

Gemini AI App – Data Provider

Overview

The **Gemini AI App – Data Provider** is a lightweight HTML-based web application that enables users to send **custom text prompts** to **Google’s Gemini API** and receive AI-generated responses tailored to business or analytical needs.

It is designed for internal use or testing, allowing developers and business users to interact directly with **Gemini 1.5 Flash**, adjusting parameters such as temperature, top-p, top-k, and output length to fine-tune responses.

No database is required — the system relies entirely on real-time API requests and client-side rendering.

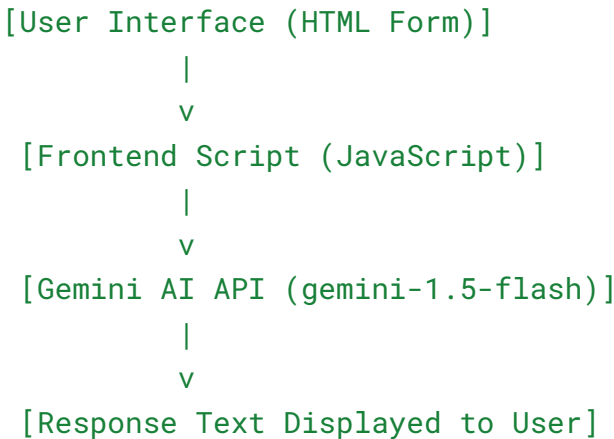
Architecture

High-Level Flow

1. **User Interface (Frontend):**
 - A clean, single-page HTML form.
 - Users input generation parameters (Max Tokens, Temperature, Top P, Top K, Model).
 - Prompts are entered in a text area labeled “Write any prompt for your business needs...”
2. **API Request Layer:**
 - JavaScript handles user inputs and sends an HTTP POST request to the **Gemini API** endpoint.
 - The model used is **gemini-1.5-flash** by default.
3. **Response Handling:**

- The Gemini API processes the input prompt and returns a **text-based completion**.
- The app displays the generated result on the same page.

System Diagram



Tech Stack

Component	Technology
Frontend	HTML, CSS, JavaScript
AI API	Google Gemini API
Model	gemini-1.5-flash
Backend (optional)	Node.js, Flask, or PHP (for API key proxying)
Database	None
Deployment	Any static web host or local server

Key UI Elements

UI Element	Function
Max Output Tokens	Sets the maximum number of tokens (word pieces) for the AI response.
Temperature	Controls creativity and randomness (lower = precise, higher = imaginative).
Top P	Probability threshold for nucleus sampling.
Top K	Restricts output to the top K most likely next tokens.
Model	Allows selecting the AI model (default: <code>gemini-1.5-flash</code>).
Prompt Text Area	Field for the user's custom text input or query.
Send Prompt Button	Sends the entered prompt and parameters to the Gemini API.

Setup Instructions

1. Prerequisites

- Obtain a **Google Gemini API key** from [Google AI Studio](#).
- Ensure internet connectivity for API calls.
- Install a simple web server (optional for local hosting).

2. Configuration

Insert your API key in the script:

```
const API_KEY = "YOUR_GEMINI_API_KEY";
```

Set default values:

```
const config = {  
  model: "gemini-1.5-flash",  
  max_output_tokens: 2048,  
  temperature: 0.6,
```

```
    top_p: 0.4,  
    top_k: 60  
};
```

API Request Example

```
const prompt = document.querySelector("#prompt").value;  
  
const payload = {  
  contents: [  
    {  
      role: "user",  
      parts: [{ text: prompt }]  
    }  
  ],  
  generationConfig: {  
    temperature: 0.6,  
    topP: 0.4,  
    topK: 60,  
    maxOutputTokens: 2048  
  }  
};  
  
fetch(`https://generativelanguage.googleapis.com/v1beta/models/gemini-  
1.5-flash:generateContent?key=${API_KEY}`, {  
  method: "POST",  
  headers: { "Content-Type": "application/json" },  
  body: JSON.stringify(payload)  
})  
  .then(response => response.json())  
  .then(data => {  
    const result = data.candidates[0].content.parts[0].text;  
    document.getElementById("output").innerText = result;  
  });
```

Output

- The Gemini API returns a **textual completion** based on the user's prompt.
- The generated text is displayed in the browser.
- No data is stored or logged.