting an LLM's response, LLM guardrails, and the best LLM based on the attributes of the application.

Pass Guaranteed Databricks - Fantastic Test Databricks-Generative-AI-Engineer-Associate Simulator

There may be some other study materials with higher profile and lower price than our products, but we can assure you that the passing rate of our Databricks-Generative-AI-Engineer-Associate learning materials is much higher than theirs. And this is the most important. According to previous data, 98 % to 99 % of the people who use our Databricks-Generative-AI-Engineer-Associate Training Questions passed the exam successfully. If you are willing to give us a trust on our Databricks-Generative-AI-Engineer-Associate exam questions, we will give you a success.

Databricks Certified Generative AI Engineer Associate Sample Questions (Q58-Q63):

NEW QUESTION # 58
A Generative Al Engineer is building a production-ready LLM system which replies directly to customers.
The solution makes use of the Foundation Model API via provisioned throughput. They are concerned that the LLM could potentially respond in a toxic or otherwise unsafe way. They also wish to perform this with the least amount of effort.
Which approach will do this?

- A. Ask users to report unsafe responses
- B. Add a regex expression on inputs and outputs to detect unsafe responses.
- C. Host Llama Guard on Foundation Model API and use it to detect unsafe responses
- D. Add some LLM calls to their chain to detect unsafe content before returning text

Answer: C

Explanation:
The task is to prevent toxic or unsafe responses in an LLM system using the Foundation Model API with minimal effort. Let's assess the options.
* Option A: Host Llama Guard on Foundation Model API and use it to detect unsafe responses
* Llama Guard is a safety-focused model designed to detect toxic or unsafe content. Hosting it via the Foundation Model API (a Databricks service) integrates seamlessly with the existing system, requiring minimal setup (just deployment and a check step), and leverages provisioned throughput for performance.
* Databricks Reference:"Foundation Model API supports hosting safety models like Llama Guard to filter outputs efficiently" ("Foundation Model API Documentation," 2023).
* Option B: Add some LLM calls to their chain to detect unsafe content before returning text
* Using additional LLM calls (e.g., prompting an LLM to classify toxicity) increases latency, complexity, and effort (crafting prompts, chaining logic), and lacks the specificity of a dedicated safety model.
* Databricks Reference:"Ad-hoc LLM checks are less efficient than purpose-built safety solutions" ("Building LLM Applications with Databricks").

* Option C: Add a regex expression on inputs and outputs to detect unsafe responses
* Regex can catch simple patterns (e.g., profanity) but fails for nuanced toxicity (e.g., sarcasm, context-dependent harm), requiring significant manual effort to maintain and update rules.
* Databricks Reference:"Regex-based filtering is limited for complex safety needs"("Generative AI Cookbook").
* Option D: Ask users to report unsafe responses
* User reporting is reactive, not preventive, and places burden on users rather than the system. It doesn't limit unsafe outputs proactively and requires additional effort for feedback handling.
* Databricks Reference:"Proactive guardrails are preferred over user-driven monitoring" ("Databricks Generative AI Engineer Guide").
Conclusion: Option A (Llama Guard on Foundation Model API) is the least-effort, most effective approach, leveraging Databricks' infrastructure for seamless safety integration.

NEW QUESTION # 59
A Generative Al Engineer is creating an LLM-based application. The documents for its retriever have been chunked to a maximum of 512 tokens each. The Generative Al Engineer knows that cost and latency are more important than quality for this application. They have several context length levels to choose from.
Which will fulfill their need?

- A. context length 512: smallest model is 0.13GB and embedding dimension 384
- B. context length 32768: smallest model is 14GB and embedding dimension 4096
- C. context length 2048: smallest model is 11GB and embedding dimension 2560
- D. context length 514; smallest model is 0.44GB and embedding dimension 768

Answer: A

Explanation:
When prioritizing cost and latency over quality in a Large Language Model (LLM)-based application, it is crucial to select a configuration that minimizes both computational resources and latency while still providing reasonable performance. Here's whyDis the best choice:
* Context length: The context length of 512 tokens aligns with the chunk size used for the documents (maximum of 512 tokens per chunk). This is sufficient for capturing the needed information and generating responses without unnecessary overhead.
* Smallest model size: The model with a size of 0.13GB is significantly smaller than the other options.
This small footprint ensures faster inference times and lower memory usage, which directly reduces both latency and cost.
* Embedding dimension: While the embedding dimension of 384 is smaller than the other options, it is still adequate for tasks where cost and speed are more important than precision and depth of understanding.
This setup achieves the desired balance between cost-efficiency and reasonable performance in a latency- sensitive, cost-conscious application.

NEW QUESTION # 60
A Generative AI Engineer has been asked to design an LLM-based application that accomplishes the following business objective: answer employee HR questions using HR PDF documentation.
Which set of high level tasks should the Generative AI Engineer's system perform?

- A. Split HR documentation into chunks and embed into a vector store. Use the employee question to retrieve best matched chunks of documentation, and use the LLM to generate a response to the employee based upon the documentation retrieved.
- B. Create an interaction matrix of historical employee questions and HR documentation. Use ALS to factorize the matrix and create embeddings. Calculate the embeddings of new queries and use them to find the best HR documentation. Use an LLM to generate a response to the employee question based upon the documentation retrieved.
- C. Use an LLM to summarize HR documentation. Provide summaries of documentation and user query into an LLM with a large context window to generate a response to the user.
- D. Calculate averaged embeddings for each HR document, compare embeddings to user query to find the best document. Pass the best document with the user query into an LLM with a large context window to generate a response to the employee.

Answer: A

Explanation:
To design an LLM-based application that can answer employee HR questions using HR PDF documentation, the most effective approach is option D. Here's why:

* Chunking and Vector Store Embedding:HR documentation tends to be lengthy, so splitting it into smaller, manageable chunks helps optimize retrieval. These chunks are then embedded into avector store(a database that stores vector representations of text). Each chunk of text is transformed into an embeddingusing a transformer-based model, which allows for efficient similarity-based retrieval.
* Using Vector Search for Retrieval:When an employee asks a question, the system converts their query into an embedding as well. This embedding is then compared with the embeddings of the document chunks in the vector store. The most semantically similar chunks are retrieved, which ensures that the answer is based on the most relevant parts of the documentation.
* LLM to Generate a Response:Once the relevant chunks are retrieved, these chunks are passed into the LLM, which uses them as context to generate a coherent and accurate response to the employee's question.
* Why Other Options Are Less Suitable:
* A (Calculate Averaged Embeddings): Averaging embeddings might dilute important information. It doesn't provide enough granularity to focus on specific sections of documents.
* B (Summarize HR Documentation): Summarization loses the detail necessary for HR-related queries, which are often specific. It would likely miss the mark for more detailed inquiries.
* C (Interaction Matrix and ALS): This approach is better suited for recommendation systems and not for HR queries, as it's focused on collaborative filtering rather than text-based retrieval.
Thus, option D is the most effective solution for providing precise and contextual answers based on HR documentation.

NEW QUESTION # 61
A Generative AI Engineer is creating an agent-based LLM system for their favorite monster truck team. The system can answer text based questions about the monster truck team, lookup event dates via an API call, or query tables on the team's latest standings. How could the Generative AI Engineer best design these capabilities into their system?

- A. Build a system prompt with all possible event dates and table information in the system prompt. Use a RAG architecture to lookup generic text questions and otherwise leverage the information in the system prompt.
- B. Ingest PDF documents about the monster truck team into a vector store and query it in a RAG architecture.
- C. Write a system prompt for the agent listing available tools and bundle it into an agent system that runs a number of calls to solve a query.
- D. Instruct the LLM to respond with "RAG", "API", or "TABLE" depending on the query, then use text parsing and conditional statements to resolve the query.

Answer: C

Explanation:
In this scenario, the Generative AI Engineer needs to design a system that can handle different types of queries about the monster truck team. The queries may involve text-based information, API lookups for event dates, or table queries for standings. The best solution is to implement atool-based agent system.
Here's how option B works, and why it's the most appropriate answer:
* System Design Using Agent-Based Model:In modern agent-based LLM systems, you can design a system where the LLM (Large Language Model) acts as a central orchestrator. The model can "decide" which tools to use based on the query. These tools can include API calls, table lookups, or natural language searches. The system should contain asystem promptthat informs the LLM about the available tools.
* System Prompt Listing Tools:By creating a well-craftedsystem prompt, the LLM knows which tools are at its disposal. For instance, one tool may query an external API for event dates, another might look up standings in a database, and a third may involve searching a vector database for general text-based information. Theagentwill be responsible for calling the appropriate tool depending on the query.
* Agent Orchestration of Calls:The agent system is designed to execute a series of steps based on the incoming query. If a user asks for the next event date, the system will recognize this as a task that requires an API call. If the user asks about standings, the agent might query the appropriate table in the database. For text-based questions, it may call a search function over ingested data. The agent orchestrates this entire process, ensuring the LLM makes calls to the right resources dynamically.
* Generative AI Tools and Context:This is a standard architecture for integrating multiple functionalities into a system where each query requires different actions. The core design in option B is efficient because it keeps the system modular and dynamic by leveraging tools rather than overloading the LLM with static information in a system prompt (like option D).
* Why Other Options Are Less Suitable:
* A (RAG Architecture): While relevant, simply ingesting PDFs into a vector store only helps with text-based retrieval. It wouldn't help with API lookups or table queries.
* C (Conditional Logic with RAG/API/TABLE): Although this approach works, it relies heavily on manual text parsing and might introduce complexity when scaling the system.
* D (System Prompt with Event Dates and Standings): Hardcoding dates and table information into a system prompt isn't scalable. As the standings or events change, the system would need constant updating, making it inefficient.

By bundling multiple tools into a single agent-based system (as in option B), the Generative AI Engineer can best handle the diverse requirements of this system.

NEW QUESTION # 62
A Generative AI Engineer wants to build an LLM-based solution to help a restaurant improve its online customer experience with bookings by automatically handling common customer inquiries. The goal of the solution is to minimize escalations to human intervention and phone calls while maintaining a personalized interaction. To design the solution, the Generative AI Engineer needs to define the input data to the LLM and the task it should perform.
Which input/output pair will support their goal?

- A. Input: Customer reviews; Output: Classify review sentiment
- B. Input: Online chat logs; Output: Cancellation options
- C. Input: Online chat logs; Output: Group the chat logs by users, followed by summarizing each user's interactions
- D. Input: Online chat logs; Output: Buttons that represent choices for booking details

Answer: D

Explanation:
Context: The goal is to improve the online customer experience in a restaurant by handling common inquiries about bookings, minimizing escalations, and maintaining personalized interactions.
Explanation of Options:
* Option A: Grouping and summarizing chat logs by user could provide insights into customer interactions but does not directly address the task of handling booking inquiries or minimizing escalations.
* Option B: Using chat logs to generate interactive buttons for booking details directly supports the goal of facilitating online bookings, minimizing the need for human intervention by providing clear, interactive options for customers to self-serve.
* Option C: Classifying sentiment of customer reviews does not directly help with booking inquiries, although it might provide valuable feedback insights.
* Option D: Providing cancellation options is helpful but narrowly focuses on one aspect of the booking process and doesn't support the broader goal of handling common inquiries about bookings.
Option Bbest supports the goal of improving online interactions by using chat logs to generate actionable items for customers, helping them complete booking tasks efficiently and reducing the need for human intervention.

NEW QUESTION # 63
.....

- Databricks-Generative-AI-Engineer-Associate Test Result □ Databricks-Generative-AI-Engineer-Associate Test Result □ □ Test Databricks-Generative-AI-Engineer-Associate Pattern □ Download ⇛ Databricks-Generative-AI-Engineer-Associate ⇚ for free by simply entering ➥ www.getvalidtest.com □ website □Databricks-Generative-AI-Engineer-Associate Valid Dumps Demo
- Databricks-Generative-AI-Engineer-Associate Accurate Answers □ Databricks-Generative-AI-Engineer-Associate Valid Dumps Demo □ New Databricks-Generative-AI-Engineer-Associate Mock Test □ Download ▹ Databricks-Generative-AI-Engineer-Associate ◃ for free by simply entering { www.pdfvce.com } website □Latest Test Databricks-Generative-AI-Engineer-Associate Discount
- 100% Pass 2025 Databricks Databricks-Generative-AI-Engineer-Associate: High Pass-Rate Test Databricks Certified Generative AI Engineer Associate Simulator □ Open website 《 www.pass4leader.com 》 and search for □ Databricks-Generative-AI-Engineer-Associate □ for free download □Latest Databricks-Generative-AI-Engineer-Associate Exam Review
- Databricks Test Databricks-Generative-AI-Engineer-Associate Simulator: Databricks Certified Generative AI Engineer Associate - Pdfvce Sample Download Free □ Open □ www.pdfvce.com □ and search for □ Databricks-Generative-AI-Engineer-Associate □ to download exam materials for free □Databricks-Generative-AI-Engineer-Associate Valid Examcollection
- Use Databricks Databricks-Generative-AI-Engineer-Associate PDF Questions [2025]-Forget About Failure □ Search for □ Databricks-Generative-AI-Engineer-Associate □ on （ www.pass4test.com ） immediately to obtain a free download □ □Latest Test Databricks-Generative-AI-Engineer-Associate Discount
- Take a Leap Forward in Your Career by Earning Databricks Databricks-Generative-AI-Engineer-Associate □ Search for 「 Databricks-Generative-AI-Engineer-Associate 」 and download it for free on { www.pdfvce.com } website □ □Databricks-Generative-AI-Engineer-Associate Test Result
- Databricks-Generative-AI-Engineer-Associate Latest Exam Pass4sure □ Databricks-Generative-AI-Engineer-Associate Test Result □ Databricks-Generative-AI-Engineer-Associate Valid Examcollection □ Open ➤ www.examcollectionpass.com □ enter [Databricks-Generative-AI-Engineer-Associate] and obtain a free download □ □Databricks-Generative-AI-Engineer-Associate Exam Consultant
- myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, kareyed271.develop-blog.com, tedcole945.blogunok.com, motionentrance.edu.np, 1.gdsfjd.net, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, einfachalles.at, Disposable vapes

BTW, DOWNLOAD part of Prep4sures Databricks-Generative-AI-Engineer-Associate dumps from Cloud Storage: https://drive.google.com/open?id=1jmHHJjJxDYYr1m0pE-9Vzp3rZ88VHrLd