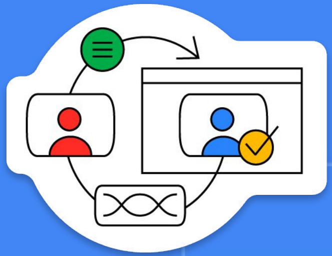
Intro to Gemini

Matt Kaufman He/Him

CEO, MK Partners

@the1mattkaufman



About Me

Matt Kaufman

Founder and CEO, MK Partners
(a Salesforce Consultancy and App Publisher)

12+ year GDG Organizer

Started coding cluelessly at age 4
Coding professionally since 2007

Husband and Father of three

Love puzzles, fixing stuff, and sci-fi

If you can only take one photo of a slide

Visit the cookbook

goo.gle/cookbook

Open Google AI Studio

aistudio.google.com

Open the docs

ai.google.dev/docs

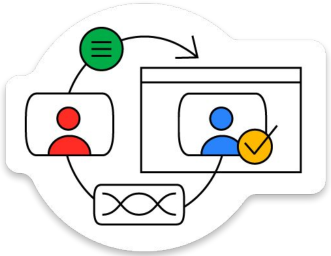
Join Discord

ai.google.dev/docs/discord

Agenda

Start developing

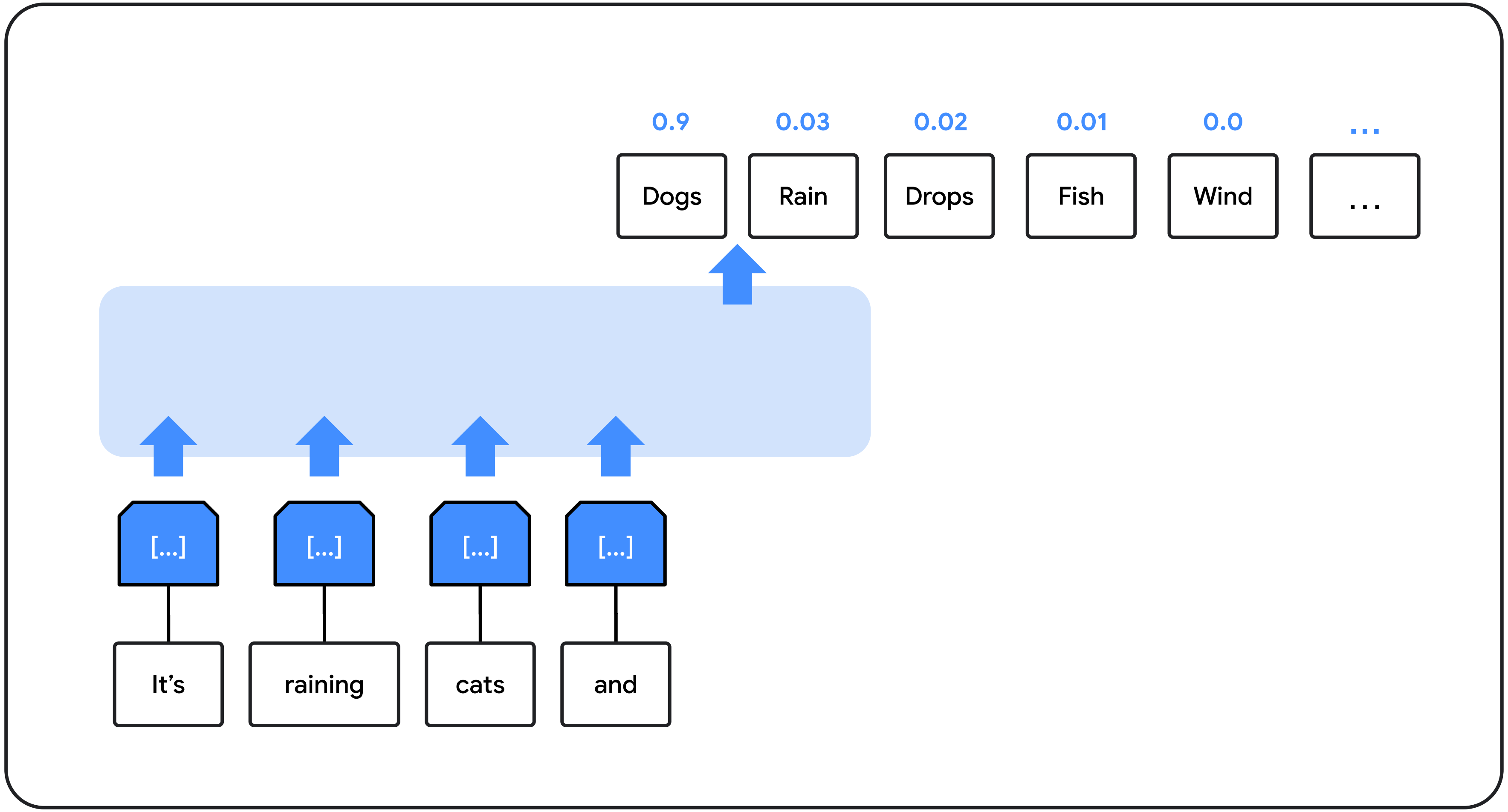
1. Background Knowledge
2. Play with AI Studio
3. Build a NodeJS web service in Project IDX
4. Deploy your Gemini Web Service to Cloud Run



what is **generative ai**?

LLMs Explained

What is an LLM?



Roses are red,
Violets are blue,
Sugar is sweet,

```
for(let i = 0; i < 10; i++) {
```

Modern LLMs are *large*.

Classic Natural Language Problems

- Entity extraction
- Classification
- Summarization
- Sentiment Analysis
- Translation
- ...

LLMs let us prototype *fast*.

User:

Explain this joke: “I used to be addicted to soap, but I'm clean now.”

Model:

Model: The joke is a pun on the word "clean". It has two meanings:

To be free of dirt or other unwanted substances.

To be free of addiction.

The joke is funny because it is unexpected. The listener expects the speaker to be talking about the first meaning of "clean", but they are actually talking about the second meaning. This creates a surprise and a laugh.

User:

Give me five neat ideas for science fair projects.

Model:

1. **Build a volcano.**

This is a classic science fair project that is both fun and educational. You can learn about the science of volcanoes, earthquakes, and plate tectonics.

2. **Make a solar oven.**

This is a great project to learn about renewable energy and how to harness the power of the sun. You can use your solar oven to cook food, dry fruit, or even melt chocolate.

3. **Create a 3D printer.**

This is a more challenging project, but it is also very rewarding. You can learn about 3D printing technology and design your own custom objects.

4. **Build a robot.**

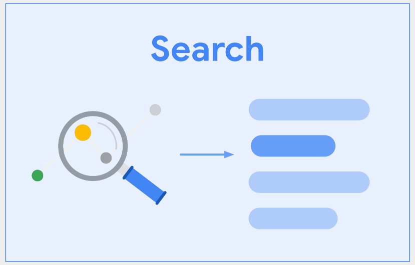
This is a great project for those who are interested in engineering and robotics. You can learn about basic electronics, mechanics, and programming.

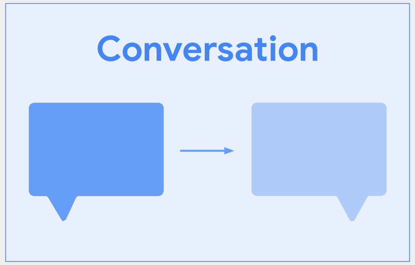
5. **Design a water filtration system.**

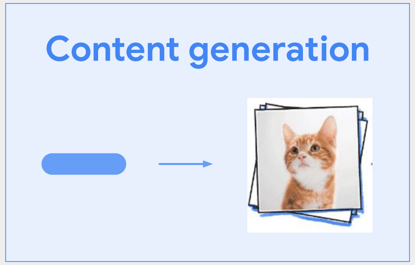
This is a project that can help you learn about the importance of clean water and how to filter it. You can use your water filtration system to clean water from a stream or river.

Why are large language models different?

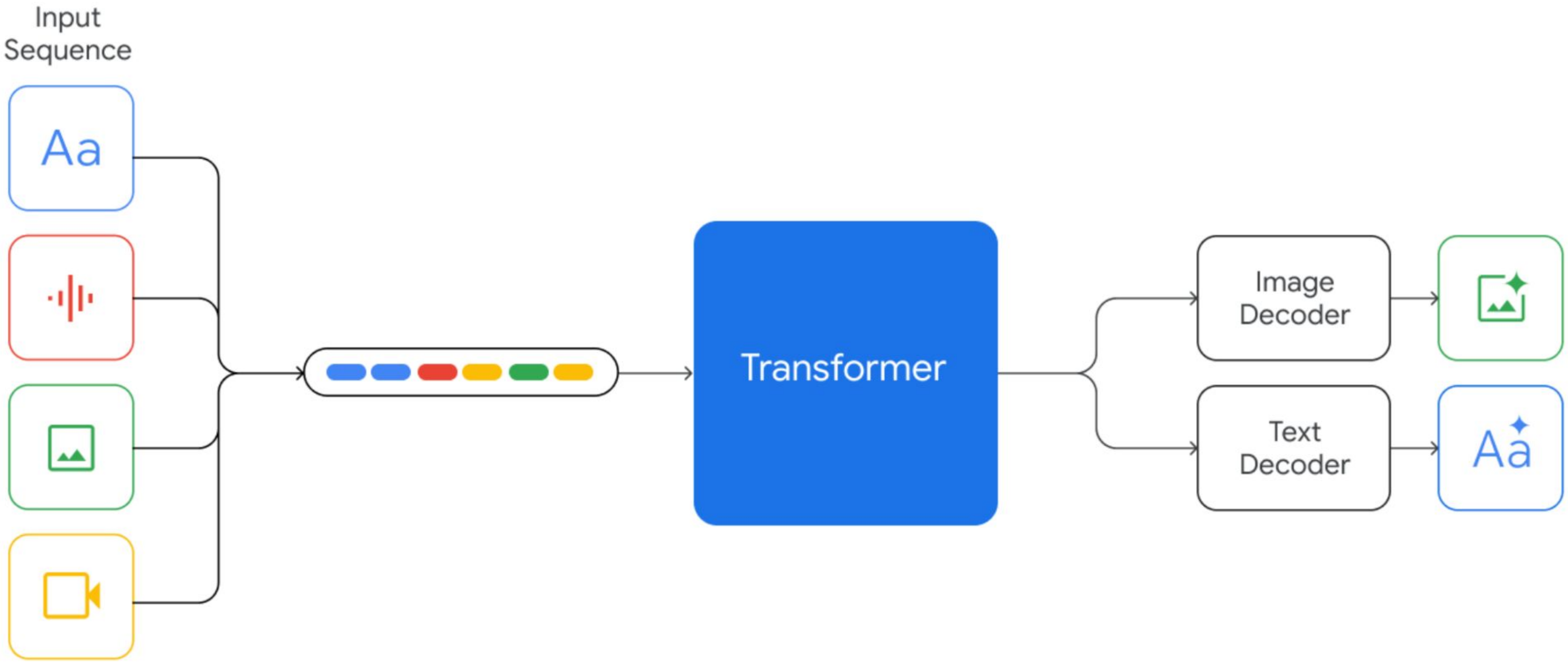
- LLMs are characterized by **emergent abilities**, or the ability to perform tasks that were not present in smaller models.
- LLMs contextual understanding of human language **changes how we interact** with data and intelligent systems.
- LLMs can find patterns and connections in **massive, disparate data corpora.**

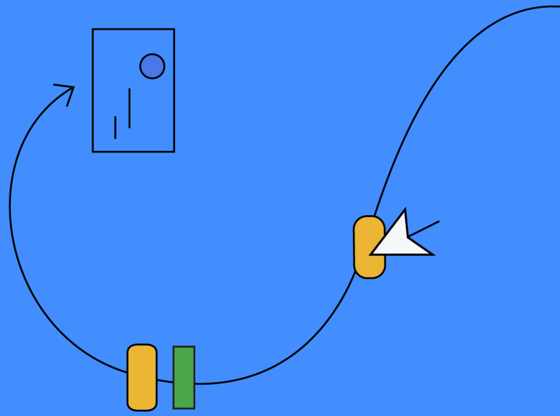

Search


Conversation


Content generation

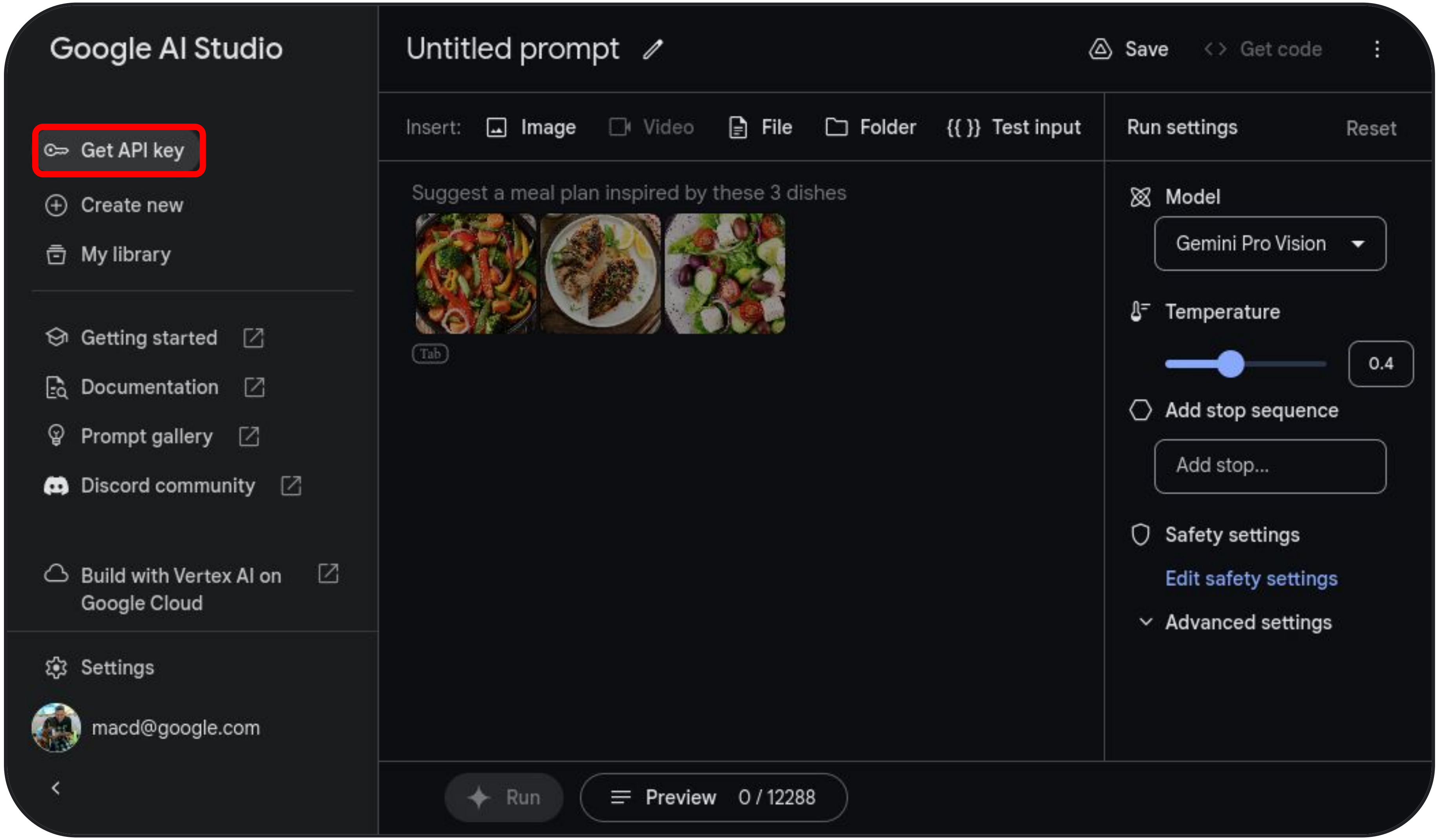
Multimodality





Getting started with the Gemini API

AI Studio



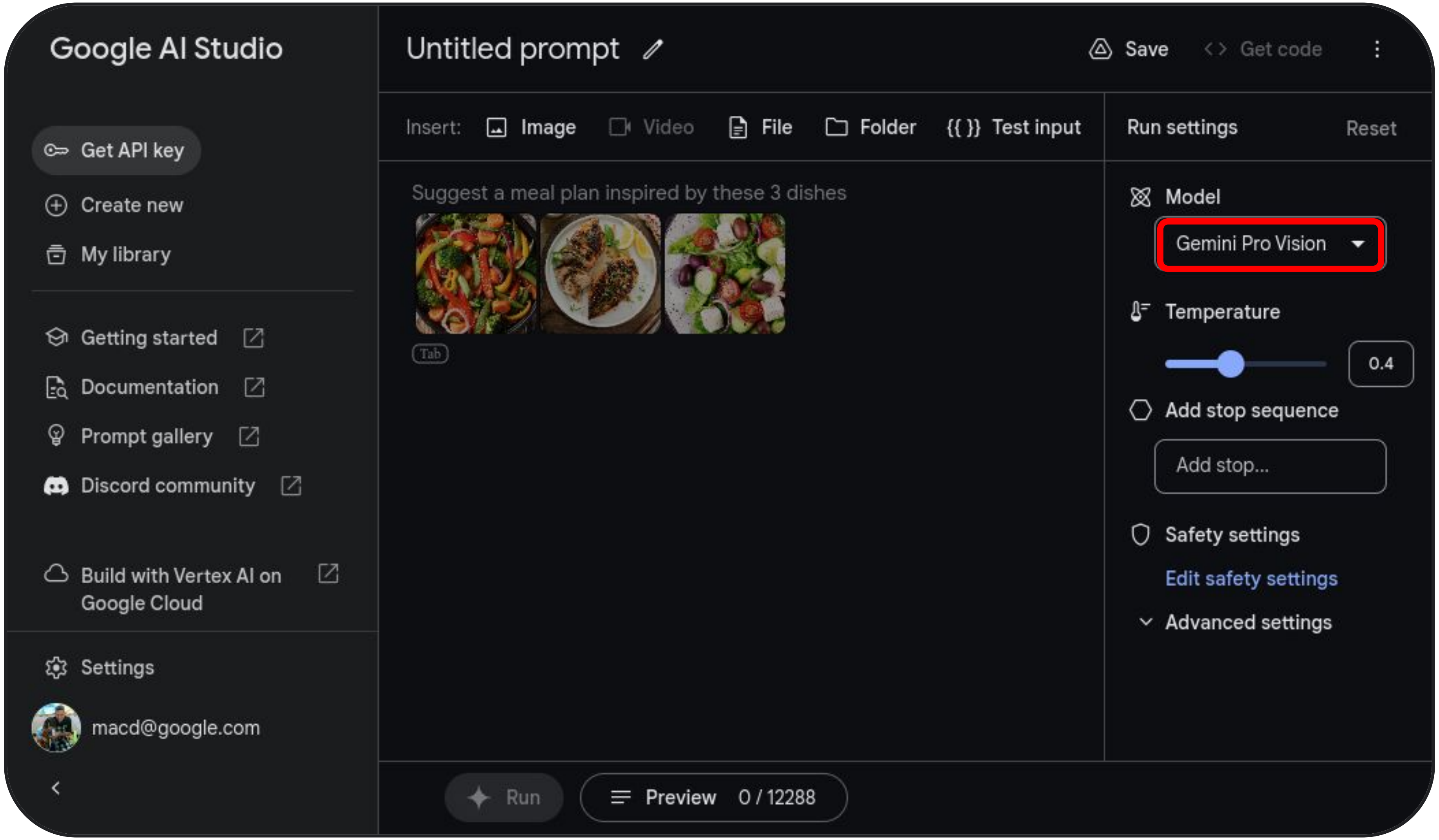
Suggest a meal plan inspired by these 3 dishes

Model: Gemini Pro Vision

Temperature: 0.4

aistudio.google.com

AI Studio



Untitled prompt

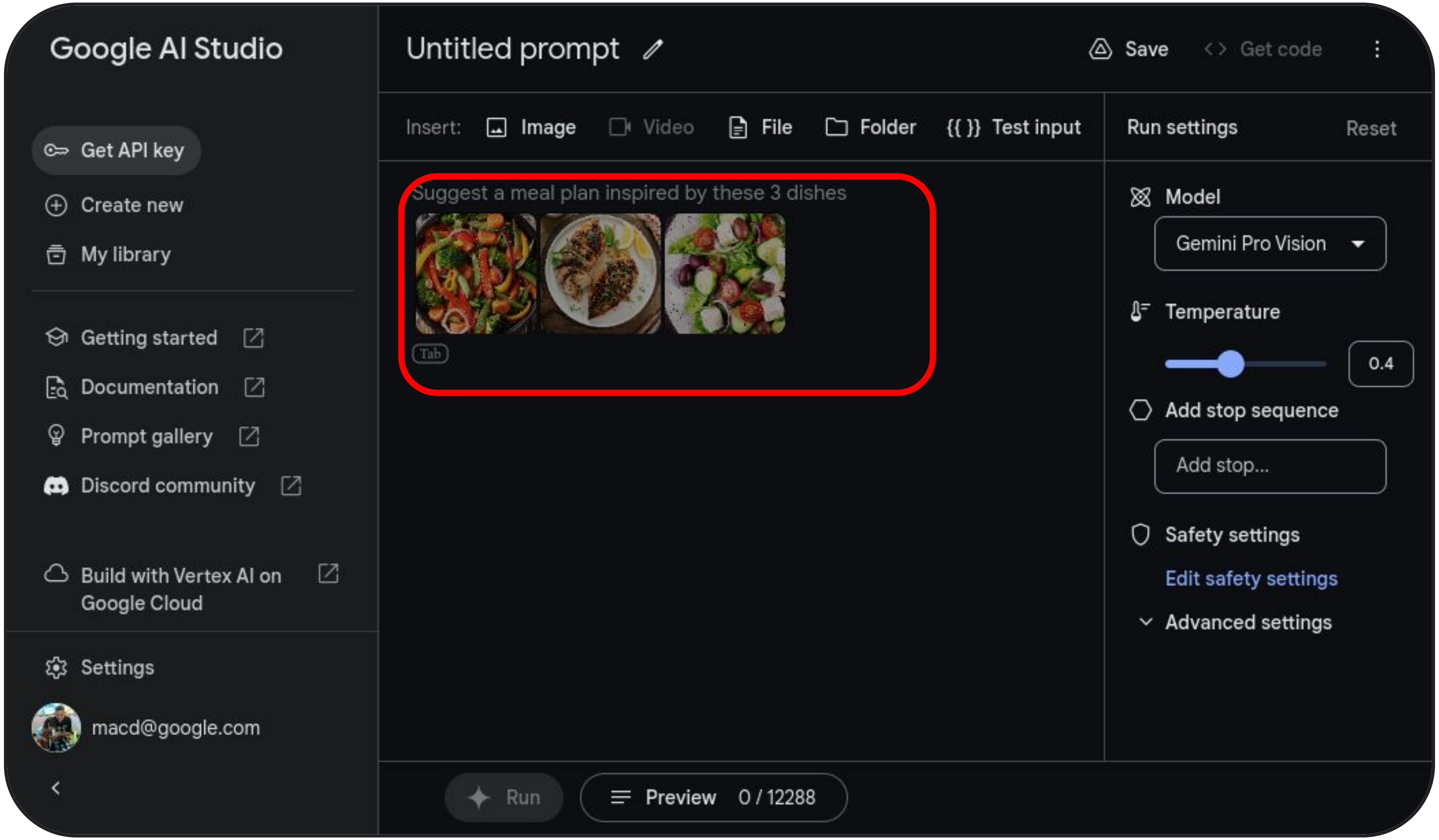
Suggest a meal plan inspired by these 3 dishes

Model: Gemini Pro Vision

Temperature: 0.4

aistudio.google.com

AI Studio



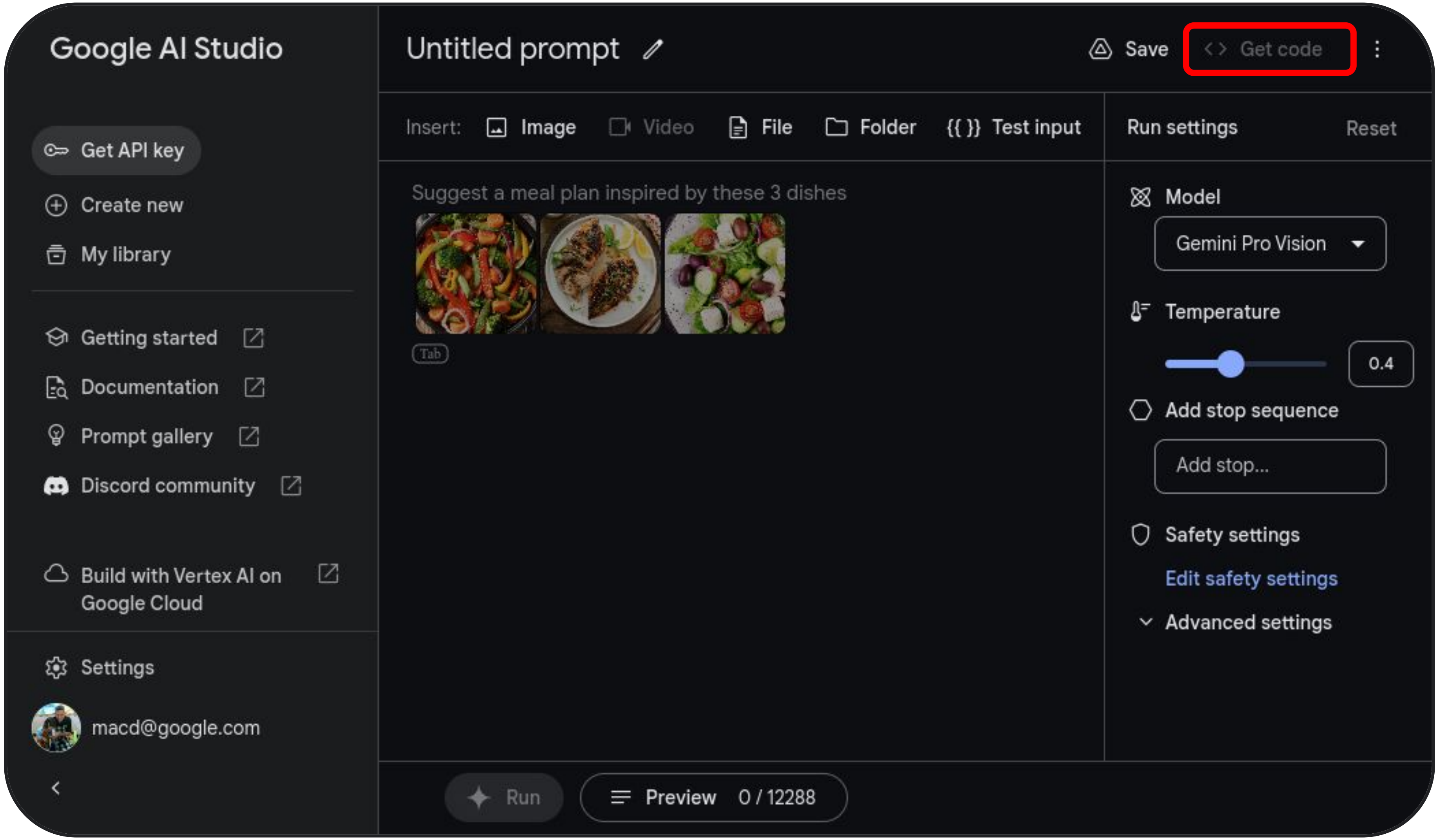
Suggest a meal plan inspired by these 3 dishes

Model: Gemini Pro Vision

Temperature: 0.4

aistudio.google.com

AI Studio



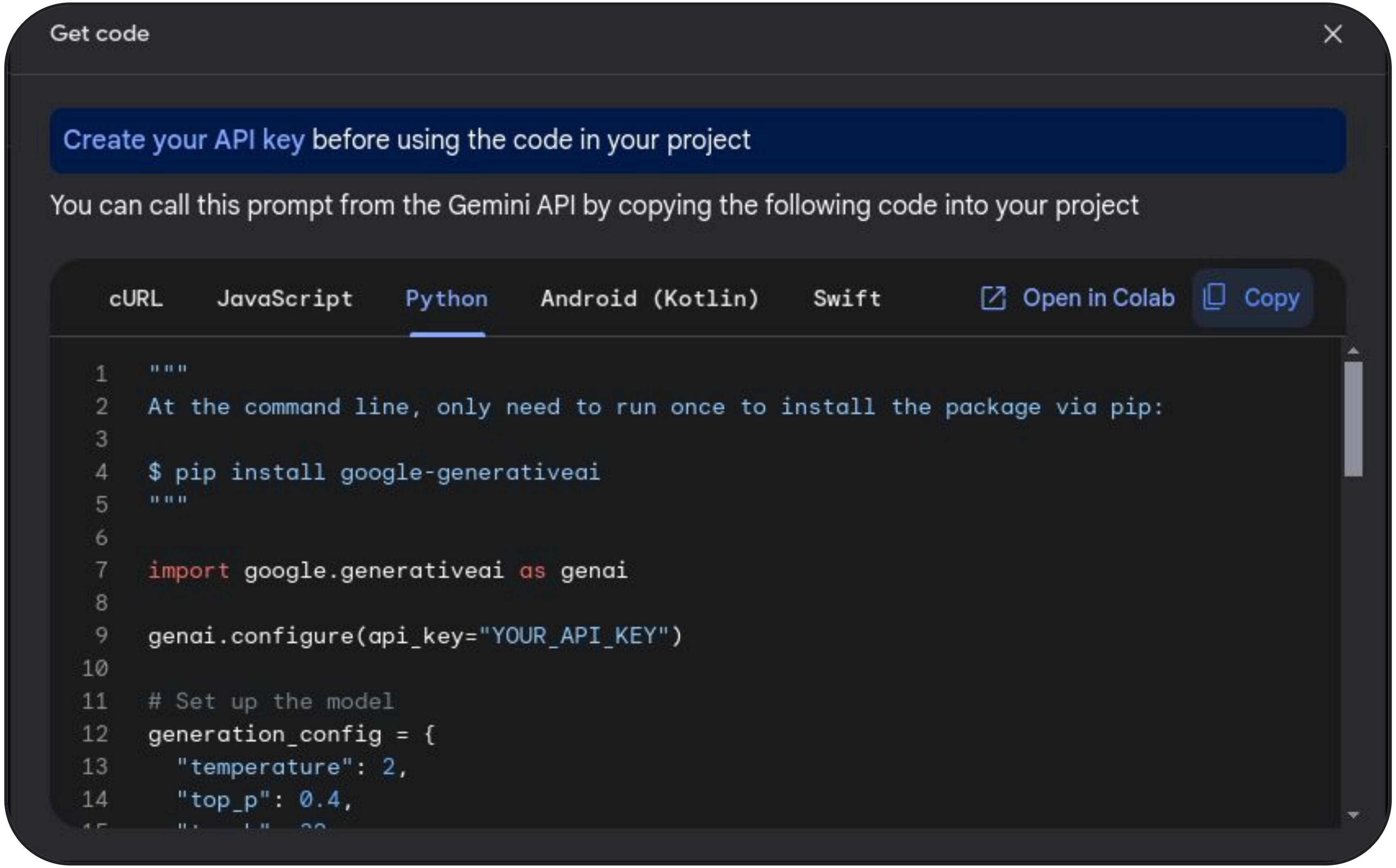
Suggest a meal plan inspired by these 3 dishes

Model: Gemini Pro Vision

Temperature: 0.4

aistudio.google.com

AI Studio



Create your API key before using the code in your project

You can call this prompt from the Gemini API by copying the following code into your project

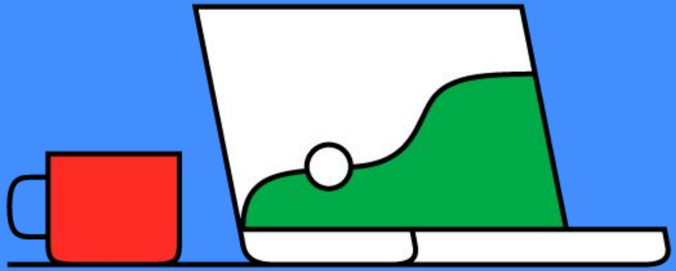
```python
"""
At the command line, only need to run once to install the package via pip:

$ pip install google-generativeai
"""

import google.generativeai as genai

genai.configure(api_key="YOUR_API_KEY")

# Set up the model
generation_config = {
  "temperature": 2,
  "top_p": 0.4,
```



Basic prompting

aistudio.google.com

Write a poem about a magic backpack.

In a world of wonder, behold,
A backpack with tales yet untold.
Its fabric shimmers, alive with might,
A portal to realms, day and night.
...

Generate a short tip about JavaScript. Good tips are short and contain a code example showing the tip.

Tip: Use Array.prototype.flat() to flatten an array of arrays.

```javascript
const array = [[1, 2], [3, 4], [5, 6]];
console.log(array.flat());

// Output: [1, 2, 3, 4, 5, 6]
```

Generate a short tip about JavaScript. Good tips are short and contain a code example showing the tip.

topic: let and const
tip: Prefer using `let` and `const` to initialise variables, instead of `var`, as `var` has no block scope or protection against mutability.

e.g. const myVar = "EXAMPLE";

topic: arrow syntax
tip:

Prompting

- Chained prompts - make a plan, then execute it
- Context - Few shot prompts
- Generation parameters - Temperature, Safety settings, Top-P, TopK

https://ai.google.dev/docs/prompt_best_practices

Code generation

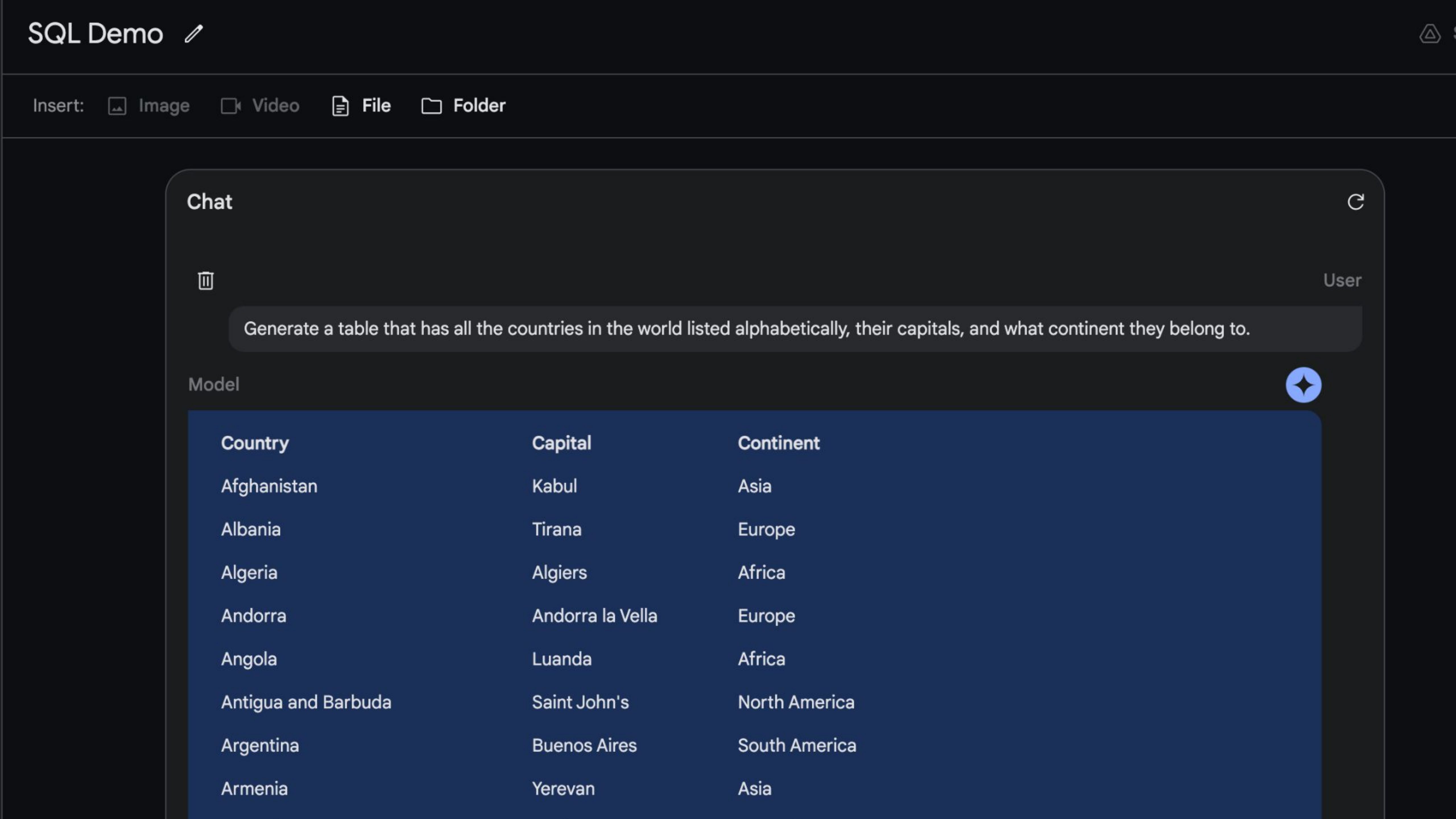
Content generation

Code Generation

Generate code based on a user prompt

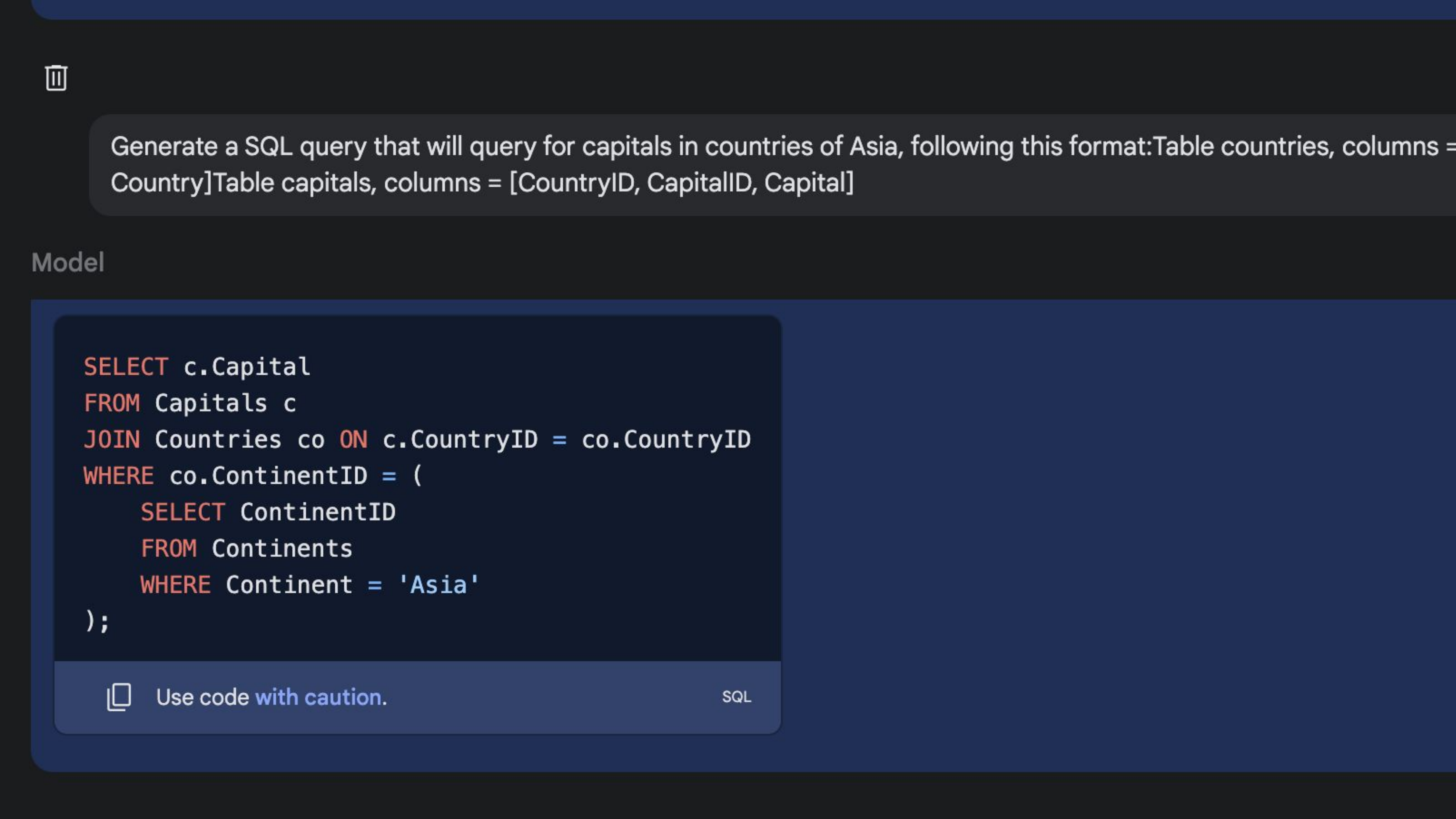
Convert human language requests to machine requests

Rapid tool building



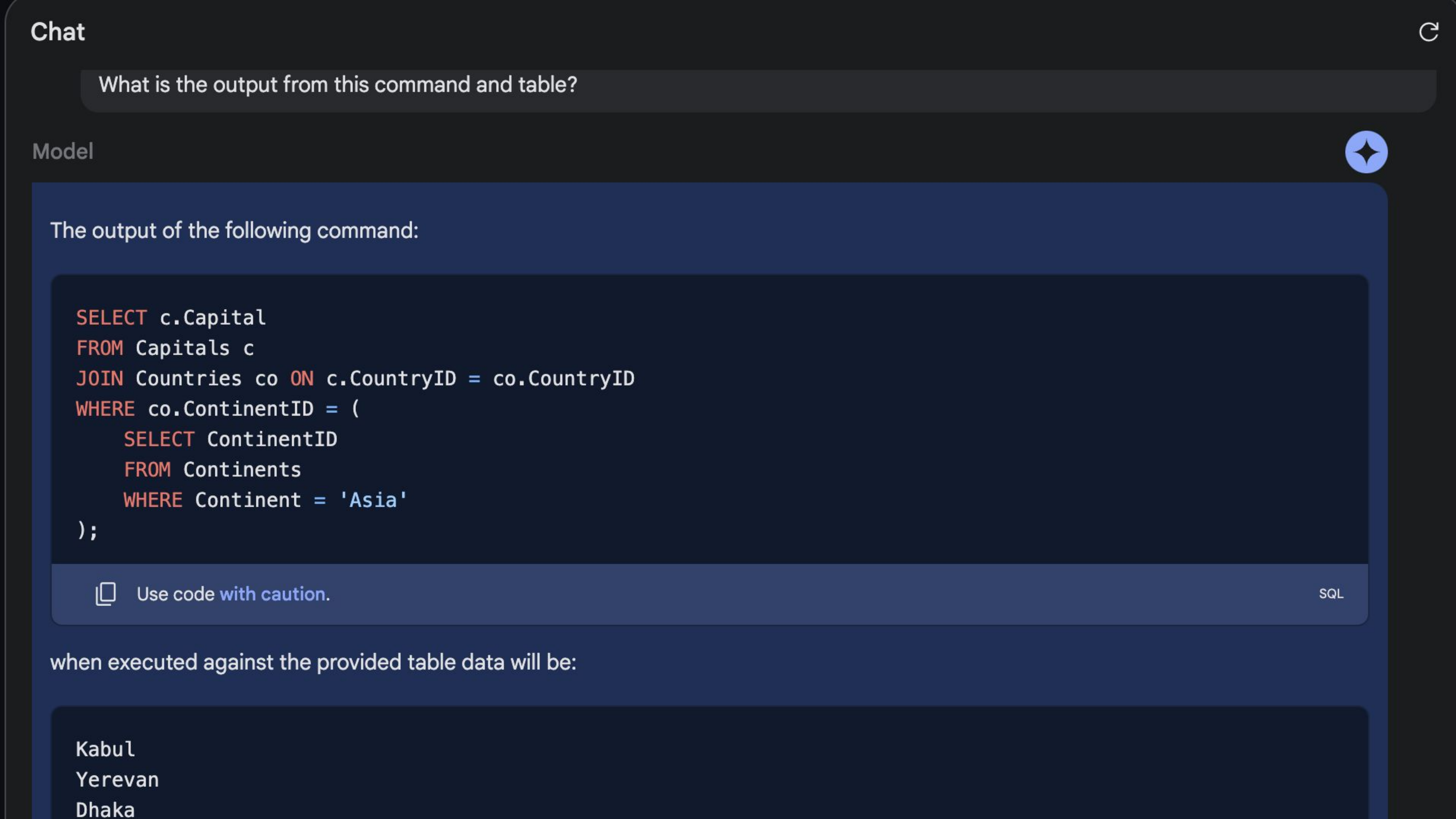
Generate a table that has all the countries in the world listed alphabetically, their capitals, and what continent they belong to.

Country	Capital	Continent
Afghanistan	Kabul	Asia
Albania	Tirana	Europe
Algeria	Algiers	Africa
Andorra	Andorra la Vella	Europe
Angola	Luanda	Africa
Antigua and Barbuda	Saint John's	North America
Argentina	Buenos Aires	South America
Armenia	Yerevan	Asia



Generate a SQL query that will query for capitals in countries of Asia, following this format:Table countries, columns =
Country]Table capitals, columns = [CountryID, CapitalID, Capital]

```sql
SELECT c.Capital
FROM Capitals c
JOIN Countries co ON c.CountryID = co.CountryID
WHERE co.ContinentID = (
    SELECT ContinentID
    FROM Continents
    WHERE Continent = 'Asia'
);
```



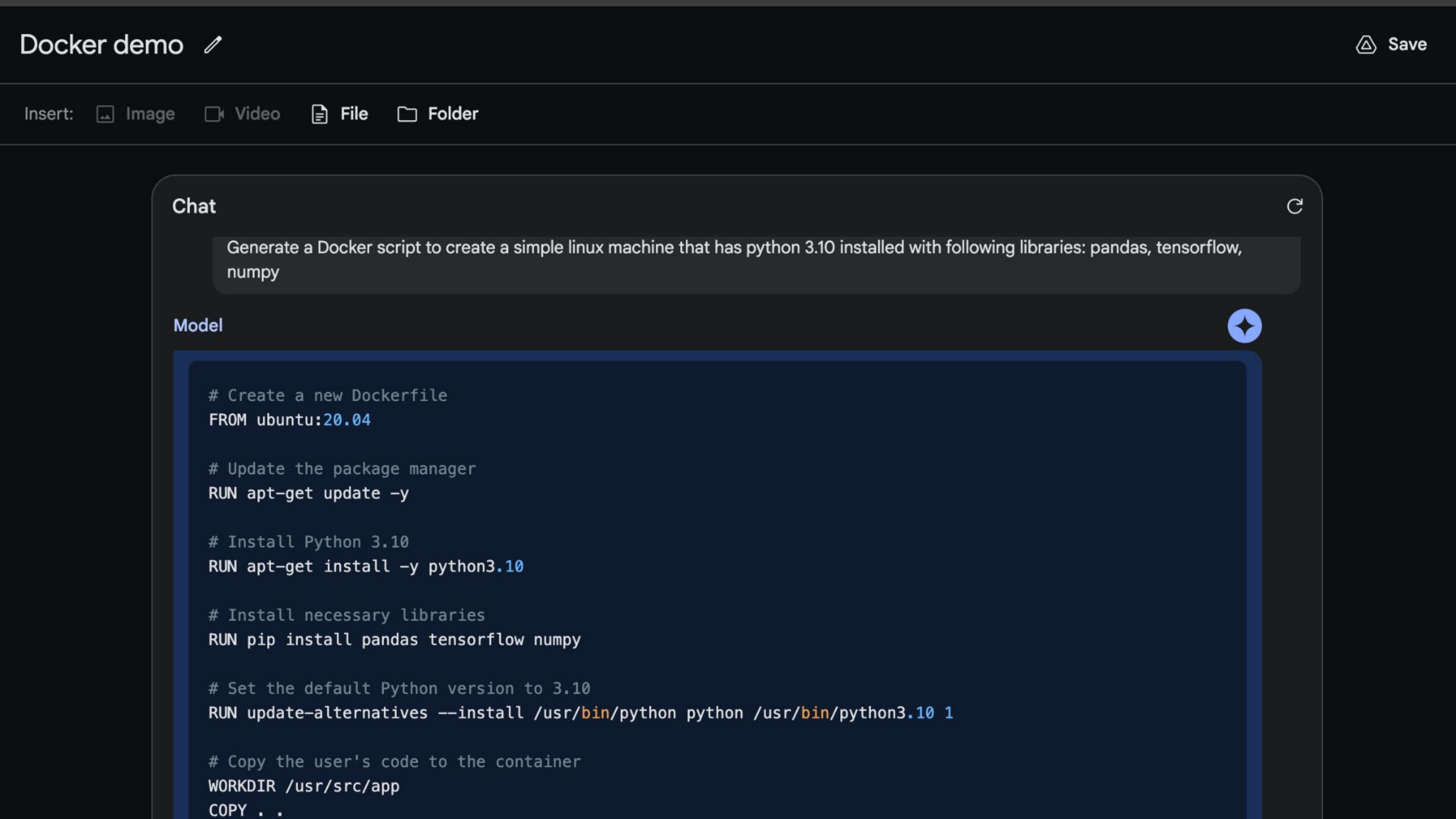
What is the output from this command and table?

The output of the following command:

```sql
SELECT c.Capital
FROM Capitals c
JOIN Countries co ON c.CountryID = co.CountryID
WHERE co.ContinentID = (
    SELECT ContinentID
    FROM Continents
    WHERE Continent = 'Asia'
);
```

when executed against the provided table data will be:

```
Kabul
Yerevan
Dhaka
```



Generate a Docker script to create a simple linux machine that has python 3.10 installed with following libraries: pandas, tensorflow, numpy

```
# Create a new Dockerfile
FROM ubuntu:20.04

# Update the package manager
RUN apt-get update -y

# Install Python 3.10
RUN apt-get install -y python3.10

# Install necessary libraries
RUN pip install pandas tensorflow numpy

# Set the default Python version to 3.10
RUN update-alternatives --install /usr/bin/python python /usr/bin/python3.10 1

# Copy the user's code to the container
WORKDIR /usr/src/app
COPY . .
```

Write a unit test for this Python function:

```python
def is_diver_neutrally_buoyant(density_water, density_diver):
  """Check if a diver is neutrally buoyant."""
  if density_water == density_diver:
    return True
```

```python
import unittest

class IsDiverNeutrallyBuoyantTest(unittest.TestCase):

    def test_diver_is_neutrally_buoyant(self):
        self.assertTrue(is_diver_neutrally_buoyant(1000, 1000))

    def test_diver_is_not_neutrally_buoyant(self):
        self.assertFalse(is_diver_neutrally_buoyant(1000, 1010))

    def test_invalid_input_types(self):
        with self.assertRaises(TypeError):
            is_diver_neutrally_buoyant("1000", 1000)
        with self.assertRaises(TypeError):
            is_diver_neutrally_buoyant(1000, "1000")
```

Search and Information Synthesis

Search & IR

BYO Data

- Models have knowledge cut-offs
- LLMs are not fact engines
- No exposure to private data

Search & IR

Use the prompt's context window

- Instructions + Context + Question all in the prompt
- Easy to implement
    - No extra code, just ask.

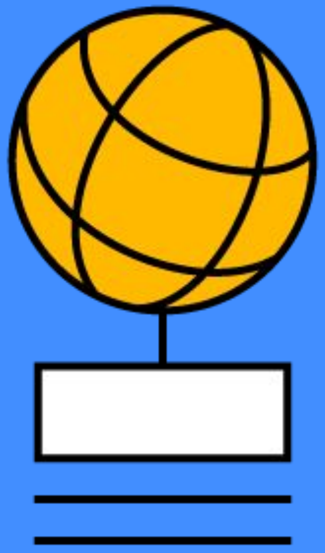
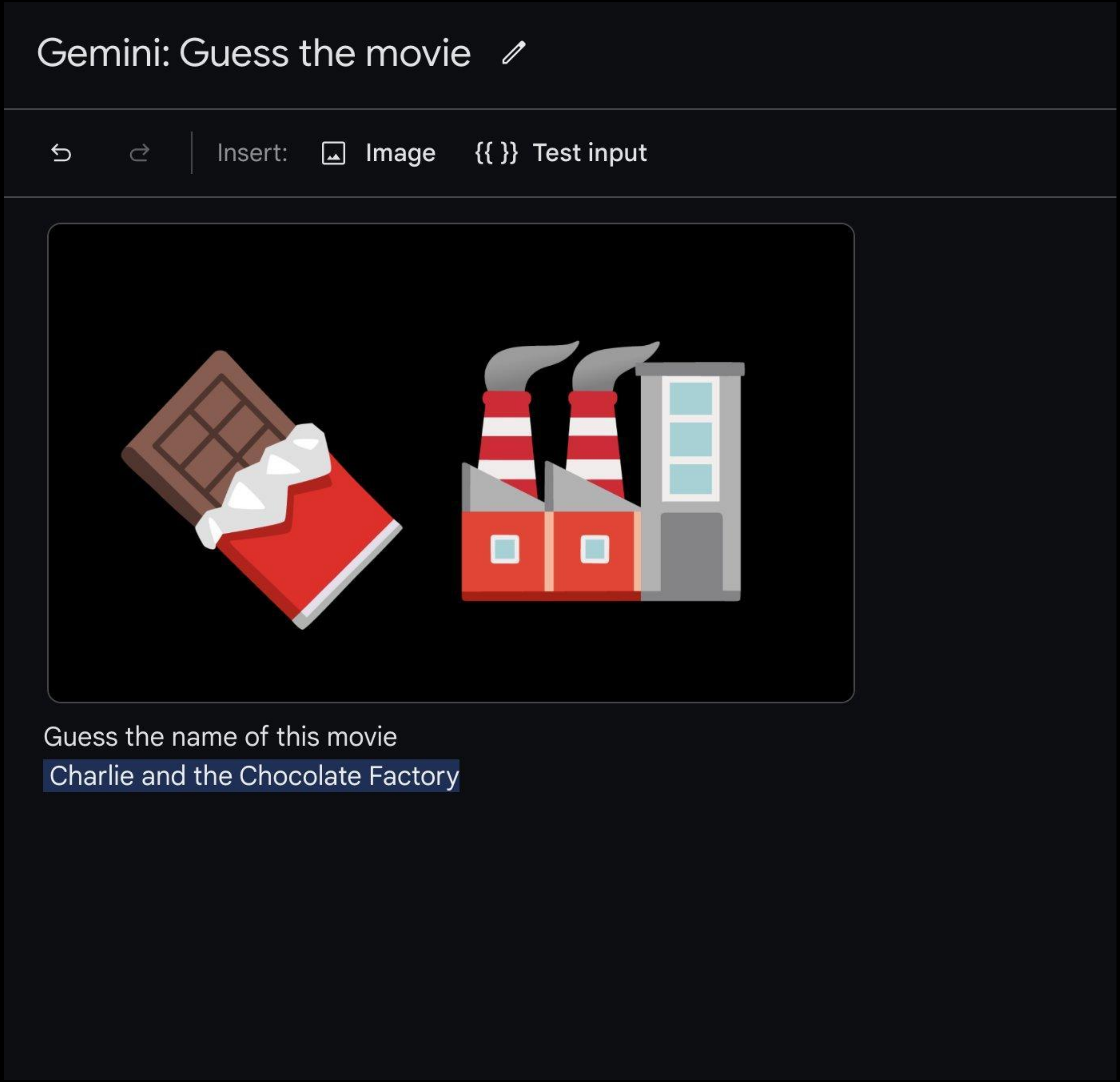
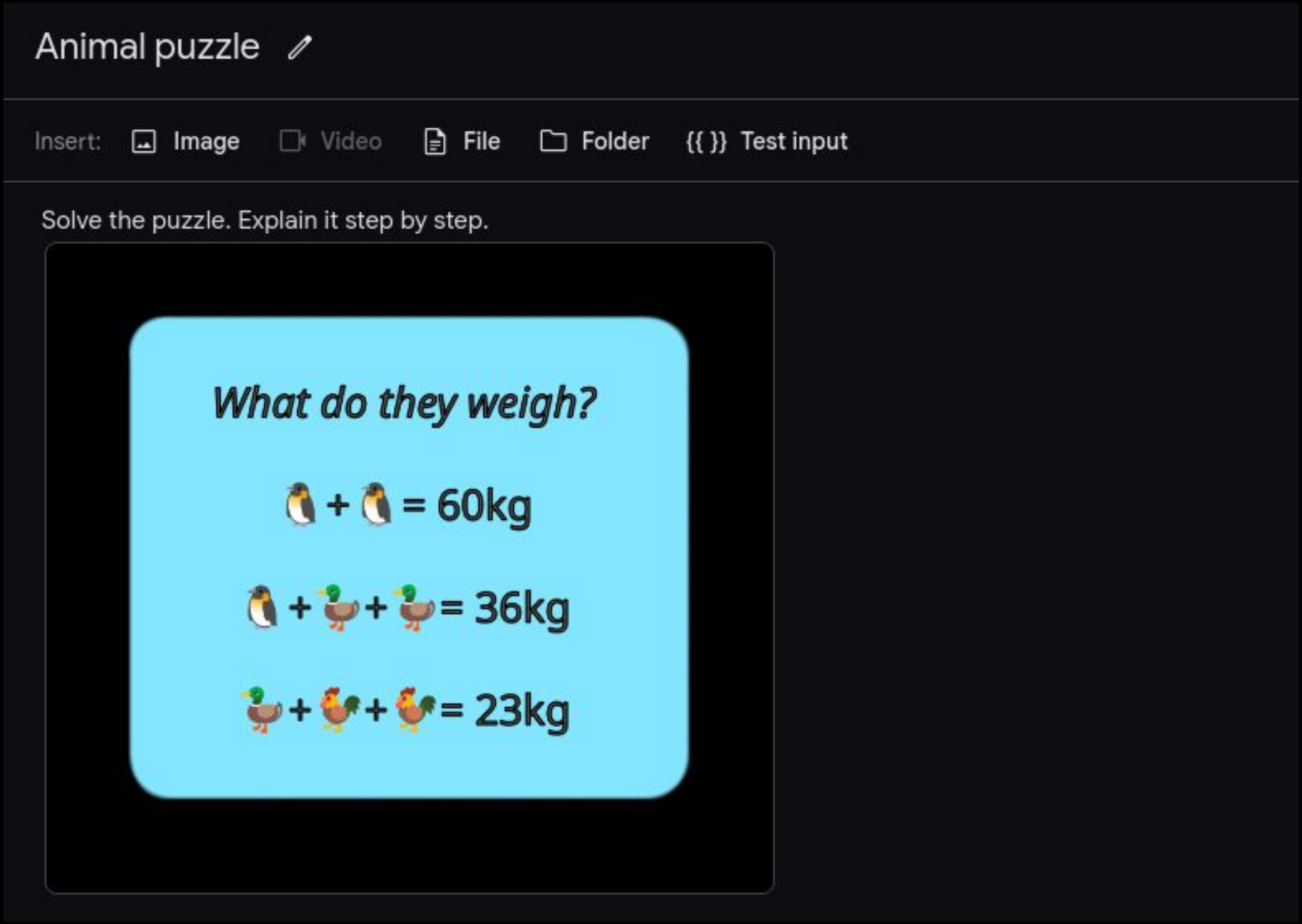


Image understanding



Guess the name of this movie

Charlie and the Chocolate Factory



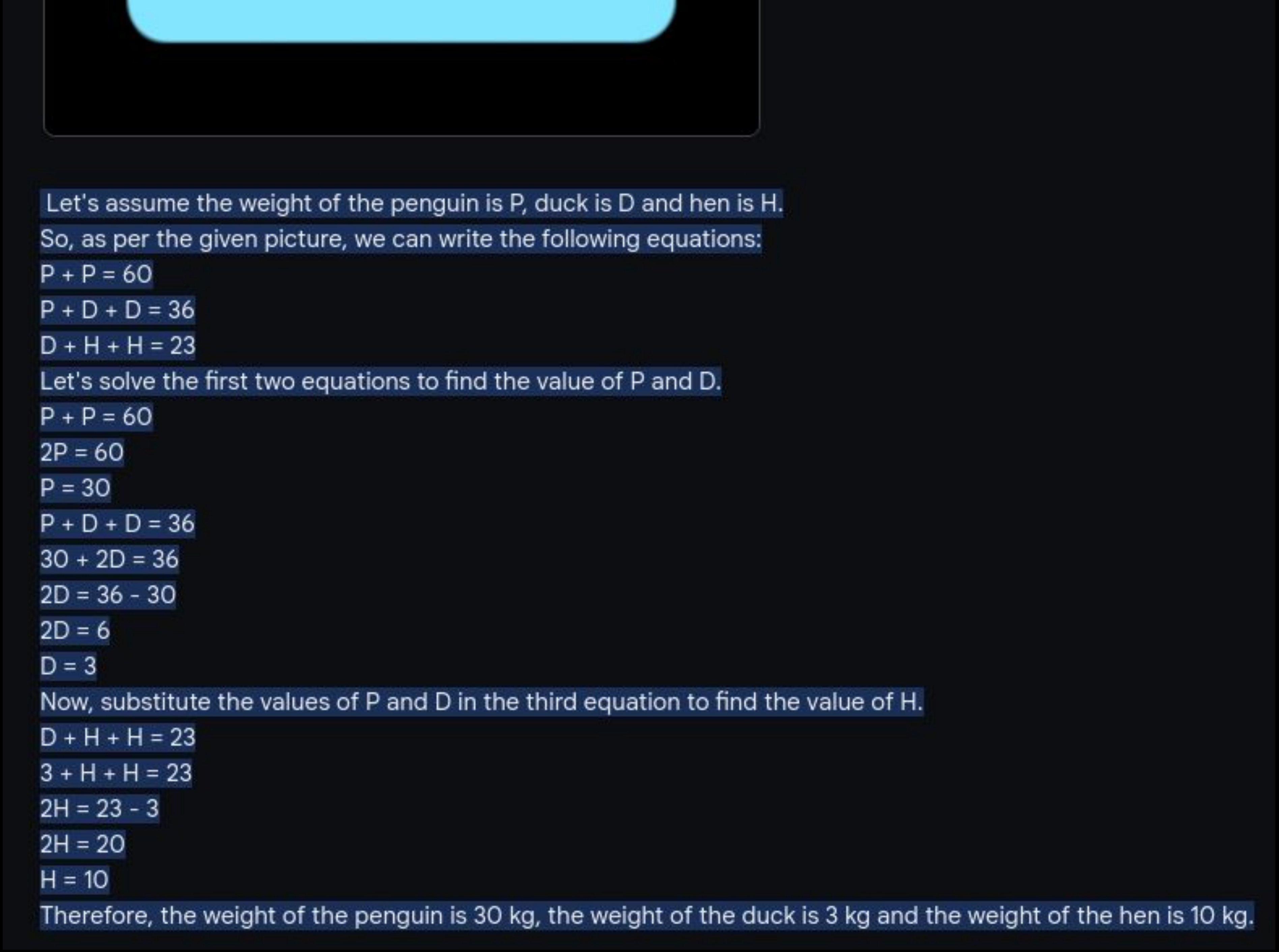
Solve the puzzle. Explain it step by step.

What do they weigh?

🐧 + 🐧 = 60kg

🐧 + 🦆 + 🦆 = 36kg

🦆 + 🐓 + 🐓 = 23kg



Let's assume the weight of the penguin is P, duck is D and hen is H.

So, as per the given picture, we can write the following equations:

$P + P = 60$

$P + D + D = 36$

$D + H + H = 23$

Let's solve the first two equations to find the value of P and D.

$P + P = 60$

$2P = 60$

$P = 30$

$P + D + D = 36$

$30 + 2D = 36$

$2D = 36 - 30$

$2D = 6$

$D = 3$

Now, substitute the values of P and D in the third equation to find the value of H.

$D + H + H = 23$

$3 + H + H = 23$

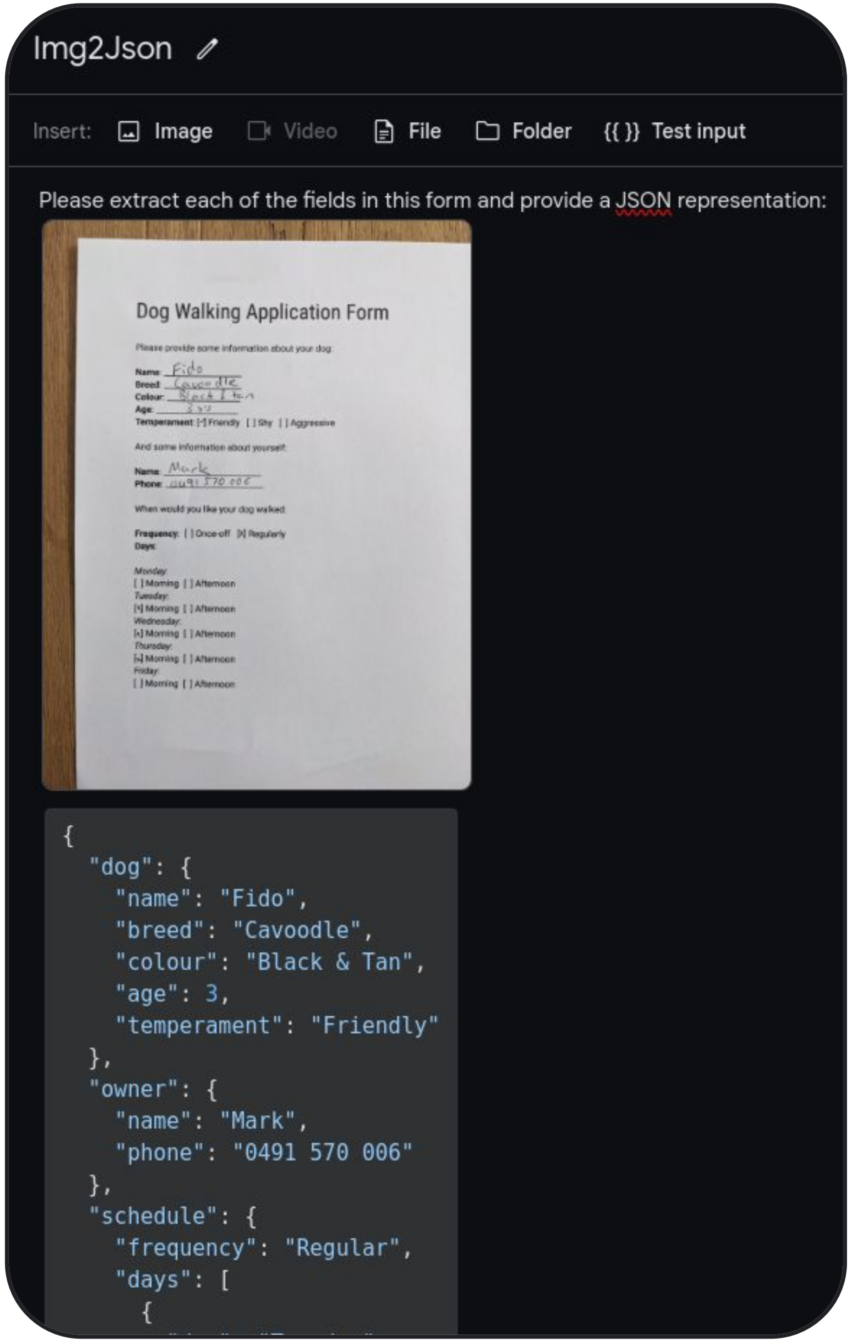
$2H = 23 - 3$

$2H = 20$

$H = 10$

Therefore, the weight of the penguin is 30 kg, the weight of the duck is 3 kg and the weight of the hen is 10 kg.

Structured Data



Please extract each of the fields in this form and provide a JSON representation:

Dog Walking Application Form

Please provide some information about your dog:

Name: Fido
Breed: Cavoodle
Colour: Black & tan
Age: 3 yo
Temperament: ☑ Friendly ☐ Shy ☐ Aggressive

And some information about yourself:

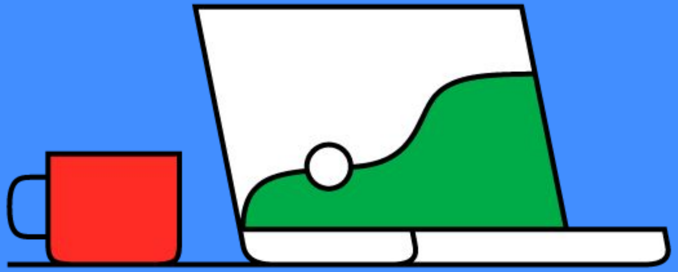
Name: Mark
Phone: 0491 570 006

When would you like your dog walked:

Frequency: ☐ Once-off ☒ Regularly
Days:

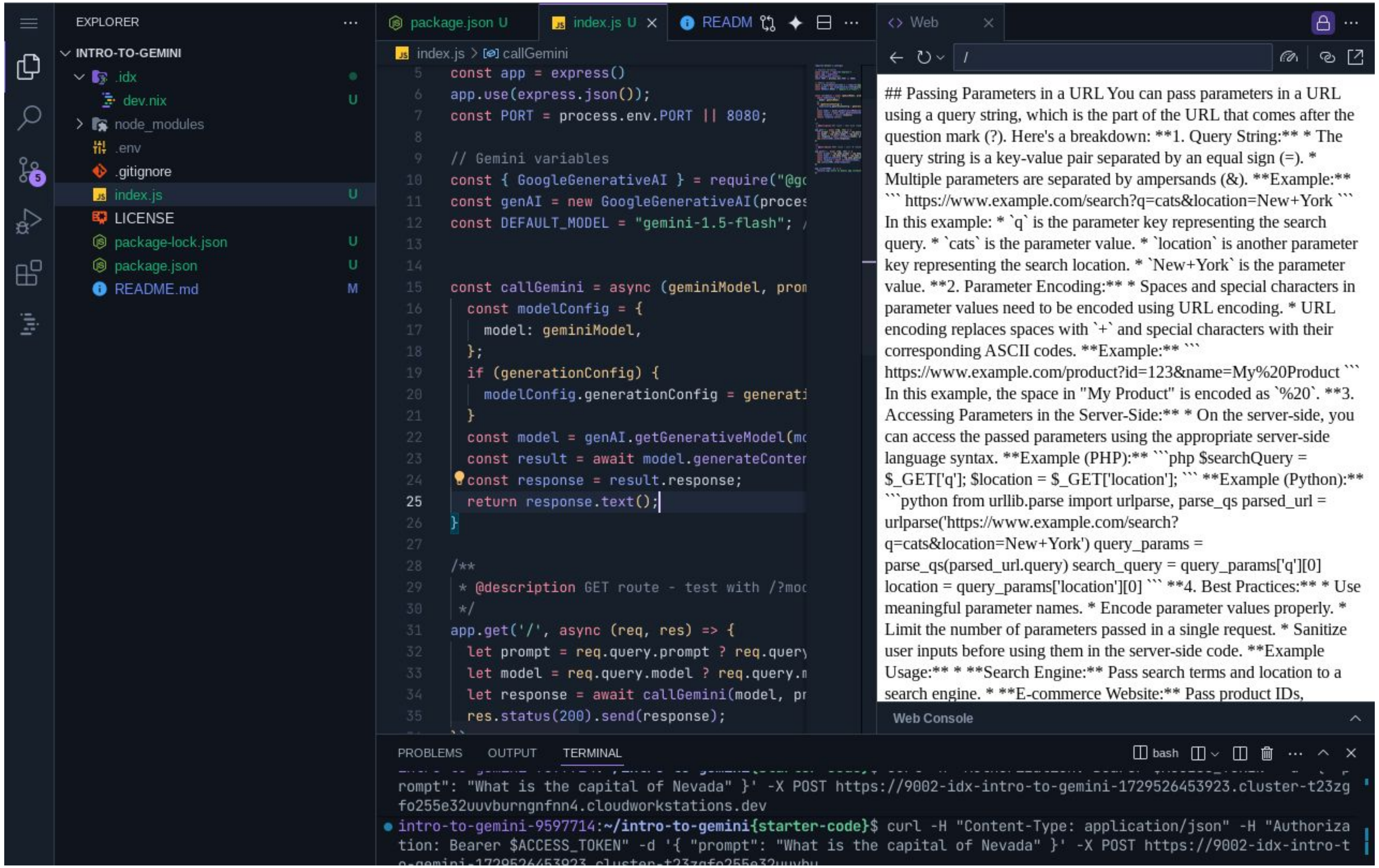
Monday:
☐ Morning ☐ Afternoon
Tuesday:
☒ Morning ☐ Afternoon
Wednesday:
☒ Morning ☐ Afternoon
Thursday:
☒ Morning ☐ Afternoon
Friday:
☐ Morning ☐ Afternoon

```
{
  "dog": {
    "name": "Fido",
    "breed": "Cavoodle",
    "colour": "Black & Tan",
    "age": 3,
    "temperament": "Friendly"
  },
  "owner": {
    "name": "Mark",
    "phone": "0491 570 006"
  },
  "schedule": {
    "frequency": "Regular",
    "days": [
      {
```



Creating a Web Service

Project IDX



```
const app = express()
app.use(express.json());
const PORT = process.env.PORT || 8080;

// Gemini variables
const { GoogleGenerativeAI } = require("@gc
const genAI = new GoogleGenerativeAI(proces
const DEFAULT_MODEL = "gemini-1.5-flash"; /


const callGemini = async (geminiModel, prom
  const modelConfig = {
    model: geminiModel,
  };
  if (generationConfig) {
    modelConfig.generationConfig = generati
  }
  const model = genAI.getGenerativeModel(mc
  const result = await model.generateConter
  const response = result.response;
  return response.text();
}

/**
 * @description GET route - test with /?moc
 */
app.get('/', async (req, res) => {
  let prompt = req.query.prompt ? req.query
  let model = req.query.model ? req.query.m
  let response = await callGemini(model, pr
  res.status(200).send(response);
```

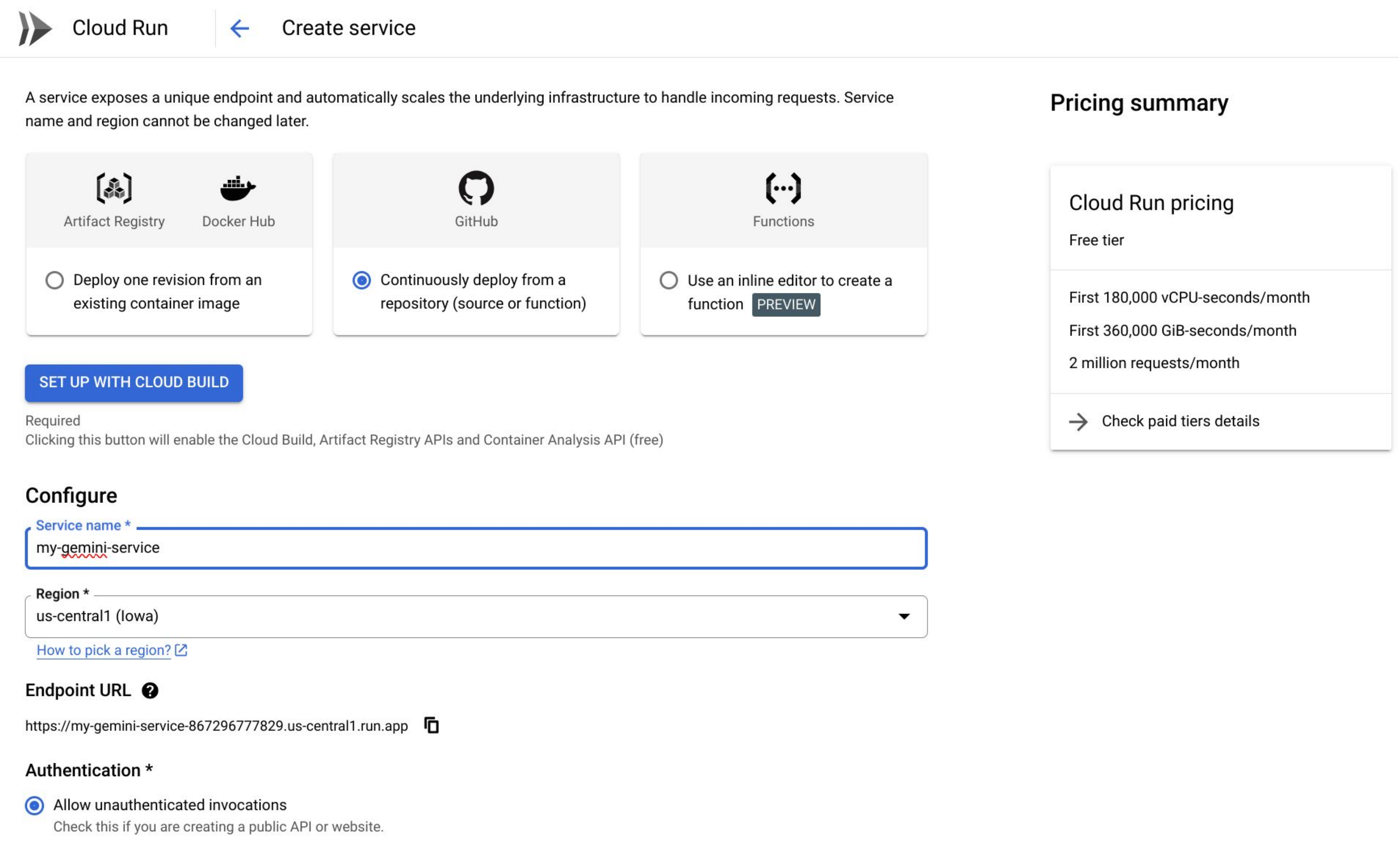
Passing Parameters in a URL You can pass parameters in a URL using a query string, which is the part of the URL that comes after the question mark (?). Here's a breakdown: **1. Query String:** * The query string is a key-value pair separated by an equal sign (=). * Multiple parameters are separated by ampersands (&). **Example:** ``` https://www.example.com/search?q=cats&location=New+York ``` In this example: * `q` is the parameter key representing the search query. * `cats` is the parameter value. * `location` is another parameter key representing the search location. * `New+York` is the parameter value. **2. Parameter Encoding:** * Spaces and special characters in parameter values need to be encoded using URL encoding. * URL encoding replaces spaces with `+` and special characters with their corresponding ASCII codes. **Example:** ``` https://www.example.com/product?id=123&name=My%20Product ``` In this example, the space in "My Product" is encoded as `%20`. **3. Accessing Parameters in the Server-Side:** * On the server-side, you can access the passed parameters using the appropriate server-side language syntax. **Example (PHP):** ```php $searchQuery = $_GET['q']; $location = $_GET['location']; ``` **Example (Python):** ```python from urllib.parse import urlparse, parse_qs parsed_url = urlparse('https://www.example.com/search?q=cats&location=New+York') query_params = parse_qs(parsed_url.query) search_query = query_params['q'][0] location = query_params['location'][0] ``` **4. Best Practices:** * Use meaningful parameter names. * Encode parameter values properly. * Limit the number of parameters passed in a single request. * Sanitize user inputs before using them in the server-side code. **Example Usage:** * **Search Engine:** Pass search terms and location to a search engine. * **E-commerce Website:** Pass product IDs,

```
rompt": "What is the capital of Nevada" }' -X POST https://9002-idx-intro-to-gemini-1729526453923.cluster-t23zg
fo255e32uuvburngnfnn4.cloudworkstations.dev
intro-to-gemini-9597714:~/intro-to-gemini{starter-code}$ curl -H "Content-Type: application/json" -H "Authoriza
tion: Bearer $ACCESS_TOKEN" -d '{ "prompt": "What is the capital of Nevada" }' -X POST https://9002-idx-intro-t
```

```
app.get('/', async (req, res) => {
  const model = genAI.getGenerativeModel({
    model: "gemini-1.5-flash"
  });
  const result = await model.generateContent(
    "What is the capital of California"
  );
  const response = result.response;
  return response.text();
})
```

Cloud Run

- Create a new Cloud Run Service
- Configure it to Build from your Repo
- Commits will update your service



A service exposes a unique endpoint and automatically scales the underlying infrastructure to handle incoming requests. Service name and region cannot be changed later.

○ Deploy one revision from an existing container image

● Continuously deploy from a repository (source or function)

○ Use an inline editor to create a function PREVIEW

Required
Clicking this button will enable the Cloud Build, Artifact Registry APIs and Container Analysis API (free)

Configure

Service name *: my-gemini-service

Region *: us-central1 (Iowa)

Endpoint URL

https://my-gemini-service-867296777829.us-central1.run.app

Authentication *

● Allow unauthenticated invocations
Check this if you are creating a public API or website.

Pricing summary

Cloud Run pricing

Free tier

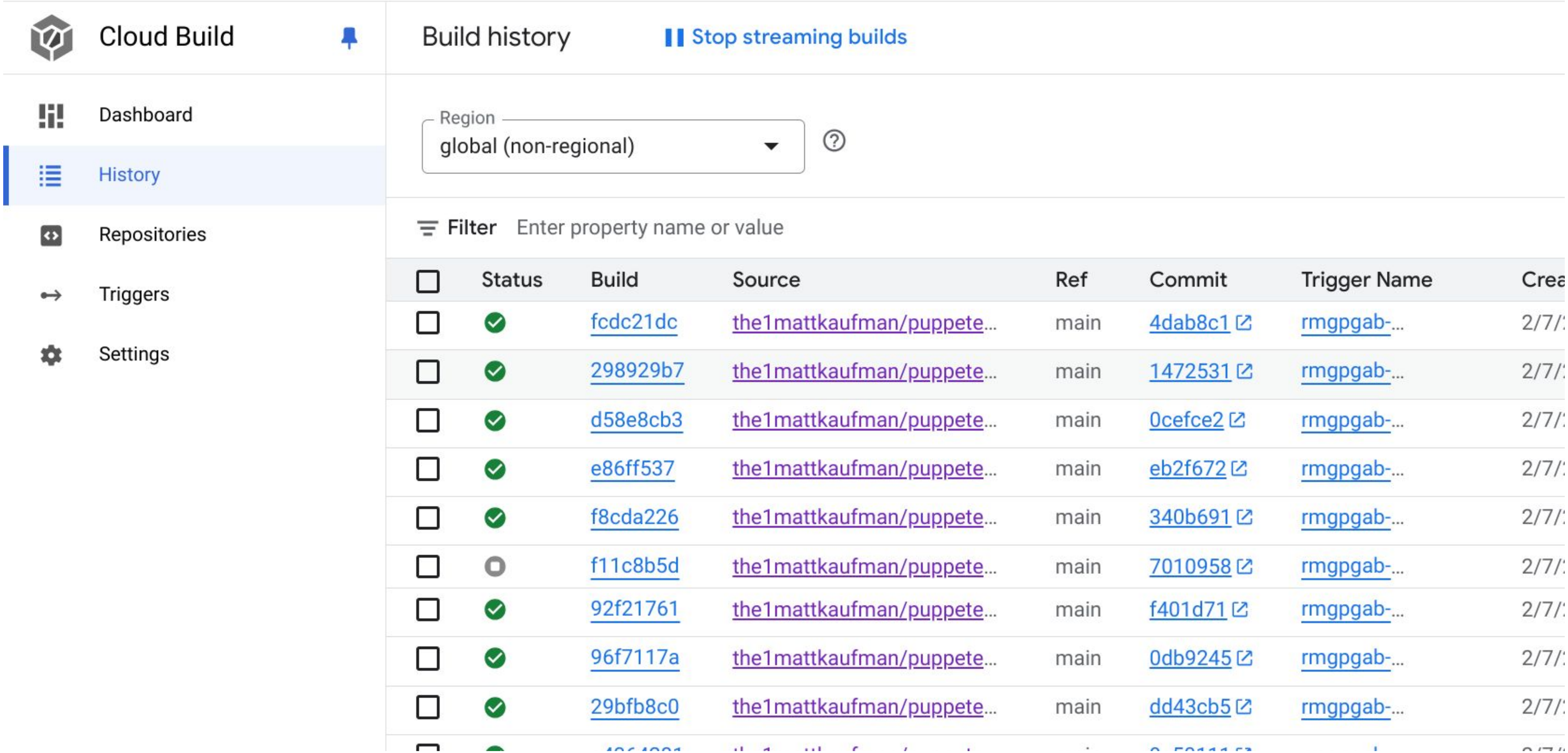
First 180,000 vCPU-seconds/month

First 360,000 GiB-seconds/month

2 million requests/month

→ Check paid tiers details

Cloud Build

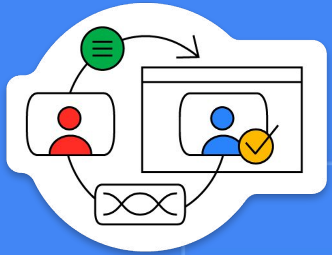


Region: global (non-regional)

☐	Status	Build	Source	Ref	Commit	Trigger Name	Crea
☐	✓	fcdc21dc	the1mattkaufman/puppete...	main	4dab8c1	rmgpgab-...	2/7/
☐	✓	298929b7	the1mattkaufman/puppete...	main	1472531	rmgpgab-...	2/7/
☐	✓	d58e8cb3	the1mattkaufman/puppete...	main	0cefce2	rmgpgab-...	2/7/
☐	✓	e86ff537	the1mattkaufman/puppete...	main	eb2f672	rmgpgab-...	2/7/
☐	✓	f8cda226	the1mattkaufman/puppete...	main	340b691	rmgpgab-...	2/7/
☐	○	f11c8b5d	the1mattkaufman/puppete...	main	7010958	rmgpgab-...	2/7/
☐	✓	92f21761	the1mattkaufman/puppete...	main	f401d71	rmgpgab-...	2/7/
☐	✓	96f7117a	the1mattkaufman/puppete...	main	0db9245	rmgpgab-...	2/7/
☐	✓	29bfb8c0	the1mattkaufman/puppete...	main	dd43cb5	rmgpgab-...	2/7/

Train the Trainer in AI

Thank you!



Matt Kaufman He/Him

CEO, MK Partners

@the1mattkaufman