

Designing and Implementing a Microsoft Azure AI Solution

Microsoft AI-102

Version Demo

Total Demo Questions: 44

Total Premium Questions: 447

Buy Premium PDF

<https://passqueen.com>

support@passqueen.com

Topic Break Down

Topic	No. of Questions
Topic 1, Plan and manage an Azure AI solution	60
Topic 2, Implement generative AI solutions	32
Topic 3, Implement an agentic solution	18
Topic 4, Implement computer vision solutions	65
Topic 5, Implement natural language processing solutions	169
Topic 6, Implement knowledge mining and information extraction solutions	57
Topic 7, Mix Questions	46
Total	447

QUESTION NO: 1

You have an Azure DevOps pipeline named Pipeline1 that is used to deploy an app. Pipeline1 includes a step that will create an Azure AI services account.

You need to add a step to Pipeline1 that will identify the created Azure AI services account. The solution must minimize development effort.

Which Azure Command-Line interface (CLI) command should you run?

- A. az resource link.
- B. az account list.
- C. az cognitiveservices account network-rule.
- D. az cognitiveservices account show.

ANSWER: D

Explanation:

`az cognitiveservices account show` is the appropriate command because it retrieves the properties of one specified Azure AI services (Azure Cognitive Services) account. In an Azure DevOps pipeline, the account creation step can provide, or the pipeline can retain, the resource group and account name. Supplying those values to `az cognitiveservices account show --name <account-name> --resource-group <resource-group-name>` lets the subsequent step confirm that the intended resource was created and obtain its resource metadata in a structured response. The returned JSON includes useful values such as the resource ID, name, location, kind, SKU, and endpoint-related properties, which can be consumed by later pipeline tasks or logged for deployment verification.

This is a direct Azure CLI operation for retrieving an existing Azure AI services account, so it requires no custom resource-discovery script or additional API integration. It is therefore well suited to the requirement to identify the account with minimal development effort. The command is part of the Azure CLI Cognitive Services account command group, documented by Microsoft at [az cognitiveservices account show](#). Azure AI services accounts are the Azure resources used to access supported AI capabilities, as described in the [Azure AI services overview](#).

QUESTION NO: 2

You are building a Chatbot by using the Microsoft Bot Framework SDK. The bot will be used to accept food orders from customers and allow the customers to customize each food item. You need to configure the bot to ask the user for additional input based on the type of item ordered. The solution must minimize development effort. Which two types of dialogs should you use? Each correct answer presents part of the solution. NOTE: Each correct selection is worth one point.

- A. adaptive.
- B. action.
- C. waterfall.
- D. prompt.
- E. input.

ANSWER: A E

Explanation:

adaptive. and **input.** are appropriate because an adaptive dialog is designed to manage conversational flow dynamically according to the current state, recognized intent, and user responses. In a food-ordering bot, this lets the conversation adapt naturally after the customer selects an item. For example, an order for a pizza can trigger questions about size and toppings, while a drink can trigger questions about flavor and quantity, without requiring a rigid, linear sequence for every possible order path.

Input dialogs provide the reusable mechanisms for collecting the required details from the customer. Adaptive Dialogs support input actions such as text, number, choice, confirmation, date/time, attachment, and OAuth input. The bot can therefore use an appropriate input type for each customization: a choice for pizza size, a multiple-choice selection for toppings, a number for quantity, and a confirmation before adding the item to the order. Combining adaptive flow control with declarative input collection reduces custom dialog-management code and supports conditional questions cleanly. See the [Microsoft documentation on adaptive dialogs](#) and the [documentation on adaptive dialog input](#).

QUESTION NO: 3

You have an Azure Cognitive Services model named Model that identifies the intent of text input. You develop an app in C# named App1.

You need to configure App1 to use Model1. Which package should you add to App1?

- A. `Azure.AI.Language.Conversations`.
- B. `SpeechServicesToolkit`.
- C. `Universal.Microsoft.CognitiveServices.Speech`.
- D. `Xamarin.Cognitive.Speech`.

ANSWER: A

Explanation:

Azure.AI.Language.Conversations is the appropriate .NET package for a C# application that sends text to an Azure AI Language conversational language understanding model and receives the detected intent. This SDK contains the conversation-analysis clients used to authenticate to an Azure AI Language resource and submit utterances for prediction. A request can specify the deployed project and deployment, then return a result containing the top intent, its confidence score, and any extracted entities. This supports the stated requirement because the application is processing text input for intent identification, rather than performing speech transcription or synthesis.

In current Azure AI Language terminology, conversational language understanding is the service capability for building custom natural-language-understanding projects that classify user utterances into intents. The .NET client library documentation shows installation of the `Azure.AI.Language.Conversations` package and use of its conversation-authoring and conversation-analysis functionality. Microsoft's conversational language understanding documentation also describes intent classification as a core prediction output for deployed projects. See the [Azure.AI.Language.Conversations .NET client library documentation](#) and the [Conversational Language Understanding overview](#).

QUESTION NO: 4 - (DRAG DROP)

DRAG DROP

You plan to build a chatbot to support task tracking.

You create a Language Understanding service named lu1.

You need to build a Language Understanding model to integrate into the chatbot. The solution must minimize development time to build the model.

Which four actions should you perform in sequence? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order. (Choose four.)

Select and Place:

Actions

Answer Area

Train the application.

Publish the application.

Add a new application.

Add example utterances.

Add the prebuilt domain ToDo.

ANSWER:

Actions

Answer Area

Train the application.

Publish the application.

Add a new application.

Add example utterances.

Add the prebuilt domain ToDo.

Add a new application.

Add the prebuilt domain ToDo.

Train the application.

Publish the application.

Explanation:

The correct sequence begins by creating a new Language Understanding application. The application is the model container where intents, entities, utterances, settings, versions, and deployment configuration are managed. Once that application exists, adding the prebuilt ToDo domain is the fastest way to establish task-tracking language capabilities. A prebuilt domain provides an existing model structure and task-related language patterns, so the solution can begin from domain-specific content rather than requiring a full model to be designed and populated manually.

After the ToDo domain has been added, the application must be trained. Training processes the model's intents, entities, and included example language so that the Language Understanding service can predict the user's intent and identify relevant task information. Training is required before the current model version can be deployed for runtime use.

The final step is publishing the application. Publishing makes the trained version available through a prediction endpoint, which is what the chatbot uses to submit user messages and receive recognized intents and entities. This creates the necessary path from the chatbot to the deployed natural-language model. The workflow therefore efficiently uses the built-in task-management domain, then prepares and exposes the resulting model for integration. Microsoft's documentation on [prebuilt domains](#) describes how these domains provide predefined intents and entities for common scenarios, and the [application publishing guidance](#) explains deployment to a prediction endpoint.

QUESTION NO: 5

You have a SQL query that combines customer data and order data. The query includes calculated columns. You need to create a database object that would allow other users to rerun the same SQL query. What should you create?

A. an Index

B. a view

C. a scalar function

D. a table

ANSWER: B

Explanation:

A view is the appropriate database object because it stores a named `SELECT` query definition that users can query repeatedly. The view can encapsulate the join between customer and order data and expose calculated columns defined by expressions in its select list. Rather than distributing and maintaining copies of the original SQL statement, users can retrieve the current results through a simple statement such as `SELECT * FROM dbo.CustomerOrdersView`. This centralizes the query logic, provides a consistent schema-like interface over the underlying data, and allows the view definition to be updated in one location if the reusable logic later changes. Views are virtual tables: SQL Server stores the query definition rather than a separate ordinary table containing duplicated result data. Subject to the permissions assigned by the database administrator, a view can also be used to provide consumers access to the required result set without requiring them to work directly with the underlying table structure. Microsoft describes views as virtual tables whose contents are defined by a query, including queries that use joins and expressions. See [Views in SQL Server](#) and [CREATE VIEW \(Transact-SQL\)](#).

QUESTION NO: 6

You have an Azure OpenAI model named All.

You are building a web app named App1 by using the Azure OpenAI SDK. You need to configure App1 to connect to All. What information must you provide?

A. the endpoint, key, and model name.

B. the deployment name, endpoint, and key.

C. the endpoint, key, and model type.

D. the deployment name, key, and model name.

ANSWER: B

Explanation:

The deployment name, endpoint, and key are required to configure an application that uses the Azure OpenAI SDK with key-based authentication. Azure OpenAI models are made available for inference by creating a deployment in an Azure OpenAI resource. The application sends requests to that deployment, so the deployment name identifies the model configuration that should process the request. The Azure OpenAI resource endpoint supplies the network address for the specific Azure resource hosting the deployment. It is typically provided in the form `https://<resource-name>.openai.azure.com/`. An API key authenticates the application to that resource when using API-key authentication. In SDK code, these values are commonly used to create the Azure OpenAI client with the endpoint and credential, while the deployment name is supplied as the model or deployment target in a request. This separation allows an organization to change the underlying base model or deployment settings while applications continue to use the deployment identifier. Microsoft's quickstarts demonstrate configuring the endpoint, credential, and deployment name for Azure OpenAI inference requests. See [Create and deploy an Azure OpenAI Service resource](#) and [Switch between OpenAI and Azure OpenAI endpoints](#).

QUESTION NO: 7

You plan to create an index for an Azure Cognitive Search service by using the Azure portal. The Cognitive Search service will connect to an Azure SQL database.

The Azure SQL database contains a table named UserMessages. Each row in UserMessages has a field named MessageCopy that contains the text of social media messages sent by a user.

Users will perform full text searches against the MessageCopy field, and the values of the field will be shown to the users.

You need to configure the properties of the index for the MessageCopy field to support the solution. Which attributes should you enable for the field?

- A. Searchable and Retrievable.
- B. Sortable and Retrievable.
- C. Searchable and Facetable.
- D. Filterable and Retrievable.

ANSWER: A

Explanation:

Searchable and Retrievable is the required configuration for **MessageCopy**. A searchable field is added to the full-text search index. For string fields, Azure AI Search analyzes the text into terms, enabling users to submit full-text queries that match words and phrases within social media messages. This is the appropriate behavior for free-form message content rather than treating the entire message as a single exact value.

The field must also be retrievable because the application needs to display the matching message text to users. Retrievable fields are included in the documents returned by search queries, subject to any query-time `$select` specification. This lets the application present the **MessageCopy** value in the results alongside any other selected fields.

These attributes can be configured when defining the index field in the Azure portal or in an index schema. Azure AI Search documentation describes searchable fields as participating in full-text search and retrievable fields as being returned in search results. See [Azure AI Search field attributes](#) and [Create a simple query in Azure AI Search](#).

QUESTION NO: 8

You have an Azure subscription that contains an Azure OpenAI resource named AH and an Azure AI Content Safety resource named CS1.

You build a chatbot that uses AI to provide generative answers to specific questions and CS1 to check input and output for objectionable content.

You need to optimize the content filter configurations by running tests on sample questions. Solution: From Content Safety Studio, you use the Protected material detection feature to run the tests.

Does this meet the requirement?

- A. Yes.
- B. No.

ANSWER: B

Explanation:

No. Protected material detection is intended to identify text or code that matches known protected content, such as copyrighted song lyrics, articles, or source-code repositories. It supports intellectual-property and copyright risk controls; it does not evaluate whether chatbot prompts and completions contain harmful or objectionable content at the configured moderation thresholds.

Optimizing a chatbot's content filtering requires testing representative input prompts and generated responses against the harmful-content categories used by Azure AI Content Safety and Azure OpenAI content filters. These categories include hate and fairness, sexual content, violence, and self-harm. Test results provide category classifications and severity levels, enabling the team to assess whether the selected thresholds appropriately block, annotate, or allow content for the intended application scenario. This is the information needed to calibrate filters using sample questions and responses.

Content Safety Studio can be used to explore and test content-safety capabilities, but protected-material results do not provide the harmful-content category and severity evaluation required by the stated objective. Microsoft describes the

available harm categories and severity levels at [Harm categories and severity levels](#), and describes protected-material detection separately in the [Protected material detection documentation](#).

QUESTION NO: 9

You build a custom Form Recognizer model.

You receive sample files to use for training the model as shown in the following table.

Name	Type	Size
File1	PDF	20 MB
File2	MP4	100 MB
File3	JPG	20 MB
File4	PDF	100 MB
File5	GIF	1 MB
File6	JPG	40 MB

Which three files can you use to train the model? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. File1
- B. File2
- C. File3
- D. File4
- E. File5
- F. File6

ANSWER: A C F

Explanation:

File1, File3, and File6 can be used because they meet the input-file requirements for training a custom Azure AI Document Intelligence model, formerly called Form Recognizer. Custom model training uses document and image formats supported by the service, including PDF and common image formats such as JPEG, PNG, and TIFF. The files represented by these selections have supported formats and satisfy the applicable training constraints shown in the table, so each can be supplied as an individual training document.

Training files are used to identify the structure and fields that the custom model must learn. Therefore, each source document must be in a supported document type and within the applicable service limits for items such as file size, image dimensions, and, for PDFs, page count. Microsoft recommends using representative samples that reflect the documents the model will process in production. For custom extraction models, training documents are stored in a container that the Document Intelligence resource can access, and the service builds the model from those valid samples.

See the [Microsoft guidance for training custom models](#) and the [Document Intelligence input requirements](#) for supported formats and limits.

QUESTION NO: 10

You have a chatbot that was built by using the Microsoft Bot Framework.

You need to debug the chatbot endpoint remotely.

Which two tools should you install on a local computer? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Fiddler
- B. Bot Framework Composer
- C. Bot Framework Emulator
- D. Bot Framework CLI
- E. ngrok
- F. nginx

ANSWER: C E

Explanation:

Bot Framework Emulator and **ngrok** together support the standard local-and-remote debugging workflow for a Bot Framework bot. Bot Framework Emulator is a desktop application that enables a developer to connect to a bot endpoint, send messages as activities, examine the conversation, inspect activity payloads, and troubleshoot authentication and endpoint behavior. It provides the Bot Framework-aware client experience needed to test the bot while its code runs under a local debugger.

ngrok provides a secure public tunnel to an HTTP endpoint running on the developer's local computer. This is required when a remote Bot Framework service or channel must deliver requests to a bot that is being debugged locally. After starting the bot locally, the developer runs ngrok for the bot's listening port and configures the resulting public HTTPS forwarding URL as the messaging endpoint. Incoming requests can then reach the local process, where breakpoints and diagnostic tools are available.

Microsoft documents using the Emulator for bot debugging and using ngrok to make a locally running bot reachable from external services. See [Debug bots with the Bot Framework Emulator](#) and [Debug a bot using ngrok](#).

QUESTION NO: 11

You are developing a new sales system that will process the video and text from a public-facing website.

You plan to monitor the sales system to ensure that it provides equitable results regardless of the user's location or background.

Which two responsible AI principles provide guidance to meet the monitoring requirements? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. transparency
- B. fairness
- C. inclusiveness
- D. reliability and safety
- E. privacy and security

ANSWER: B C

Explanation:

fairness and **inclusiveness** provide the responsible AI guidance needed to monitor whether the sales system delivers equitable results across users' locations and backgrounds. Fairness requires AI systems to allocate opportunities, resources, and outcomes equitably, and to avoid creating or reinforcing unfair bias. For this system, that means monitoring model outputs and business outcomes for meaningful disparities among relevant groups, such as people in different geographic regions, language communities, or demographic cohorts. Monitoring should identify differences in accuracy, error rates, recommendations, and access to sales-related opportunities so that issues can be investigated and mitigated.

Inclusiveness guides designers to ensure that AI systems empower and engage people with diverse characteristics, circumstances, and needs. Because this public-facing solution processes both text and video, inclusive monitoring should consider whether people using different languages, dialects, accents, connectivity conditions, devices, or accessibility accommodations receive comparably effective service. Together, these principles support ongoing evaluation of representation in data, performance across user groups, and the real-world impact of the system, rather than evaluating only overall aggregate performance. Microsoft describes these principles in its [Responsible AI principles guidance](#) and its [Responsible use of Azure AI services overview](#).

QUESTION NO: 12

You have an Azure OpenAI model.

You have 500 prompt-completion pairs that will be used as training data to fine-tune the model. You need to prepare the training data.

Which format should you use for the training data file?

- A. XML.
- B. JSONL.
- C. CSV.
- D. TSV.

ANSWER: B

Explanation:

JSONL is the required format for an Azure OpenAI fine-tuning training file. JSON Lines is a UTF-8 text format in which each line is a separate, valid JSON object representing one training example. Therefore, the 500 prompt-completion pairs are represented as 500 individual JSON records, one record per line. This line-oriented structure enables the fine-tuning service to process examples independently and supports scalable dataset handling.

For prompt-completion fine-tuning data, an example record uses properties such as `prompt` and `completion`, with the values containing the input and desired model output. The precise schema can vary according to the model and fine-tuning method; for example, supported chat-model fine-tuning uses a `messages` array. In all cases, the uploaded training dataset itself must be prepared as a JSONL file rather than as a tabular or markup file. The file should also be validated before a fine-tuning job is created, ensuring that each line contains well-formed JSON and conforms to the selected model's required training-example structure.

Microsoft documents the JSONL requirement and the supported example schemas in [Azure OpenAI fine-tuning](#) and provides further dataset guidance in [fine-tuning considerations](#).

QUESTION NO: 13

You are building a retail kiosk system that will use a custom neural voice. You acquire audio samples and consent from the voice talent. You need to create a voice talent profile. What should you upload to the profile?

- A. a five-minute wav or mp3 file of the voice talent describing the kiosk system.
- B. a five-minute .flac audio file and the associated transcript as a w file.
- C. a .wav or mp3 file of the voice talent consenting to the creation of a synthetic version of their voice.

D. a .zip file that contains 10-second .wav files and the associated transcripts as .txt files.

ANSWER: C

Explanation:

A **.wav or mp3 file of the voice talent consenting to the creation of a synthetic version of their voice** is required because a voice talent profile is the consent and identity record for a person whose voice will be used to create a Custom Neural Voice. The profile is created by providing the voice talent's recorded consent statement. This statement documents that the talent has knowingly authorized Microsoft and the customer to create and use a synthetic voice based on that person's voice. It is a governance and responsible-AI requirement that must be completed before the custom voice can be trained and deployed.

The consent recording is distinct from the speech recordings used as training data. After the talent profile is established and consent is verified, the collected audio samples and their matching transcripts can be organized into a training dataset for creating the custom neural voice model. Azure AI Speech supports consent-statement recordings in WAV or MP3 formats for this profile workflow. Microsoft documents the consent process and voice talent profile as required elements of Custom Neural Voice, including the need for explicit talent authorization. See [Custom Neural Voice documentation](#) and [Custom Voice guidance](#).

QUESTION NO: 14

You are building an app that will share user images.

You need to configure the app to perform the following actions when a user uploads an image: Categorize the image as either a photograph or a drawing.

Generate a caption for the image.

The solution must minimize development effort.

Which two services should you include in the solution? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. object detection in Computer Vision.
- B. content tags in Computer Vision.
- C. image descriptions in Computer Vision.
- D. image type detection in Computer Vision.
- E. image classification in Custom Vision.

ANSWER: C D

Explanation:

image descriptions in Computer Vision. is appropriate because Azure AI Vision provides prebuilt image-description capability that generates a human-readable natural-language caption for an image and includes a confidence score. This meets the caption requirement without collecting training images, labeling data, training a model, or developing custom caption-generation logic. A caption is intended to summarize the visible image content in a short phrase or sentence, which directly supports an image-sharing application.

image type detection in Computer Vision. is also appropriate because the prebuilt image-analysis response can identify image characteristics, including whether content is photographic and whether it is likely clip art or a line drawing. The application can use those returned image-type signals to categorize an uploaded image as a photograph or drawing. Because both capabilities are provided by the same Azure AI Vision service, the solution requires only service provisioning and calls to its image-analysis API, minimizing implementation and operational effort. Microsoft describes captioning as an Image Analysis capability in its [Image captioning documentation](#). The service's broader supported visual-analysis features are documented in the [Azure AI Vision Image Analysis overview](#).

QUESTION NO: 15

Note: This question is part of a series of questions that present the same scenario. Each question in the series contains a unique solution that might meet the stated goals. Some question sets might have more than one correct solution, while others might not have a correct solution.

After you answer a question in this section, you will NOT be able to return to it. As a result, these questions will not appear in the review screen.

You have a chatbot that uses question answering in Azure Cognitive Service for Language Users report that the responses of the chatbot lack formality when answering spurious questions You need to ensure that the chatbot provides formal responses to spurious questions.

Solution: From Language Studio, you change the chitchat source to qna_chitchat_professional.tsv. and then retrain and republish the model.

Does this meet the goal?

A. Yes.

B. No.

ANSWER: A

Explanation:

Yes. The `qna_chitchat_professional.tsv` file is the professional personality chit-chat dataset for Question Answering. Chit-chat is designed to provide responses when a user asks conversational, irrelevant, or otherwise unanticipated questions that are not covered by the main knowledge source. The professional dataset supplies question-and-answer pairs phrased in a more formal tone than the default or casual personality alternatives. Therefore, changing the chit-chat source to this dataset directly addresses the requirement that responses to spurious questions be formal.

After changing the chit-chat personality or source, retraining is required so that the updated chit-chat Q&A pairs are incorporated into the project's model. Republishing then makes that newly trained version available to the chatbot endpoint. These steps ensure that the chatbot uses the professional-formality responses at runtime for matching chit-chat queries.

Microsoft documents that Question Answering supports predefined chit-chat datasets, including a professional personality, and that the selected chit-chat content is added to the project and published with it. See [Add chit-chat to a knowledge base](#) and [Question Answering documentation](#).

QUESTION NO: 16

A customer uses Azure Cognitive Search.

The customer plans to enable a server-side encryption and use customer-managed keys (CMK) stored in Azure.

What are three implications of the planned change? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

A. The index size will increase.

B. Query times will increase.

C. A self-signed X.509 certificate is required.

D. The index size will decrease.

E. Query times will decrease.

F. Azure Key Vault is required.

ANSWER: A B F

Explanation:

Customer-managed keys add a customer-controlled encryption layer to Azure AI Search encryption at rest. The service requires access to a customer-owned encryption key stored in Azure Key Vault; therefore, Azure Key Vault is required for this configuration. The Azure AI Search service must also be granted access to use the key, normally through a managed identity and the appropriate Key Vault permissions. This dependency means that availability and access to the key are essential to continued access to encrypted search-service content.

Using customer-managed keys has documented operational and capacity implications. Azure AI Search notes that encryption with customer-managed keys increases index size because additional encrypted data and metadata must be maintained. Capacity planning should account for this growth, particularly for services with large indexes or limited storage headroom. Customer-managed-key encryption can also increase query times. Encryption-related processing can add latency, so workloads with strict query-latency requirements should be tested and monitored after CMK is enabled. These implications are specific tradeoffs of choosing customer-managed encryption control rather than the default Microsoft-managed keys. See [Configure customer-managed keys in Azure AI Search](#) and [Azure AI Search data protection overview](#).

QUESTION NO: 17

You develop a Conversational Language Understanding model by using Language Studio

During testing, users receive incorrect responses to requests that do NOT relate to the capabilities of the model.

You need to ensure that the model identifies spurious requests. What should you do?

- A. Enable active learning.
- B. Add examples to the custom intents.
- C. Add examples to the None intent.
- D. Add entities.

ANSWER: C**Explanation:**

Add examples to the None intent. In a Conversational Language Understanding (CLU) project, the None intent is designed for utterances that do not belong to any of the application's supported intents. Training it with representative spurious, unrelated, or out-of-domain requests teaches the classifier the boundary of the solution's capabilities. For example, if a project handles employee benefits questions, the None intent can include unrelated utterances such as requests for weather forecasts, restaurant recommendations, or technical support unrelated to benefits.

CLU evaluates a user utterance against the intents learned during training. A well-populated None intent helps the model recognize that an incoming request should not be mapped to an in-scope business action. The application can then provide a controlled fallback response, such as informing the user what it can help with or handing the conversation to another system. Microsoft guidance recommends including diverse None utterances that represent the real out-of-scope requests users are likely to submit, then testing and refining those examples as new patterns emerge. This improves reliable intent classification and reduces inappropriate responses for unsupported requests. See [None intent in CLU](#) and [Build and train a CLU model](#).

QUESTION NO: 18

You have an Azure subscription that contains an Azure OpenAI resource named OpenAI1 and a user named User1.

You need to ensure that User1 can upload datasets to OpenAI1 and finetune the existing models. The solution must follow the principle of least privilege.

Which role should you assign to User1?

- A. Cognitive Services Contributor.
- B. Contributor.

- C. Cognitive Services OpenAI User.
- D. Cognitive Services OpenAI Contributor.

ANSWER: D

Explanation:

Cognitive Services OpenAI Contributor. is the least-privilege built-in Azure role for this requirement. Fine-tuning requires access to Azure OpenAI data-plane capabilities beyond merely calling deployed models: the user must be able to upload and manage training files, create and monitor fine-tuning jobs, and work with the resulting customized models. The Cognitive Services OpenAI Contributor role is specifically designed to grant comprehensive Azure OpenAI capabilities, including fine-tuning and model deployment operations, while not granting permission to manage the Azure OpenAI resource itself through Azure Resource Manager. This makes it appropriate when User1 needs to perform model-customization work but does not need broad administrative control over the subscription or the underlying Azure AI service resource. Assign the role at the narrowest applicable scope, normally directly on the OpenAI1 resource, to further limit User1's permissions. Microsoft's Azure OpenAI role-based access control guidance identifies this role as providing full Azure OpenAI access, including the ability to fine-tune and deploy models. See [Azure OpenAI RBAC guidance](#) and the [Cognitive Services OpenAI Contributor built-in role definition](#).

QUESTION NO: 19

You have an Azure Cognitive Search solution and an enrichment pipeline that performs Sentiment Analysis on social media posts.

You need to define a knowledge store that will include the social media posts and the Sentiment Analysis results.

Which two fields should you include in the definition? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. storageContainer.
- B. tables.
- C. storageConnectionString.
- D. files.
- E. objects.

ANSWER: C E

Explanation:

A knowledge store is configured in an Azure AI Search skillset to persist content produced by an enrichment pipeline in Azure Storage. **storageConnectionString.** is required because it provides the connection information that Azure AI Search uses to access the target Azure Storage account where enriched data will be written. The connection string is a top-level property of the knowledge store definition.

objects. is the appropriate projection field for retaining the social media post content together with its sentiment enrichment results as JSON documents in Azure Blob Storage. An object projection maps nodes from the enriched document tree into a JSON representation, making it suitable for storing source-level content and skill outputs such as sentiment labels and confidence scores in a form that downstream applications can consume. The projection also specifies its destination container and the source context to project.

With a storage connection and an object projection, the skillset can persist the enriched social-media records and their sentiment-analysis data outside the search index. For details, see [Knowledge store in Azure AI Search](#) and [Knowledge store projections](#).

QUESTION NO: 20

You need to implement a table projection to generate a physical expression of an Azure Cognitive Search index.

Which three properties should you specify in the skillset definition JSON configuration table node? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. tableName
- B. generatedKeyName
- C. dataSource
- D. dataSourceConnection
- E. source

ANSWER: A B E

Explanation:

A knowledge store table projection writes shaped enrichment results to Azure Table Storage as rows. The table projection definition therefore requires `tableName`, `generatedKeyName`, and `source`. `tableName` identifies the Azure Table Storage table that receives the projected records. It provides the physical destination for the tabular representation of the enriched content.

`generatedKeyName` specifies the name of a generated key field in the projected table. Azure AI Search uses this field to produce a unique identifier for projection rows, which supports the relationship and identity requirements of table-based knowledge-store output.

`source` identifies the node in the enrichment tree from which the projection obtains its data. In common implementations, this is a path to an object produced by a Shaper skill. The Shaper skill organizes the desired enriched fields into a structure that can be written as table columns. Together, these settings identify the destination table, establish row identity, and connect the projection to the enriched source data. Microsoft documents these required table projection properties and shows them in knowledge-store projection examples. See [Knowledge store projections overview](#) and [Knowledge store projection examples](#).

QUESTION NO: 21

You are building a solution that uses Azure AI Language to process customer emails.

You need to identify people, organizations, locations, and monetary amounts in the email text.

Which two entity-recognition capabilities should you use? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. Named entity recognition.
- B. Personally identifiable information detection.
- C. Key phrase extraction.
- D. Language detection.
- E. Text summarization.

ANSWER: A B

Explanation:

Named entity recognition is appropriate because it identifies and categorizes entities in unstructured text, including people, organizations, locations, dates, and quantities. The returned entities include category and confidence information that the application can use for downstream processing.

Personally identifiable information detection is also appropriate when the application must identify sensitive entities such as names, phone numbers, email addresses, addresses, and financial account-related information. PII detection can identify sensitive data and supports redaction scenarios.

Key phrase extraction returns important concepts or phrases but does not provide the entity categories needed for this requirement. Language detection identifies the language of text rather than extracting entities. See [Named entity recognition in Azure AI Language](#) and [PII detection in Azure AI Language](#).

QUESTION NO: 22

You have an Azure subscription.

You are building a social media app that will enable users to share images.

You need to ensure that inappropriate content uploaded by the users is blocked. The solution must minimize development effort

What are two tools that you can use? Each correct answer presents a complete solution NOTE: Each correct selection is worth one point.

- A. Microsoft Defender for Cloud Apps.
- B. Azure AI Custom Vision.
- C. Azure AI Vision.
- D. Azure AI Content Safety.
- E. Azure AI Document Intelligence.

ANSWER: C D

Explanation:

Azure AI Content Safety is a complete, managed image-moderation solution for this requirement. Its Image API evaluates submitted images for harmful-content categories, including hate, self-harm, sexual content, and violence, and returns severity levels. The application can submit each uploaded image before publishing it, compare the returned severity to a chosen threshold, and block the upload when the threshold is met. This provides a direct moderation workflow without building, training, or hosting a custom machine-learning model. See the [Azure AI Content Safety overview](#).

Azure AI Vision is also suitable because its prebuilt image-analysis capabilities include adult and racy-content detection. A social-media application can call the service for an uploaded image and use its adult/racy confidence results to apply a blocking policy before making the image available to other users. Because the capability is provided by an Azure managed service and accessed through REST APIs or SDKs, it minimizes implementation effort while providing a complete automated screening path for this type of inappropriate content. Microsoft documents this capability in [Detect adult content in images with Azure AI Vision](#).

QUESTION NO: 23

You are developing a method for an application that uses the Translator API.

The method will receive the content of a webpage, and then translate the content into Greek (el). The result will also contain a transliteration that uses the Roman alphabet.

You need to create the URI for the call to the Translator API.

You have the following URI.

<https://api.cognitive.microsofttranslator.com/translate?api-version=3.0>

Which three additional query parameters should you include in the URI? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. `toScript=Cyrl`
- B. `from=el`
- C. `textType=html`
- D. `to=el`
- E. `textType=plain`
- F. `toScript=Latn`

ANSWER: C D F

Explanation:

The required query parameters are **`textType=html`**, **`to=el`**, and **`toScript=Latn`**. The Translator Translate operation requires at least one `to` parameter, and `el` is the language code for Greek. Therefore, `to=el` specifies that Greek is the target language for the translated webpage content.

Because the input is webpage content, `textType=html` instructs Translator to process the text as HTML rather than ordinary plain text. This enables the service to handle supported HTML markup appropriately while translating the textual content.

The Translate operation supports requesting a target script with `toScript`. Using `toScript=Latn` requests the Latin script, which is the Roman alphabet specified in the requirement. Translator can return transliteration information when a target script is requested and the language/script combination supports it. Greek supports transliteration into Latin script, so this parameter satisfies the requirement for a Roman-alphabet result.

For parameter definitions and request examples, see the [Translator Translate method reference](#). For supported language and script transliteration combinations, see [Translator language support](#).

QUESTION NO: 24

You use the Custom Vision service to build a classifier.

After training is complete, you need to evaluate the classifier.

Which two metrics are available for review? Each correct answer presents a complete solution.

NOTE: Each correct selection is worth one point.

- A. recall
- B. F-score
- C. weighted accuracy
- D. precision
- E. area under the curve (AUC)

ANSWER: A D

Explanation:

Custom Vision classification training results provide **recall** and **precision** as evaluation metrics for each tag. These metrics are calculated from the evaluation images that Custom Vision uses to assess the trained iteration, and they help determine whether the classifier is suitable for deployment or requires additional, better-balanced training data.

Recall measures the proportion of actual instances of a tag that the classifier successfully identifies. It is useful when the solution must avoid missing items that belong to a particular class. For example, strong recall for a defect tag means the model identifies a high proportion of the images that truly contain that defect.

Precision measures how often predictions for a tag are correct. It is useful when predictions must be trustworthy and when falsely assigning a tag has a meaningful operational cost. Reviewing precision and recall together gives a practical view of classifier behavior per tag and supports selecting an appropriate probability threshold for use in testing or an application workflow. Microsoft describes reviewing these per-tag performance measures after training a classifier in the Custom Vision portal. See [Build a classifier with Custom Vision](#) and [Test your Custom Vision model](#).

QUESTION NO: 25

You have a blog that allows users to append feedback comments. Some of the feedback comments contain harmful content that includes discriminatory language.

You need to create a prototype of a solution that will detect the harmful content. The solution must minimize development effort.

Which two actions should you perform? Each correct answer presents part of the solution. NOTE: Each correct selection is worth one point.

- A. Sign in to Content Safety Studio and select Moderate text content.
- B. From the Azure portal, create an Azure AI Content Safety resource.
- C. From the Azure portal, create an Azure OpenAI resource.
- D. Sign in to Azure AI Foundry and select Safety + security.
- E. Sign in to Content Safety Studio and select Protected material detection for text.

ANSWER: A B

Explanation:

Creating an **Azure AI Content Safety resource** provides the Azure resource required to access Content Safety capabilities. The service is designed to evaluate text for harmful content categories, including hate-related content, which is relevant to discriminatory language in blog comments. The resource supplies the endpoint and authentication credentials needed to use the service in a prototype and later integrate it into the blog's comment-submission workflow.

Signing in to **Content Safety Studio** and selecting **Moderate text content** is also appropriate because it enables rapid, low-code validation of the text-moderation capability. A developer can submit representative feedback comments, inspect the detected harm categories and severity levels, and determine suitable thresholds before writing application code. This directly supports the requirement to minimize development effort while confirming that the service identifies the harmful language the blog must handle. Azure AI Content Safety supports text analysis for categories such as hate, self-harm, sexual content, and violence, and its moderation experience is intended for evaluating content-safety results during development. For implementation details, see [Azure AI Content Safety overview](#) and [Analyze text with Azure AI Content Safety](#).

QUESTION NO: 26

You have a local folder that contains the files shown in the following table.

Name	Format	Length (mins)	Size (MB)
File1	WMV	34	400
File2	AVI	90	1,200
File3	MOV	300	980
File4	MP4	80	1,800

You need to analyze the files by using Azure AI Video Indexer. Which files can you upload to the Video Indexer website?

- A. File1, File2, and File4 only.
- B. File1 and File2 only.
- C. File1, File2, and File3 only.
- D. File1, File2, File3, and File4.
- E. File1 and File3 only.

ANSWER: E

Explanation:

File1 and File3 only. is correct because Azure AI Video Indexer accepts supported media files submitted through its web portal for indexing and AI analysis. Video Indexer processes audiovisual content and audio content to generate insights such as transcripts, translations, detected speakers, keyframes, scenes, labels, OCR results from video frames, and other media intelligence. The files represented by File1 and File3 in the table use media formats that are supported for upload, so they can be selected from the local folder and submitted to the Video Indexer website.

Supported upload eligibility is determined by the media file's container and encoding characteristics, not merely by the fact that a file is stored locally. When a supported file is uploaded, Video Indexer ingests it, creates an indexed video asset, and makes the resulting insights available after processing. This allows the service to analyze either video with audio or audio-only content where the format is supported. Microsoft documents the supported media types and the upload workflow for Video Indexer; these are the authoritative sources for determining whether a local file can be uploaded and indexed. See [Azure AI Video Indexer overview](#) and [Upload and index videos in Azure AI Video Indexer](#).

QUESTION NO: 27

You successfully run the following HTTP request.

```
POST https://management.azure.com/subscriptions/18c51a87-3a69-47a8-aedc-
a54745f708a1/resourceGroups/RG1/providers/Microsoft.CognitiveServices/accounts/
contosol/regenerateKey?api-version=2017-04-18 Body>{"keyName": "Key2"}
```

What is the result of the request?

- A. A key for Azure Cognitive Services was generated in Azure Key Vault.
- B. A new query key was generated.
- C. The primary subscription key and the secondary subscription key were rotated.
- D. The secondary subscription key was reset.

ANSWER: D

Explanation:

The secondary subscription key was reset. The request targets the Azure Resource Manager `regenerateKey` operation for the Cognitive Services account named `contosol`. Its request body explicitly supplies `{"keyName": "Key2"}`, so Azure regenerates the account key identified as Key2. Cognitive Services accounts expose two access keys to support safe key

rotation. One key can remain active in applications while the other is regenerated; after clients are updated to use the new value, the remaining key can be regenerated in a subsequent rotation step.

This is a management-plane request because it is sent to `management.azure.com` and identifies the target resource through its subscription, resource group, provider namespace, and account name. A successful response returns the account key values, including the newly generated value for the requested key. Azure AI services documentation describes using the two keys for rotation without interrupting authenticated client traffic, while the REST API definition identifies `keyName` as the key to regenerate. See [Accounts - Regenerate Key REST API](#) and [Authenticate requests to Azure AI services](#).

QUESTION NO: 28

You are building an app that will use the Azure AI Speech service.

You need to ensure that the app can authenticate to the service by using a Microsoft Entra ID token. Which two action should you perform? Each answer part of the solution.

NOTE: Each correct selection is worth one point.

- A. Create a Conditional Access.
- B. Create a private endpoint.
- C. Request an X.509 certificate.
- D. Create a custom subdomain.
- E. Enable a virtual network service endpoint.
- F. Register an application in Microsoft Entra ID (or use a managed identity) and acquire an OAuth 2.0 access token for Azure AI services/Speech.
- G. Assign the Entra ID principal an appropriate Azure RBAC role on the Speech (Cognitive Services) resource (for example, Cognitive Services User).

ANSWER: D F G

Explanation:

Microsoft Entra ID authentication for Azure AI Speech requires a resource configured with a custom subdomain. The custom subdomain provides the resource-specific endpoint needed when requesting and presenting Microsoft Entra access tokens to Azure AI services. The application must also have an identity that can obtain an OAuth 2.0 access token. This can be an application registered in Microsoft Entra ID using an application credential, or a workload using a managed identity; in either case, the token must be requested for the Azure AI services resource audience or scope.

Authentication alone does not grant access to the Speech resource. The identity represented by the token must be authorized through Azure role-based access control on that resource. Assigning an appropriate built-in Cognitive Services role, such as *Cognitive Services User* when it meets the required access level, enables the principal to invoke the service using its Microsoft Entra token. Together, the custom subdomain, token-acquiring identity, and RBAC assignment establish the endpoint, authentication, and authorization required for token-based access. For implementation details, see [Authenticate requests to Azure AI services](#) and [Configure Microsoft Entra ID authentication for Azure AI Speech](#).

QUESTION NO: 29

You are developing the smart e-commerce project.

You need to implement autocompletion as part of the Cognitive Search solution.

Which three actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Make API queries to the autocomplete endpoint and include suggesterName in the body.
- B. Add a suggester that has the three product name fields as source fields.
- C. Make API queries to the search endpoint and include the product name fields in the searchFields query parameter.
- D. Add a suggester for each of the three product name fields.
- E. Set the searchAnalyzer property for the three product name variants.
- F. Set the analyzer property for the three product name variants.

Use the default standard Lucene analyzer ("analyzer": null) or a language analyzer (for example, "analyzer": "en.Microsoft") on the field.

Reference: <https://docs.microsoft.com/en-us/azure/search/index-add-suggesters>

ANSWER: A B F

Explanation:

Azure AI Search autocomplete is built on a suggester defined in the search index. The action “Add a suggester that has the three product name fields as source fields” is required because a single suggester can include multiple eligible source fields, allowing prefixes from all three product-name variants to contribute to typeahead results. Azure AI Search creates the prefix-oriented data needed by autocomplete when the index is built or updated.

The fields used as suggester source fields must also use a supported analyzer. Therefore, “Set the analyzer property for the three product name variants. Use the default standard Lucene analyzer ("analyzer": null) or a language analyzer (for example, "analyzer": "en.Microsoft") on the field.” is required. Standard Lucene and Microsoft language analyzers provide the tokenization behavior supported for suggester fields.

Finally, “Make API queries to the autocomplete endpoint and include suggesterName in the body.” is required at query time. The autocomplete request supplies the partial user input and identifies the index suggester to use through `suggesterName`. This connects the client’s typeahead request to the configured product-name source fields. See [Add suggesters in Azure AI Search](#) and [Add autocomplete and suggestions in Azure AI Search](#).

QUESTION NO: 30

You build a Language Understanding model by using the Language Understanding portal. You export the model as a JSON file as shown in the following sample.

```
{
  "text": "average amount of rain by month at chicago last year",
  "intent": "Weather.CheckWeatherValue",
  "entities": [
    {
      "entity": "Weather.WeatherRange",
      "startPos": 0,
      "endPos": 6,
      "children": []
    },
    {
      "entity": "Weather.WeatherCondition",
      "startPos": 18,
      "endPos": 21,
      "children": []
    },
    {
      "entity": "Weather.Historic",
      "startPos": 23,
      "endPos": 30,
      "children": []
    }
  ]
}
```

To what does the Weather.Historic entity correspond in the utterance?

- A. by month.
- B. chicago.
- C. rain.
- D. location.

ANSWER: A

Explanation:

by month. is the text span represented by the `Weather.Historic` entity. In a LUIS application export, each labeled entity includes its entity name together with character-position information, typically `startPos` and `endPos`. Those offsets identify the exact portion of the training utterance that was annotated, rather than merely describing the general purpose of the entity. The offsets shown for `Weather.Historic` align with the words *by month* in the utterance.

The hierarchical name also provides useful semantic context: `Weather` is the parent entity and `Historic` identifies a child entity used for a historical or time-based weather request. The phrase *by month* expresses how historical weather information should be grouped or requested, so it is the appropriate text to receive that label. Entity labeling is performed on literal text spans in example utterances, and those labels are retained in the exported application definition for use during training and prediction.

For more detail on LUIS entity concepts and labeling utterance text, see [LUIS entity types](#) and [Add and label entities in LUIS](#).

QUESTION NO: 31

You are developing the knowledgebase by using Azure Cognitive Search.

You need to process wiki content to meet the technical requirements.

What should you include in the solution?

- A. an indexer for Azure Blob storage attached to a skillset that contains the language detection skill and the text translation skill
- B. an indexer for Azure Blob storage attached to a skillset that contains the language detection skill
- C. an indexer for Azure Cosmos DB attached to a skillset that contains the document extraction skill and the text translation skill
- D. an indexer for Azure Cosmos DB attached to a skillset that contains the language detection skill and the text translation skill

ANSWER: C

Explanation:

The correct solution is an indexer for Azure Cosmos DB attached to a skillset that contains the document extraction skill and the text translation skill. An Azure AI Search indexer must be configured for the data source that stores the wiki content. The Azure Cosmos DB indexer provides the connection between the Cosmos DB data source and the search index, including the ability to run an attached skillset during indexing. This supports automated ingestion and enrichment rather than requiring a separate process to copy the wiki data to another storage service.

The Document Extraction skill is appropriate when the content to enrich is a document file supplied in a field, such as binary or Base64-encoded content. It extracts text and metadata so that downstream skills can operate on the document's textual content. The Text Translation skill can then translate extracted wiki text into the required target language for consistent indexing and search. The translation service can identify the source language when one is not explicitly supplied, allowing multilingual source content to be processed without a separate detection stage. This combined indexer and skillset design produces enriched, searchable text directly from the Cosmos DB-hosted wiki content.

See [Index data from Azure Cosmos DB for NoSQL](#) and [Document Extraction skill](#).

QUESTION NO: 32 - (DRAG DROP)

DRAG DROP

You train a Custom Vision model used in a mobile app.

You receive 1,000 new images that do not have any associated data.

You need to use the images to retrain the model. The solution must minimize how long it takes to retrain the model.

Which three actions should you perform in the Custom Vision portal? To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.

Select and Place:

Actions	Answer Area
Upload the images by category.	
Get suggested tags.	
Upload all the images.	⬅
Group the images locally into category folders.	➡
Review the suggestions and confirm the tags.	
Tag the images manually.	⬆
	⬇

ANSWER:**Actions****Answer Area****Explanation:**

The fastest retraining preparation workflow is to upload all the images, obtain suggested tags, and then review and confirm those tags. Uploading the full set places the new images into the existing Custom Vision project, where they can be evaluated against the project's current trained model. Because the project already has learned visual patterns and tag definitions, Custom Vision can use that prior learning to propose likely tags for the newly added images.

Suggested tags are valuable here because they turn a large, initially unlabelled image set into a review-oriented labeling task. Instead of creating every label from scratch, the reviewer can validate the model's proposed tags and make any needed corrections. Confirmed tags become the ground-truth annotations associated with the uploaded images, making those images ready to contribute to the next training iteration. This workflow uses the model's existing knowledge while retaining human validation, which helps keep the training data reliable.

The sequence matters: tag suggestions can only be generated after the images exist in the project, and reviewing them must follow tag generation because confirmation establishes the final labels used for retraining. Once confirmation is complete, the project contains additional labeled examples and can be trained again with the expanded dataset. Microsoft documents the Custom Vision image upload, tagging, and training workflow in [Build a classifier with Custom Vision](#). Additional service guidance is available in the [Custom Vision documentation](#).

QUESTION NO: 33

You have an Azure subscription that contains an Azure App Service app named App1. You provision a multi-service Azure Cognitive Services resource named CSAccount1.

You need to configure App1 to access CSAccount1. The solution must minimize administrative effort. What should you use to configure App1?

- A. a system-assigned managed identity and an X.509 certificate.
- B. the endpoint URI and an OAuth token.
- C. the endpoint URI and a shared access signature (SAS) token.
- D. the endpoint URI and subscription key.

ANSWER: D**Explanation:**

The endpoint URI and subscription key is the appropriate configuration for App1 to access the multi-service Azure AI services resource with minimal administrative effort. A multi-service resource exposes a common endpoint that applications use to send requests to the individual Azure AI services enabled through that resource. The resource also provides two subscription keys, allowing an application to authenticate requests simply by supplying a key in the required request header along with the endpoint.

This approach has a small setup footprint: obtain an endpoint and either resource key from CSAccount1, then store them as protected App Service application settings. App Service settings are injected into the application as environment variables and are encrypted at rest, so the application can retrieve the values without embedding credentials in source code. The two available keys also support key rotation with minimal disruption: configure the alternate key, regenerate the former key, and repeat as needed.

For the key-based authentication flow and required request headers, see [Authenticate requests to Azure AI services](#). For details about multi-service resources and the shared endpoint/key model, see [Azure AI services multi-service resource](#).

QUESTION NO: 34

You have an Azure subscription that contains an Azure AI Document Intelligence resource named DM. DI1 uses the Standard SO pricing tier

You have the files shown in the following table.

Name	Size	Description
File1.pdf	800 MB	Contains scanned images
File2.jpg	1 KB	An image that has 25 x 25 pixels
File3.tiff	5 MB	An image that has 5000 x 5000 pixels

Which files can you analyze by using DI1?

- A. File1.pdf only.
- B. File2.jpg only.
- C. File3 tiff only.
- D. File2.jpg and File3.tiff only.
- E. File1.pdf, File2.jpg, and File3.tiff.

ANSWER: E

Explanation:

File1.pdf, File2.jpg, and File3.tiff. is correct because Azure AI Document Intelligence supports PDF documents and common image formats, including JPEG and TIFF, as input for document analysis. The Standard (S0) tier supports these formats for Document Intelligence models, subject to the applicable service limits for the selected model and API version, such as document size and page-count limits. Thus, the PDF file, JPEG image, and TIFF image listed can each be submitted to the Document Intelligence resource for analysis.

Document Intelligence accepts PDF input for document-oriented workflows and accepts raster-image input formats for extracting text, layout, key-value pairs, tables, and other model-specific results. TIFF is supported as an image document format, including multi-page TIFF where applicable. The S0 tier is the paid tier intended for production use and provides the capacity needed to process supported document types within the service's documented constraints. Review the current input requirements before submitting documents, because limits can vary by model and API version.

See [Document Intelligence input requirements](#) and [Azure AI Document Intelligence overview](#).

QUESTION NO: 35

You are developing an application that uses an Azure OpenAI chat model to generate purchase order records.

You need the model response to conform to a JSON structure that contains an order number, a supplier name, and an array of line items. The solution must minimize response-parsing errors.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Define a JSON Schema for the expected response.
- B. Configure the request to use structured outputs.
- C. Set the temperature to 1.
- D. Create a fine-tuning job for every purchase order format.
- E. Enable semantic ranker in Azure AI Search.

ANSWER: A B

Explanation:

Define a JSON Schema for the expected response. is required to describe the required object structure, property types, and nested line-item array. The schema provides the contract that the model output must follow.

Configure the request to use structured outputs. is also required. Structured outputs instruct a supported Azure OpenAI model to generate output that adheres to the supplied JSON Schema. This is more reliable than only asking in natural language for JSON, and it reduces application-side validation and parsing complexity. See [Structured outputs in Azure OpenAI](#).

QUESTION NO: 36 - (HOTSPOT)

You have an app named App1 that uses Azure AI Document Intelligence to analyze medical You have an app named App1 that uses Azure AI Document Intelligence for patients. You send a request to App1 and receive the following response.

```

{
  "status": "succeeded",
  "createdDateTime": "2023-09-14T21:01:02Z",
  "lastUpdatedDateTime": "2023-09-14T21:01:03Z",
  "analyzeResult": {
    "apiVersion": "2023-07-31",
    "modelId": "prebuilt-healthInsuranceCard.us",
    "stringIndexType": "utf16CodeUnit",
    "content": "Blood Pressure 118/72",
    "pages": [
      {
        ...
        "words": [
          {
            "content": "Blood",
            "polygon": [ ... ],
            "confidence": 0.766,
            "span": { ... }
          },
          {
            "content": "Pressure",
            "polygon": [ ... ],
            "confidence": 0.716,
            "span": { ... }
          },
          {
            "content": "118/72",
            "polygon": [ ... ],

```

For each of the following statements, select Yes if the statement is true. Otherwise, select No. NOTE: Each correct selection is worth point.

Answer Area

Statements

The chosen model is suitable for the intended use case.

The text content was recognized with greater than 70 percent confidence.

The form elements were recognized with greater than 70 percent confidence.

Yes

No

☐
☐
☐
☐
☐
☐

ANSWER:

Answer Area

Statements

The chosen model is suitable for the intended use case.

The text content was recognized with greater than 70 percent confidence.

The form elements were recognized with greater than 70 percent confidence.

Yes

No



Explanation:

The correct selections are **No** for the suitability statement, **Yes** for the text-content confidence statement, and **No** for the form-elements confidence statement. The response identifies the model as `prebuilt-healthInsuranceCard.us`. Azure AI Document Intelligence prebuilt models are specialized for particular document types, and this model is intended to recognize information on United States health insurance cards. A document containing a patient vital reading such as “Blood Pressure 118/72” is medical-record content rather than an insurance-card document, so the model is not suitable for the intended medical-document analysis use case. Microsoft describes the available prebuilt model capabilities in the [Document Intelligence prebuilt models overview](#).

The raw OCR results shown in the response support the text-confidence selection. In the `pages` and `words` data, recognized text is accompanied by a confidence score. The visible values for “Blood” and “Pressure” are 0.766 and 0.716. Since each value is greater than 0.70, the recognized text content meets the stated greater-than-70-percent threshold. Confidence values are returned as decimal values from zero through one, so 0.716 represents 71.6 percent confidence. The response schema and confidence behavior are documented in Microsoft’s [Analyze Document response documentation](#).

Form elements require evidence of structured extraction, such as document fields, key-value pairs, or form-field results, together with their associated confidence values. The displayed information provides OCR word recognition, but it does not provide form-element results demonstrating confidence above 0.70. Therefore, the form-elements statement is correctly marked No.

QUESTION NO: 37 - (HOTSPOT)

HOTSPOT

You are reviewing the design of a chatbot. The chatbot includes a language generation file that contains the following fragment.

```
# Greet(user)
```

```
- ${Greeting()}, ${user.name}
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

NOTE: Each correct selection is worth one point.

Hot Area:

Answer Area

Statements	Yes	No
<code>\${user.name}</code> retrieves the user name by using a prompt.	☐	☐
<code>Greet()</code> is the name of the language generation template.	☐	☐
<code>\${Greeting() }</code> is a reference to a template in the language generation file.	☐	☐

ANSWER:

Answer Area

Statements	Yes	No
<code>\${user.name}</code> retrieves the user name by using a prompt.	☐	☒
<code>Greet()</code> is the name of the language generation template.	☐	☒
<code>\${Greeting() }</code> is a reference to a template in the language generation file.	☒	☐

Explanation:

Language Generation (LG) templates in Bot Framework Composer use a concise syntax for declaring templates, accepting parameters, reading values, and composing one template from another. In the fragment shown, `# Greet(user)` declares a template whose identifier is `Greet` and whose parameter is `user`. The template can then use that supplied object while generating its response. The expression `${user.name}` evaluates the `name` property on the available `user` object. This is data access and text interpolation: the value must already be present in the object or in the conversational state used to supply the object. A prompt is a dialog action that collects input earlier in the conversation, rather than something performed by this LG property expression.

The expression `${Greeting() }` uses LG template invocation syntax. It evaluates the template named `Greeting` and inserts the generated result into the text produced by the surrounding `Greet` template. This lets a bot keep reusable phrases, such as time-sensitive or localized greetings, in their own templates while combining them naturally with values such as a user's name. Accordingly, the marked responses are No for the first statement, No for the second statement, and Yes for the third statement. Microsoft documents LG template declarations, parameterized templates, expressions, and template references in the [Bot Framework language generation overview](#) and the [Composer language generation guidance](#).

QUESTION NO: 38

You are building a Conversational Language Understanding model.

You need to ensure that the model will support the following sample utterances: Set all the lights to on.

Turn off the lights in the living room.

What is the current thermostat temperature?

Lower the temperature of the thermostat by five degrees. Which three elements should you add to the model?

Each correct answer presents part of the solution. NOTE: Each correct selection is worth one point.

A. a location Intent.

- B. a change setting entity.
- C. a device intent.
- D. a change setting intent.
- E. a query setting intent.
- F. a device entity.

ANSWER: D E F

Explanation:

A **change setting intent** is required to identify utterances in which the user requests a modification to a device or its state. This includes setting lights to on, turning lights off, and lowering a thermostat temperature. An intent represents the overall goal of the user's utterance, so a single change-oriented intent can route these commands to the appropriate device-control workflow.

A **query setting intent** is required for requests that retrieve information rather than modify a setting. Asking for the current thermostat temperature has a distinct user goal: obtaining the present value of a device setting. Separating this goal from change requests enables the application to invoke a read/query operation instead of a control operation.

A **device entity** is required to extract the device being discussed, such as *lights* or *thermostat*. Entities identify relevant information within an utterance that the application needs after intent classification. The extracted device value allows downstream logic to select the appropriate device and process the recognized change or query action. Conversational Language Understanding schemas use intents to represent user actions and entities to extract important data from utterances. See [Conversational Language Understanding overview](#) and [Build a CLU schema](#).

QUESTION NO: 39

You have an Azure Cognitive Search solution and a collection of blog posts that include a category field. You need to index the posts. The solution must meet the following requirements:

Include the category field in the search results.

Ensure that users can search for words in the category field.

Ensure that users can perform drill down filtering based on category. Which index attributes should you configure for the category field?

- A. searchable, facetable, and retrievable.
- B. retrievable, filterable, and sortable.
- C. retrievable, facetable, and key.
- D. searchable, sortable, and retrievable.

ANSWER: A

Explanation:

The correct configuration is **searchable, facetable, and retrievable**. A field must be retrievable for its stored value to be included in documents returned by Azure AI Search query results. This directly meets the requirement to include each post's category in the search results.

The searchable attribute enables full-text search over the field. Azure AI Search applies text analysis to searchable string fields, allowing query terms to match words in category values. This fulfills the requirement that users can search for words contained in the category field.

Faceted navigation provides the drill-down experience. When a field is facetable, Azure AI Search can generate category buckets and counts from matching documents. An application can present these buckets to users and use a selected

category to refine subsequent results. Category is a typical low-cardinality metadata field for which faceting is designed. In production, applications commonly also configure such a field as filterable when they need to submit an OData filter after a user selects a facet, but the stated attribute combination that satisfies returning the value, full-text searching it, and providing facet drill-down is searchable, facetable, and retrievable.

For attribute behavior, see [Create an index in Azure AI Search](#). For drill-down and facet queries, see [Faceted navigation in Azure AI Search](#).

QUESTION NO: 40

You have an Azure Cognitive Search solution and a collection of handwritten letters stored as JPEG files.

You plan to index the collection. The solution must ensure that queries can be performed on the contents of the letters.

You need to create an indexer that has a skillset. Which skill should you include?

- A. key phrase extraction.
- B. optical character recognition (OCR).
- C. document extraction.
- D. image analysis.

ANSWER: B

Explanation:

The correct skill is **optical character recognition (OCR)**. The source files are JPEG images, so the written letter content is not initially available as searchable text in an Azure AI Search index. Adding the OCR skill to the indexer's skillset enables an enrichment pipeline to recognize text in normalized images and emit that recognized text as an output field. That output can then be mapped to a searchable string field in the target index, allowing users to issue full-text queries against the contents of the letters.

Azure AI Search's OCR skill is designed to extract both printed and handwritten text from image inputs. For blob indexing scenarios, image extraction and normalization must be enabled in the indexer configuration so that image content is provided to the skillset. The OCR output should be projected or mapped into the index schema with searching enabled. This approach retains the original JPEG files while creating an indexable textual representation of their handwritten content.

See the [OCR cognitive skill documentation](#) and Microsoft's guidance for [image and OCR enrichment scenarios in Azure AI Search](#).

QUESTION NO: 41

You develop a custom question answering project in Azure Cognitive Service for Language. The project will be used by a chatbot. You need to configure the project to engage in multi-turn conversations. What should you do?

- A. Add follow-up prompts.
- B. Enable active learning.
- C. Add alternate questions.
- D. Enable chit-chat.

ANSWER: A

Explanation:

Add follow-up prompts. In Azure AI Language custom question answering, follow-up prompts enable a guided, multi-turn experience. A prompt is associated with a question-and-answer pair and represents a suggested next question that the

chatbot can offer after returning that pair's answer. The user's selection is sent as the next query, allowing the chatbot to progress through related questions and answers as a conversational flow. Prompts can link to existing question-and-answer pairs in the same project, so an author can design branching paths such as a troubleshooting sequence, a product-selection workflow, or an information-gathering dialog. The application that calls the question answering endpoint receives the returned prompts and is responsible for displaying them to the user, commonly as buttons or suggested replies, then submitting the selected prompt in the next turn. This capability is specifically designed for contextual, guided dialog rather than merely finding a standalone answer. Configure prompts while authoring the project, then publish the project so the deployed chatbot runtime can retrieve the answers and associated follow-up prompt metadata. For implementation guidance, see [Multi-turn conversation in Azure AI Language question answering](#) and the [custom question answering concepts documentation](#).

QUESTION NO: 42

You have an Azure AI Search solution that indexes PDF files from an Azure Blob Storage container.

You need to ensure that the index is updated when files are added, changed, or deleted in the container. The solution must minimize the amount of data processed during each indexer run.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Create an Azure Blob Storage data source.
- B. Create and schedule an indexer.
- C. Rebuild the search index after every file upload.
- D. Create a synonym map for the PDF content.
- E. Enable semantic ranker on the index.

ANSWER: A B

Explanation:

Create an Azure Blob Storage data source. is required because an Azure AI Search indexer needs a data source definition that identifies the storage account and container from which it reads the files. Blob indexers maintain state for the documents they process.

Create and schedule an indexer. is also required. When an indexer runs again, it uses built-in change detection to identify blobs that have changed since the previous run and updates the corresponding search documents. Blob deletions can be detected by configuring a supported deletion-detection policy, such as native soft delete, so removed source files are also removed from the index. This avoids reprocessing unchanged files during each run. See [Index data from Azure Blob Storage](#) and [Change and deletion detection for blob indexers](#).

QUESTION NO: 43

You are developing a voice application by using Azure AI Speech.

You need to improve speech recognition accuracy for product names and technical terms that are uncommon in general language.

Which two actions should you perform? Each correct answer presents part of the solution.

NOTE: Each correct selection is worth one point.

- A. Create a custom speech project.
- B. Provide text data that contains the domain-specific vocabulary.
- C. Create a custom neural voice.

- D. Enable speaker recognition.
- E. Use speech synthesis markup language (SSML).

ANSWER: A B

Explanation:

Create a custom speech project to customize speech-to-text recognition for the application domain. A custom speech project provides the workspace in Speech Studio for preparing data, training a custom model where supported, and evaluating recognition quality.

You should also **provide text data that contains the domain-specific vocabulary**. Text data helps the service learn language patterns and terminology that are relevant to the application, such as product names and technical expressions. This can improve recognition of terms that are underrepresented in the base model.

Audio data with transcriptions can be used when acoustic characteristics, such as specialized recording environments or speakers, also require adaptation. Text data is particularly relevant when the primary issue is domain vocabulary. See [Custom Speech overview](#).

QUESTION NO: 44

You need to implement a table projection to generate a physical expression of an Azure Cognitive Search index.

Which three properties should you specify in the skillset definition JSON configuration table node? Each correct answer presents part of the solution. (Choose three.)

NOTE: Each correct selection is worth one point.

- A. tableName
- B. generatedKeyName
- C. dataSource
- D. dataSourceConnection
- E. source

ANSWER: A B E

Explanation:

A table projection is configured in a skillset's `knowledgeStore.projections.tables` collection. The `tableName` property identifies the destination Azure Storage table where the projected entities are persisted. This physical table is created or populated in the storage account configured for the knowledge store.

The `generatedKeyName` property specifies the name of the key column that Azure AI Search generates for projection rows. A generated key provides a stable, unique identifier for each entity written to the table and supports relationships between projected data, including cases where the source produces multiple rows.

The `source` property identifies the enrichment-tree path containing the data to project. In typical implementations, this path points to the output of a Shaper skill, which creates the structured shape needed for a table projection. Together, `tableName`, `generatedKeyName`, and `source` define the storage target, row identity, and enriched content for the projection.

Microsoft documents table projections and their required configuration pattern in the [Knowledge store projections overview](#) and provides working JSON patterns in [Knowledge store projection examples](#).