NANYANG TECHNOLOGICAL UNIVERSITY

School of Computer Science and Engineering (SCSE)

SC4052: Cloud Computing

Academic Year: 2023-2024, Semester 2

Assignment 2

Teoh Xi Sheng (U2120456L)

Contents

1. Build a Quiz Generator with Google Gen-AI 3
2. Using Google Gen-AI to Generate Cloud Computing Quizzes 3
3. Using Nemobot with OpenAI API to Generate Quizzes 5
4. Exploring Browser-based LLM (WebGPU) 7
Appendix A: Nemobot Code Reference 8

1. Build a Quiz Generator with Google Gen-AI

The following is the summary of the tasks for creating this quiz generator:

Backend

This backend server will handle user requests, process requests by forwarding them to Google's Vertex AI, and return the responses in JSON format.

1. Using Google's Cloud Shell editor, build a RESTful web server using the Flask module from Python to handle POST and GET requests.
2. Program the web server such that it will handle parameters from the URL.
3. Process the parameters such as "topic" and "num_q" together with the prompt using the Vertex AI module from Python.
4. The module will send a request to the Vertex AI from Google Cloud and return the quizzes in JSON format.
5. Host the web server on Google Cloud so requests can be sent over the internet.
6. Backend URL: https://quiz-generator-2v42b5qzta-de.a.run.app/

Frontend

This frontend server allows creators to host quizzes generated by Vertex AI and learners to participate in live quizzes hosted like "Kahoot".

1. Upon logging into Quizaic, the creator can generate a quiz template by inputting the various parameters such as "Topic", "Number of Questions", and a "Custom Generator URL", which will be the backend URL that we have created previously.
2. After submitting the template, the creator can expect a quiz template created by Vertex AI in a user-friendly format.
3. The creator can then host a live quiz using the template for participants to participate in.

While building this application seemed relatively straightforward, we can appreciate the multiple cloud services provided by Google that made this possible.

Google Cloud Service	Usage
Cloud Shell	Provides users with a cloud development environment with CLI to interact directly with the services.
Cloud Build	Executes and builds the application into artifacts such as Docker containers
Artifact Registry	After building the app, the container will be hosted in the registry for deployment.
Cloud Run	We use this service to deploy the instance of the container application to the internet.
Cloud Storage	This is mainly used to store application data. For example, we can store the Vertex AI prompt that we have prompted for edits later.
Vertex AI	This is a SaaS like OpenAI's ChatGPT. We interact with it using API call.

2. Using Google Gen-AI to Generate Cloud Computing Quizzes

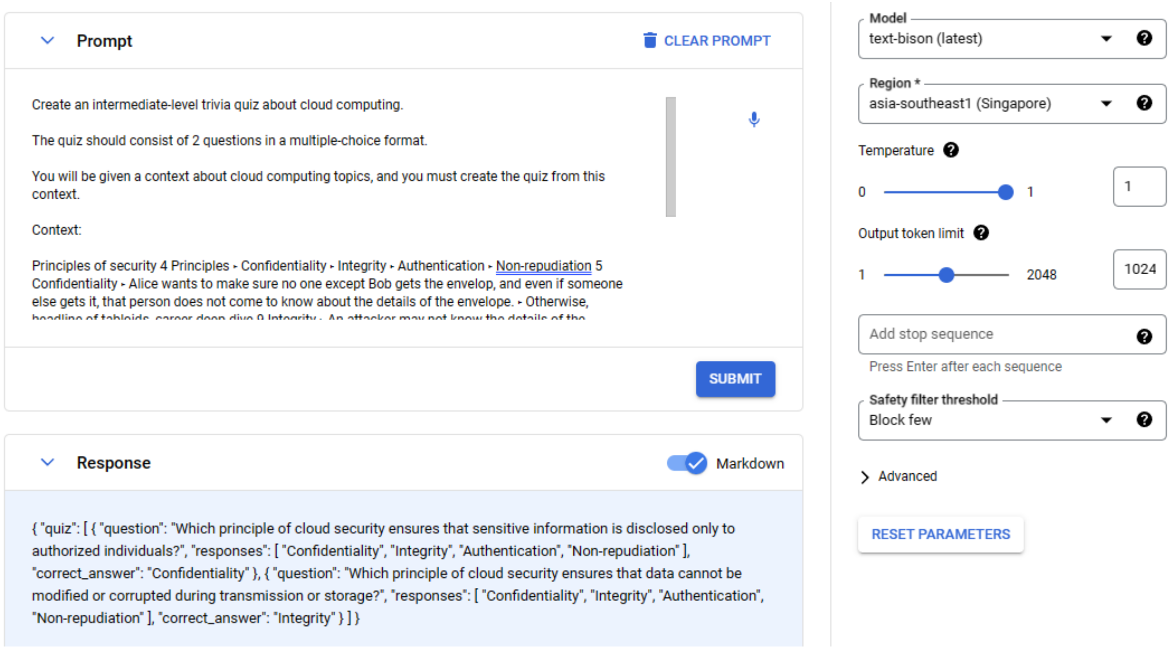
There are two possible ways to generate cloud computing quizzes using Google's Vertex AI.

Method 1: Quizaic with Vertex AI.

We can host a fully functional quiz using Quizaic as front-end and the backend server connected to Vertex AI. This method will be using the Vertex AI parameters that have been pre-configured, which means the creator has no control over settings such as the language model and temperature.

Method 2: Using Vertex AI directly.

If we would like to generate the quiz based on a certain specific knowledge, we can make use of the Vertex AI but feeding in some contexts. For this experiment, we will feed in texts from slide 4-12 from lecture 5 – Cloud Security.


Prompt

Create an intermediate-level trivia quiz about cloud computing.

The quiz should consist of 2 questions in a multiple-choice format.

You will be given a context about cloud computing topics, and you must create the quiz from this context.

Context:

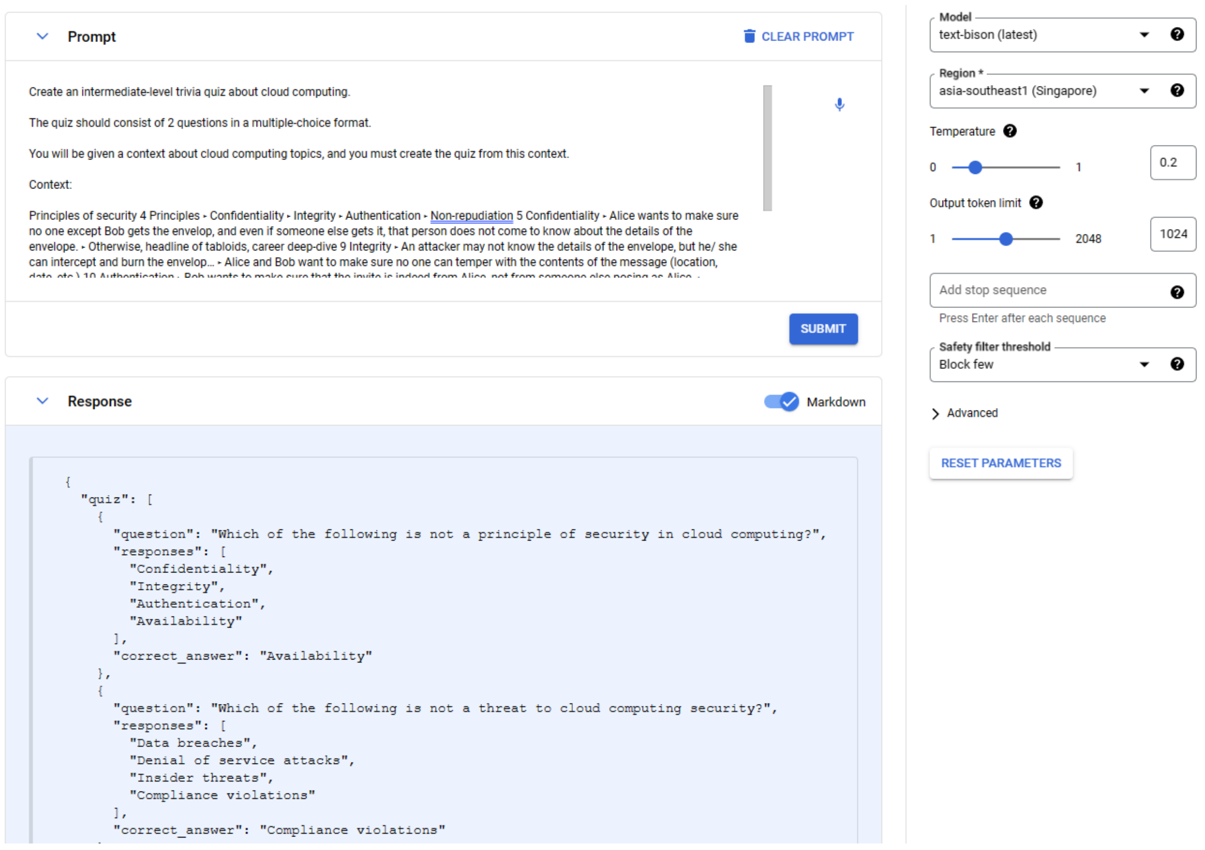
Principles of security 4 Principles ‣ Confidentiality ‣ Integrity ‣ Authentication ‣ Non-repudiation 5 Confidentiality ‣ Alice wants to make sure no one except Bob gets the envelop, and even if someone else gets it, that person does not come to know about the details of the envelope. ‣ Otherwise, headline of tabloids, career deep-dive 9 Integrity ‣ An attacker may not know the details of the

Response

{ "quiz": [{ "question": "Which principle of cloud security ensures that sensitive information is disclosed only to authorized individuals?", "responses": ["Confidentiality", "Integrity", "Authentication", "Non-repudiation"], "correct_answer": "Confidentiality" }, { "question": "Which principle of cloud security ensures that data cannot be modified or corrupted during transmission or storage?", "responses": ["Confidentiality", "Integrity", "Authentication", "Non-repudiation"], "correct_answer": "Integrity" }] }

Model: text-bison (latest)
Region *: asia-southeast1 (Singapore)
Temperature: 1
Output token limit: 1024
Add stop sequence
Press Enter after each sequence
Safety filter threshold: Block few

As we use the same prompt but setting the temperature to 0.2, it reduces the randomness of the response. It produces a more deterministic and conservative outputs despite feeding in a small context. In the scenario below, we can observe that both questions are asking similar questions, unlike when temperature = 1.


Prompt

Create an intermediate-level trivia quiz about cloud computing.

The quiz should consist of 2 questions in a multiple-choice format.

You will be given a context about cloud computing topics, and you must create the quiz from this context.

Context:

Principles of security 4 Principles ‣ Confidentiality ‣ Integrity ‣ Authentication ‣ Non-repudiation 5 Confidentiality ‣ Alice wants to make sure no one except Bob gets the envelop, and even if someone else gets it, that person does not come to know about the details of the envelope. ‣ Otherwise, headline of tabloids, career deep-dive 9 Integrity ‣ An attacker may not know the details of the envelope, but he/ she can intercept and burn the envelop... ‣ Alice and Bob want to make sure no one can temper with the contents of the message (location, date, etc.) 10 Authentication ‣ Bob wants to make sure that the invite is indeed from Alice, not from someone else posing as Alice. ‣

Response

```
{
  "quiz": [
    {
      "question": "Which of the following is not a principle of security in cloud computing?",
      "responses": [
        "Confidentiality",
        "Integrity",
        "Authentication",
        "Availability"
      ],
      "correct_answer": "Availability"
    },
    {
      "question": "Which of the following is not a threat to cloud computing security?",
      "responses": [
        "Data breaches",
        "Denial of service attacks",
        "Insider threats",
        "Compliance violations"
      ],
      "correct_answer": "Compliance violations"
```

Model: text-bison (latest)
Region *: asia-southeast1 (Singapore)
Temperature: 0.2
Output token limit: 1024
Add stop sequence
Press Enter after each sequence
Safety filter threshold: Block few

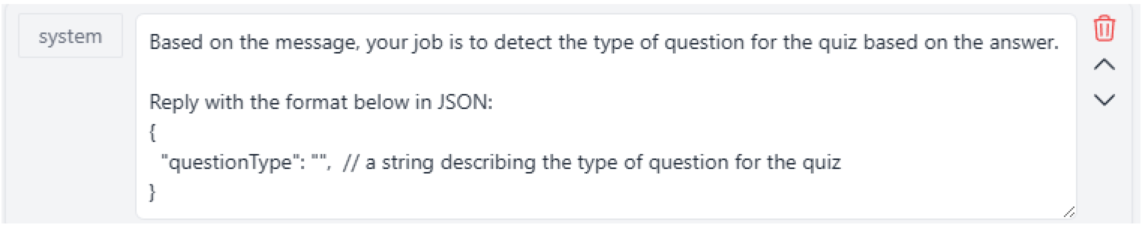
3. Using Nemobot with OpenAI API to Generate Quizzes

Using OpenAI API and Nemobot, we can create and perform prompt engineering by customising our LLM functions for specific application. Using Nemobot, I have created the following LLM functions so that the chatbot can generate quiz based on a given knowledge: Detailed code can be found in Appendix.

```
// Detect the content given, learn the knowledge, and add in other knowledge on
a similar topic so that it can be used to generate quiz questions.
const knowledgebase = parseKnowledgeBase({messagge});

// Detect the type of questions to generate
const questionType = parseQuestionType({message});

// Detect the number of questions to generate
const questionCount = parseQuestionCount({message});

// Generate the quiz questions
const generateQuestions({knowledgeBase, questionType, questionCount});
```

The initial prompt wasn't accurate despite letting it know what it should detect and response.

For example, in `parseQuestionType({message})`:

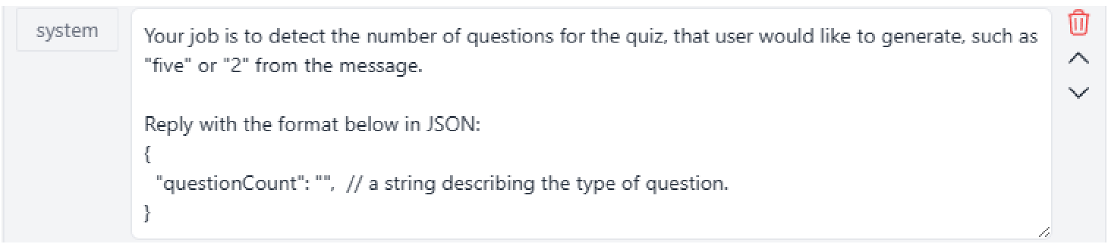


system

```
Based on the message, your job is to detect the type of question for the quiz based on the answer.

Reply with the format below in JSON:
{
  "questionType": "",  // a string describing the type of question for the quiz
}
```

Nemobot: What type of questions do you like to generate?		
User Response	**Nemobot Interpretation**	**Expected Interpretation**
"t f"	"true/false"	"true/false"
"multiple-choice type"	"multiple-choice"	"multiple-choice"
"a b c"	"Open-ended"	"multiple-choice"

In `parseQuestionCount({message})`:



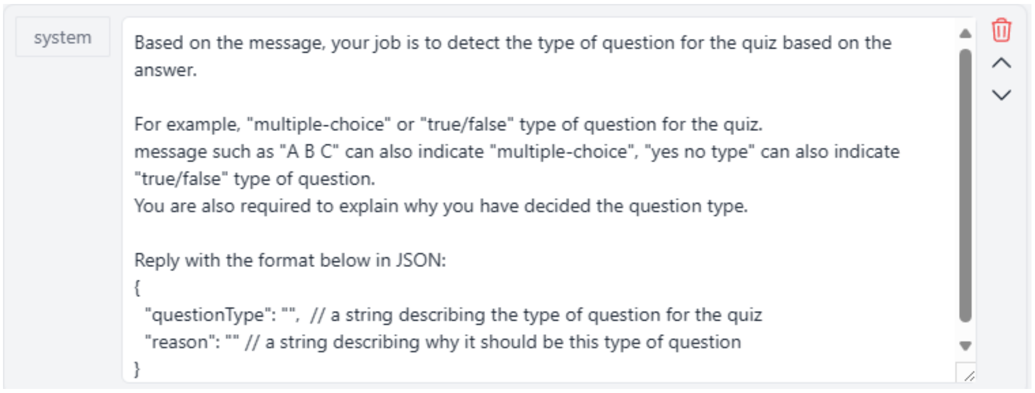
system

```
Your job is to detect the number of questions for the quiz, that user would like to generate, such as "five" or "2" from the message.

Reply with the format below in JSON:
{
  "questionCount": "",  // a string describing the type of question.
}
```

Nemobot: What type of questions do you like to generate?		
User Response	**Nemobot Interpretation**	**Expected Interpretation**
"eleven"	"eleven"	"eleven"
"ten + one"	"one"	"eleven"

With the help of unit testing, I have improved the accuracy and refined the functions by giving examples and requesting that reasoning to be included in its response (as suggested by the Nemobot tutorial).

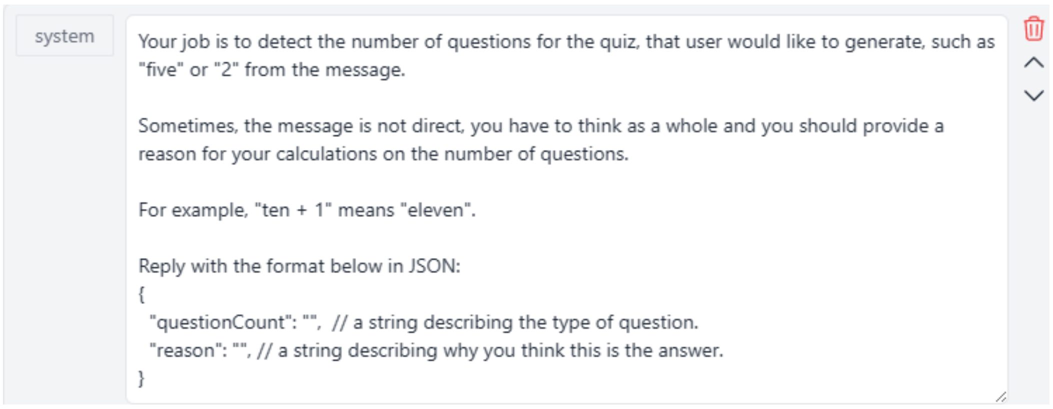
Improved `parseQuestionType({message})`:



system

```
Based on the message, your job is to detect the type of question for the quiz based on the answer.

For example, "multiple-choice" or "true/false" type of question for the quiz.
message such as "A B C" can also indicate "multiple-choice", "yes no type" can also indicate "true/false" type of question.
You are also required to explain why you have decided the question type.

Reply with the format below in JSON:
{
  "questionType": "",  // a string describing the type of question for the quiz
  "reason": "" // a string describing why it should be this type of question
}
```

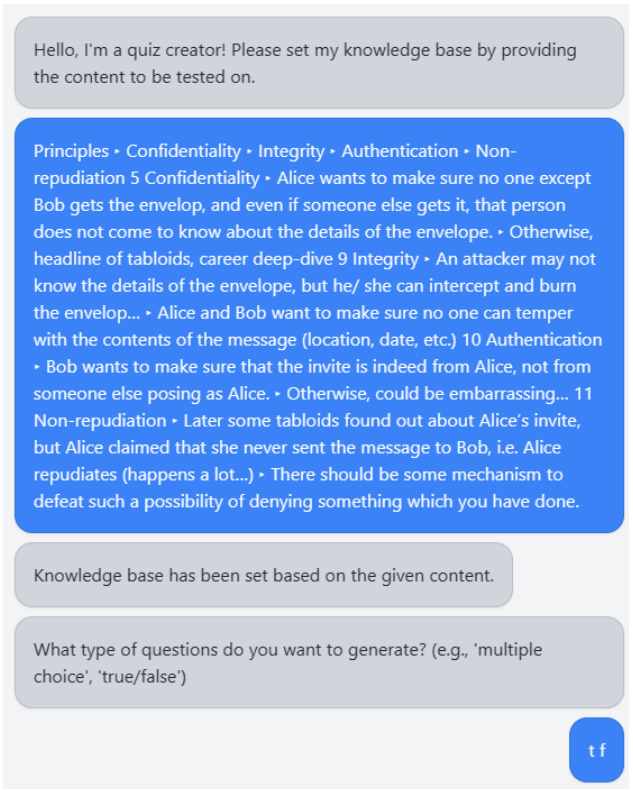
Improved `parseQuestionCount({message})`:



system

```
Your job is to detect the number of questions for the quiz, that user would like to generate, such as "five" or "2" from the message.

Sometimes, the message is not direct, you have to think as a whole and you should provide a reason for your calculations on the number of questions.

For example, "ten + 1" means "eleven".

Reply with the format below in JSON:
{
  "questionCount": "",  // a string describing the type of question.
  "reason": "", // a string describing why you think this is the answer.
}
```

Here is how the Nemobot perform. Full demo can be found in the screen recording in the zip folder.



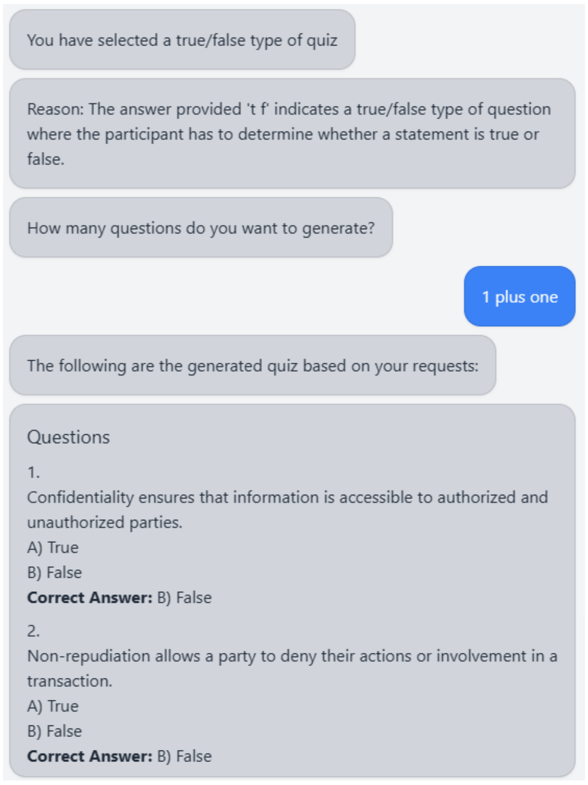
Hello, I'm a quiz creator! Please set my knowledge base by providing the content to be tested on.

Principles ‣ Confidentiality ‣ Integrity ‣ Authentication ‣ Non-repudiation 5 Confidentiality ‣ Alice wants to make sure no one except Bob gets the envelop, and even if someone else gets it, that person does not come to know about the details of the envelope. ‣ Otherwise, headline of tabloids, career deep-dive 9 Integrity ‣ An attacker may not know the details of the envelope, but he/ she can intercept and burn the envelop... ‣ Alice and Bob want to make sure no one can temper with the contents of the message (location, date, etc.) 10 Authentication ‣ Bob wants to make sure that the invite is indeed from Alice, not from someone else posing as Alice. ‣ Otherwise, could be embarrassing... 11 Non-repudiation ‣ Later some tabloids found out about Alice's invite, but Alice claimed that she never sent the message to Bob, i.e. Alice repudiates (happens a lot...) ‣ There should be some mechanism to defeat such a possibility of denying something which you have done.

Knowledge base has been set based on the given content.

What type of questions do you want to generate? (e.g., 'multiple choice', 'true/false')

t f



You have selected a true/false type of quiz

Reason: The answer provided 't f' indicates a true/false type of question where the participant has to determine whether a statement is true or false.

How many questions do you want to generate?

1 plus one

The following are the generated quiz based on your requests:

Questions

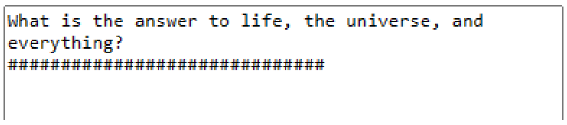
1.
Confidentiality ensures that information is accessible to authorized and unauthorized parties.
A) True
B) False
Correct Answer: B) False

2.
Non-repudiation allows a party to deny their actions or involvement in a transaction.
A) True
B) False
Correct Answer: B) False

4. Exploring Browser-based LLM (WebGPU)

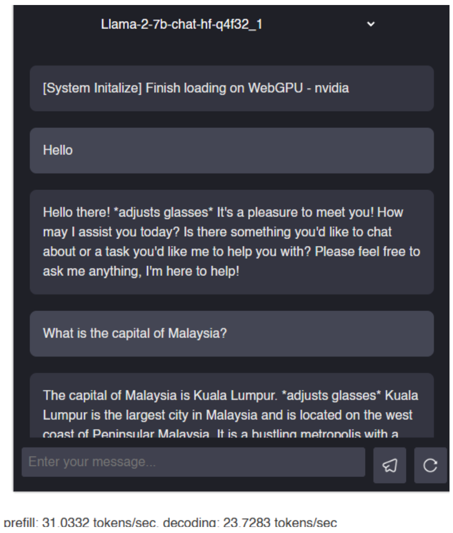
Instead of relying on cloud-based storage and execution of LLMs, advancements in WebGPU technology now enable clients to access and run LLMs directly on their local devices without internet connectivity. WebGPU allows web browsers to utilize the device's GPU as a hardware accelerator, enabling the models to be executed without the need for cloud GPU processing.

The WebGPT project doesn't work on my local machine or the web, and the model was responding with "#", similar to the issue raised here. The issue persists despite using the GPT2 1.5B model.



```
What is the answer to life, the universe, and
everything?
################################
```

However, as I researched WebGPU more, I came across another project, WebLLM, that supports more LLMs with huge parameters and is working on my machine.



Llama-2-7b-chat-hf-q4f32_1

[System Initalize] Finish loading on WebGPU - nvidia

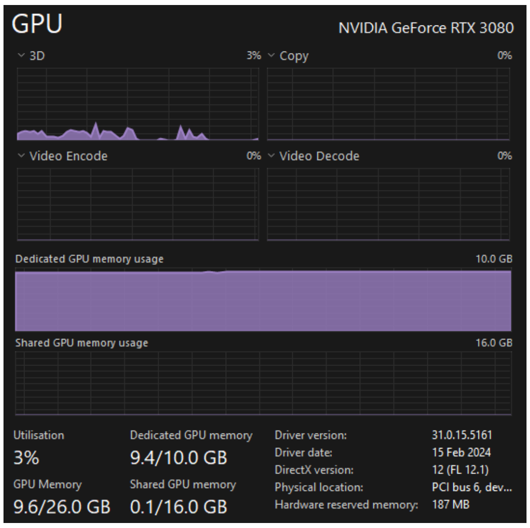
Hello

Hello there! *adjusts glasses* It's a pleasure to meet you! How may I assist you today? Is there something you'd like to chat about or a task you'd like me to help you with? Please feel free to ask me anything, I'm here to help!

What is the capital of Malaysia?

The capital of Malaysia is Kuala Lumpur. *adjusts glasses* Kuala Lumpur is the largest city in Malaysia and is located on the west coast of Peninsular Malaysia. It is a bustling metropolis with a

prefill: 31.0332 tokens/sec, decoding: 23.7283 tokens/sec



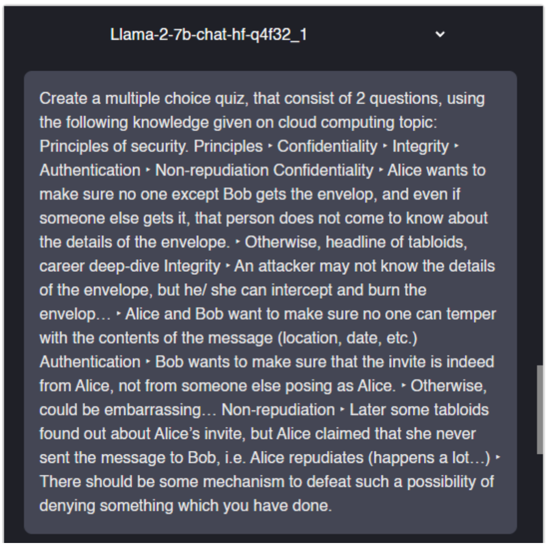
GPU NVIDIA GeForce RTX 3080

3D 3%
Copy 0%
Video Encode 0%
Video Decode 0%
Dedicated GPU memory usage 10.0 GB
Shared GPU memory usage 16.0 GB

Utilisation 3%
Dedicated GPU memory 9.4/10.0 GB
GPU Memory 9.6/26.0 GB
Shared GPU memory 0.1/16.0 GB
Driver version: 31.0.15.5161
Driver date: 15 Feb 2024
DirectX version: 12 (FL 12.1)
Physical location: PCI bus 6, dev...
Hardware reserved memory: 187 MB

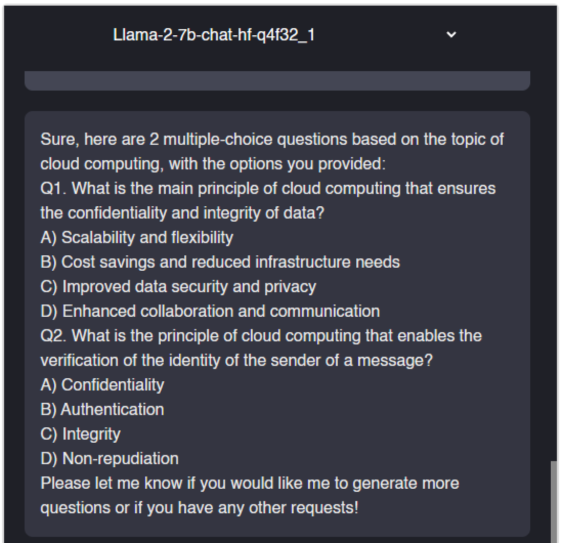
One way to tell that the WebGPU is running locally is that the 2.7b parameters LLM is consuming almost all of my Nvidia GPU's memory. The WebGPU API marks a significant improvement in the accessibility and scalability of AI technology, making it more accessible on mobile devices.

This is how the Llama 2.7B model responds to the prompt to create quizzes:



Llama-2-7b-chat-hf-q4f32_1

Create a multiple choice quiz, that consist of 2 questions, using the following knowledge given on cloud computing topic: Principles of security. Principles • Confidentiality • Integrity • Authentication • Non-repudiation Confidentiality • Alice wants to make sure no one except Bob gets the envelop, and even if someone else gets it, that person does not come to know about the details of the envelope. • Otherwise, headline of tabloids, career deep-dive Integrity • An attacker may not know the details of the envelope, but he/ she can intercept and burn the envelop… • Alice and Bob want to make sure no one can temper with the contents of the message (location, date, etc.) Authentication • Bob wants to make sure that the invite is indeed from Alice, not from someone else posing as Alice. • Otherwise, could be embarrassing… Non-repudiation • Later some tabloids found out about Alice's invite, but Alice claimed that she never sent the message to Bob, i.e. Alice repudiates (happens a lot…) • There should be some mechanism to defeat such a possibility of denying something which you have done.



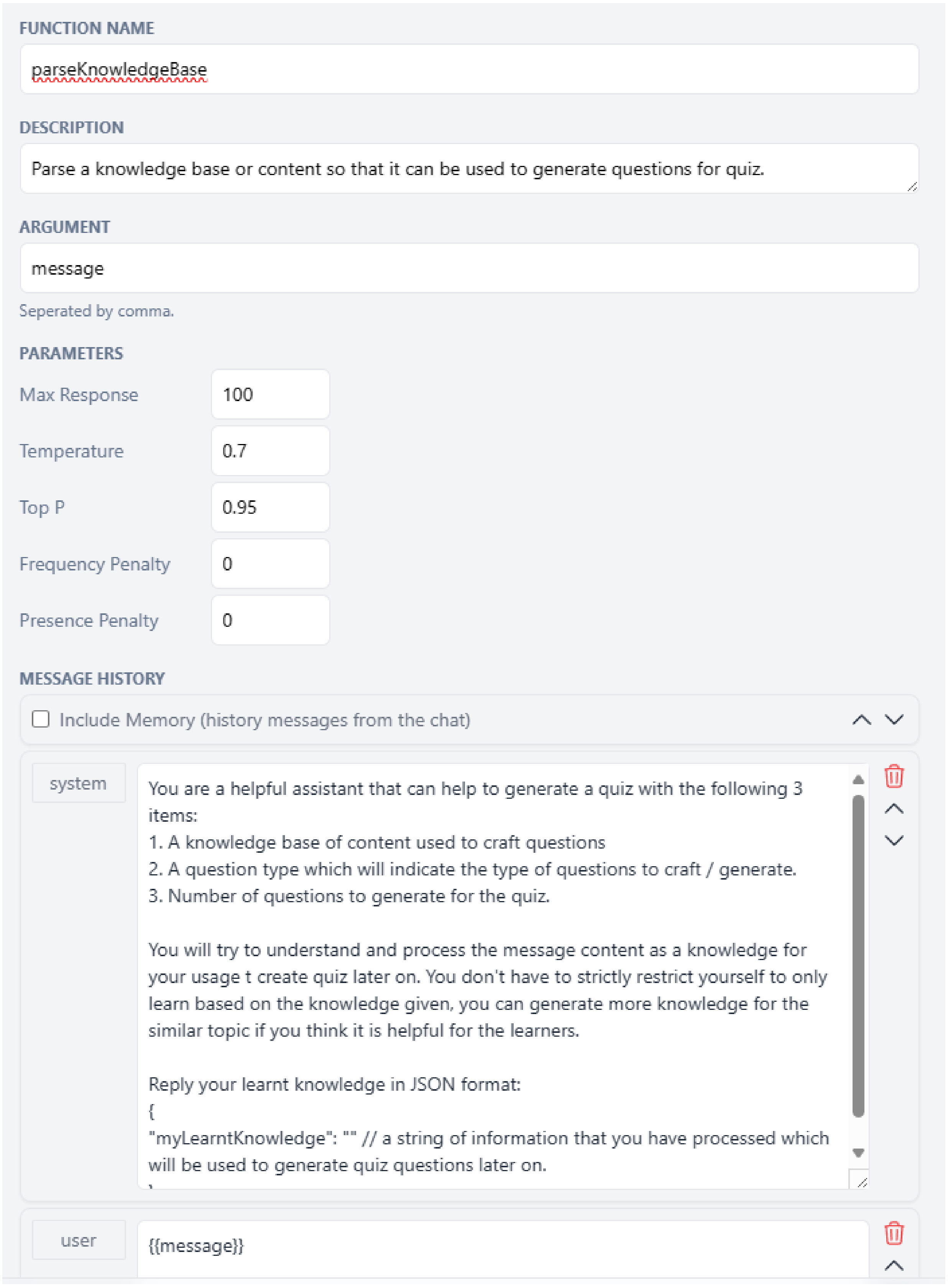
Llama-2-7b-chat-hf-q4f32_1

Sure, here are 2 multiple-choice questions based on the topic of cloud computing, with the options you provided:
Q1. What is the main principle of cloud computing that ensures the confidentiality and integrity of data?
A) Scalability and flexibility
B) Cost savings and reduced infrastructure needs
C) Improved data security and privacy
D) Enhanced collaboration and communication
Q2. What is the principle of cloud computing that enables the verification of the identity of the sender of a message?
A) Confidentiality
B) Authentication
C) Integrity
D) Non-repudiation
Please let me know if you would like me to generate more questions or if you have any other requests!

Appendix A: Nemobot Code Reference

The following is the code for the Nemobot and LLM functions. It can be loaded in New Playground.

GitHub Repository: https://github.com/xeroxis-xs/NTU-SC4052-Cloud-Computing



FUNCTION NAME

parseKnowledgeBase

DESCRIPTION

Parse a knowledge base or content so that it can be used to generate questions for quiz.

ARGUMENT

message

Seperated by comma.

PARAMETERS

Max Response	100
Temperature	0.7
Top P	0.95
Frequency Penalty	0
Presence Penalty	0

MESSAGE HISTORY

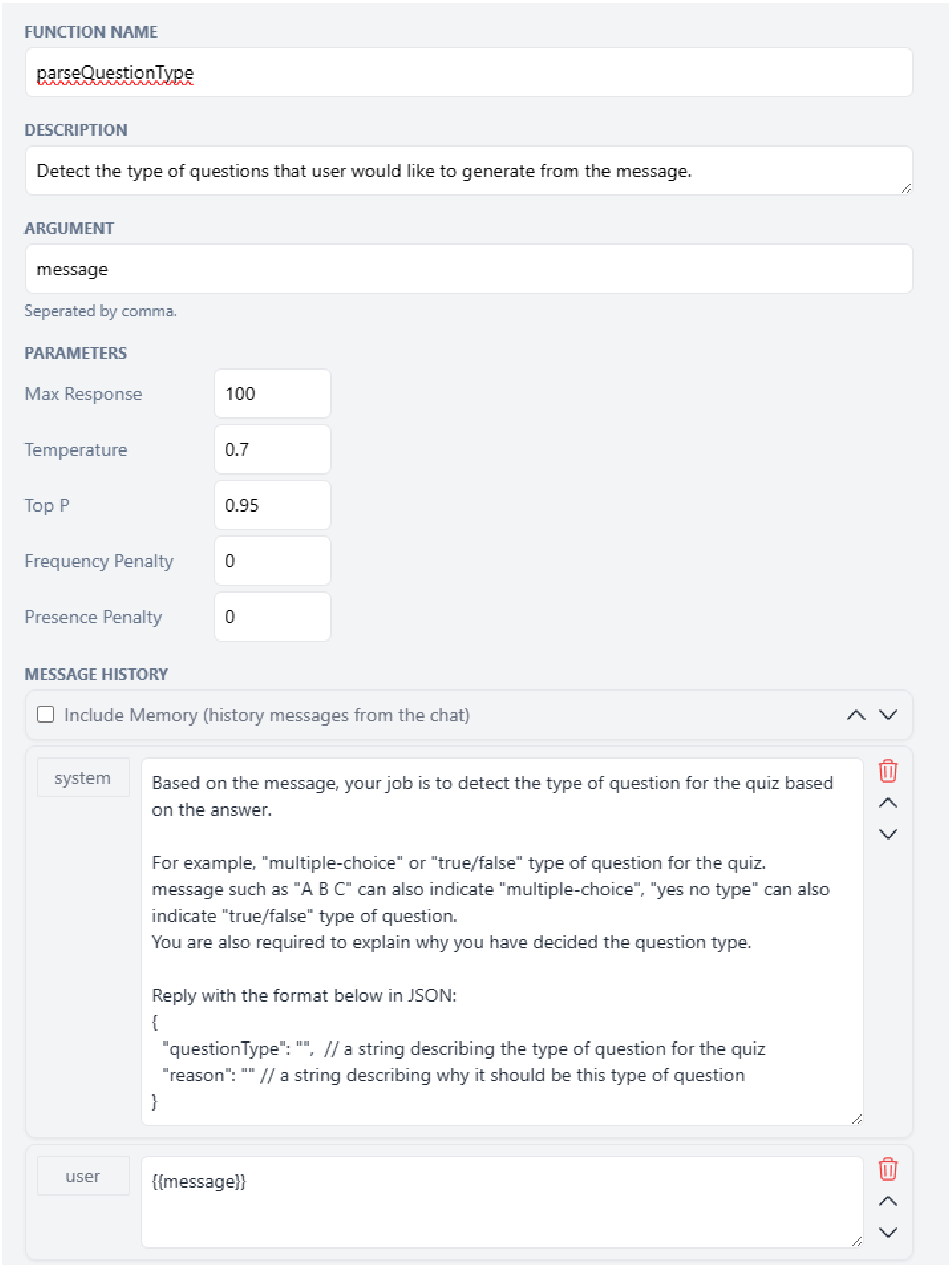
☐ Include Memory (history messages from the chat)

system

```
You are a helpful assistant that can help to generate a quiz with the following 3 items:
1. A knowledge base of content used to craft questions
2. A question type which will indicate the type of questions to craft / generate.
3. Number of questions to generate for the quiz.

You will try to understand and process the message content as a knowledge for your usage t create quiz later on. You don't have to strictly restrict yourself to only learn based on the knowledge given, you can generate more knowledge for the similar topic if you think it is helpful for the learners.

Reply your learnt knowledge in JSON format:
{
"myLearntKnowledge": "" // a string of information that you have processed which will be used to generate quiz questions later on.
```

user

```
{{message}}
```



FUNCTION NAME

parseQuestionType

DESCRIPTION

Detect the type of questions that user would like to generate from the message.

ARGUMENT

message

Seperated by comma.

PARAMETERS

Max Response	100
Temperature	0.7
Top P	0.95
Frequency Penalty	0
Presence Penalty	0

MESSAGE HISTORY

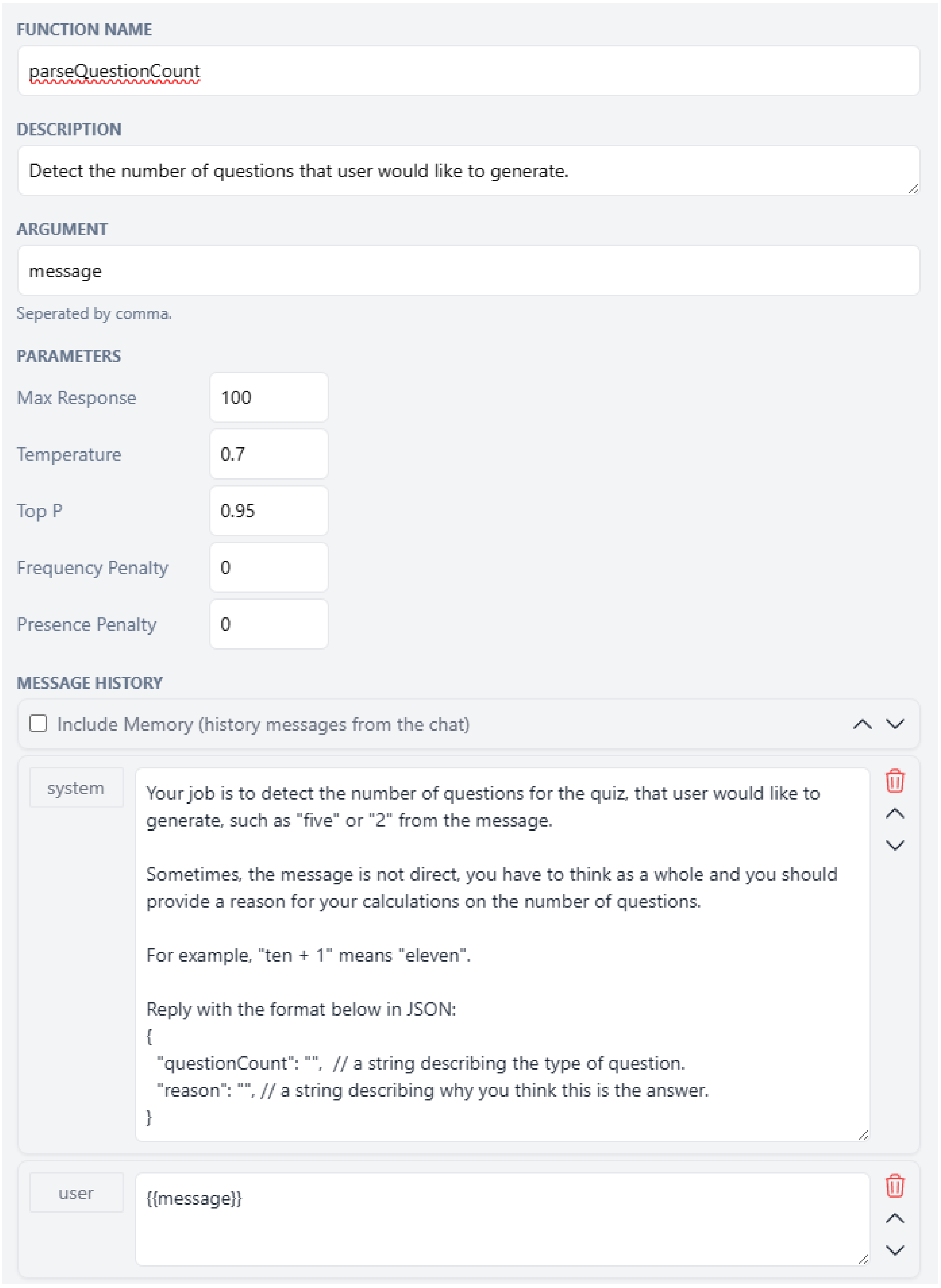
☐ Include Memory (history messages from the chat)

system

```
Based on the message, your job is to detect the type of question for the quiz based on the answer.

For example, "multiple-choice" or "true/false" type of question for the quiz.
message such as "A B C" can also indicate "multiple-choice", "yes no type" can also indicate "true/false" type of question.
You are also required to explain why you have decided the question type.

Reply with the format below in JSON:
{
  "questionType": "",  // a string describing the type of question for the quiz
  "reason": "" // a string describing why it should be this type of question
}
```

user

```
{{message}}
```



FUNCTION NAME

parseQuestionCount

DESCRIPTION

Detect the number of questions that user would like to generate.

ARGUMENT

message

Seperated by comma.

PARAMETERS

Max Response	100
Temperature	0.7
Top P	0.95
Frequency Penalty	0
Presence Penalty	0

MESSAGE HISTORY

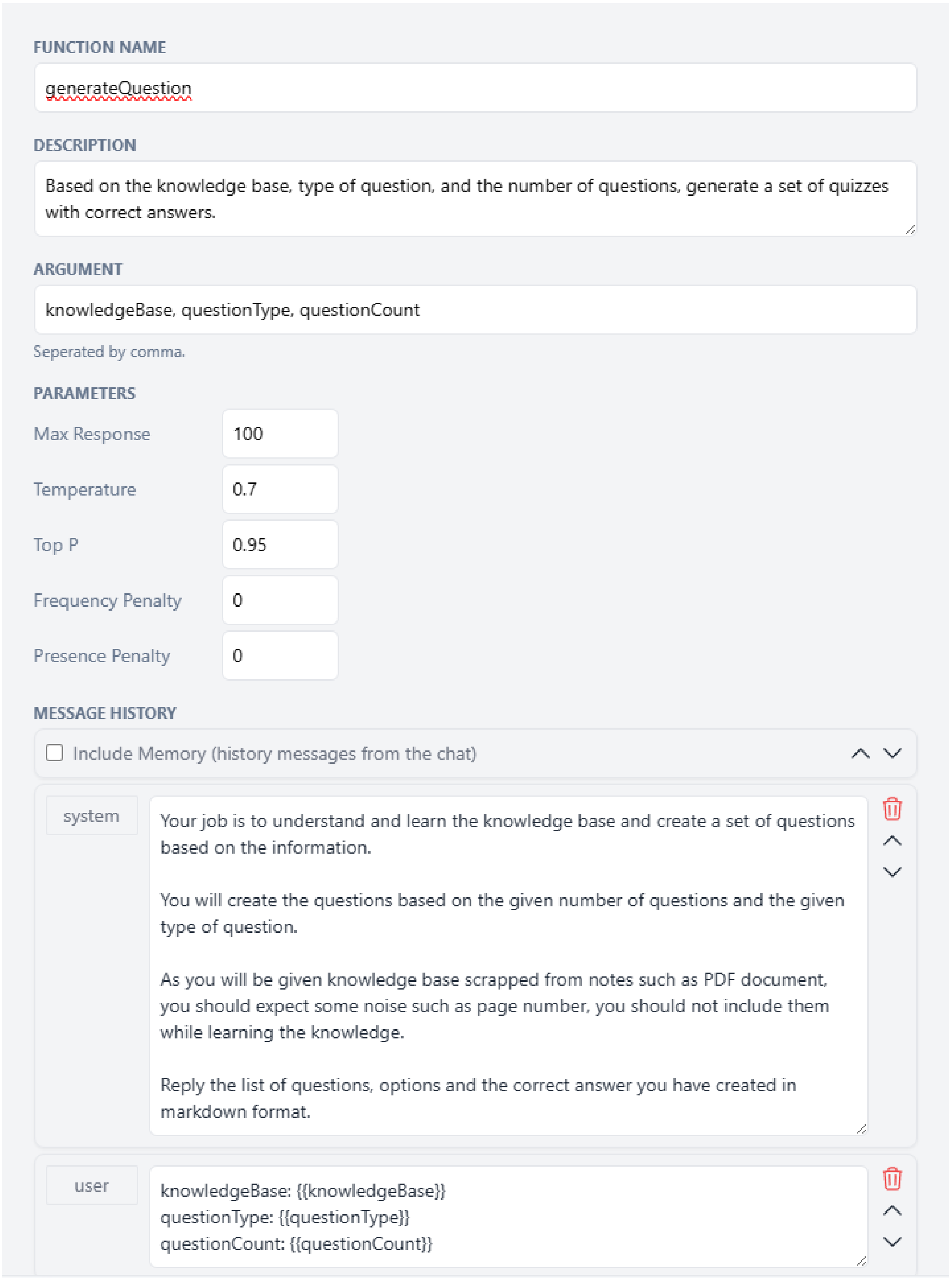
☐ Include Memory (history messages from the chat)

system

```
Your job is to detect the number of questions for the quiz, that user would like to
generate, such as "five" or "2" from the message.

Sometimes, the message is not direct, you have to think as a whole and you should
provide a reason for your calculations on the number of questions.

For example, "ten + 1" means "eleven".

Reply with the format below in JSON:
{
  "questionCount": "",  // a string describing the type of question.
  "reason": "", // a string describing why you think this is the answer.
}
```

user

```
{{message}}
```



FUNCTION NAME

generateQuestion

DESCRIPTION

Based on the knowledge base, type of question, and the number of questions, generate a set of quizzes with correct answers.

ARGUMENT

knowledgeBase, questionType, questionCount

Seperated by comma.

PARAMETERS

Max Response	100
Temperature	0.7
Top P	0.95
Frequency Penalty	0
Presence Penalty	0

MESSAGE HISTORY

☐ Include Memory (history messages from the chat)

system

Your job is to understand and learn the knowledge base and create a set of questions based on the information.

You will create the questions based on the given number of questions and the given type of question.

As you will be given knowledge base scrapped from notes such as PDF document, you should expect some noise such as page number, you should not include them while learning the knowledge.

Reply the list of questions, options and the correct answer you have created in markdown format.

user

knowledgeBase: {{knowledgeBase}}
questionType: {{questionType}}
questionCount: {{questionCount}}