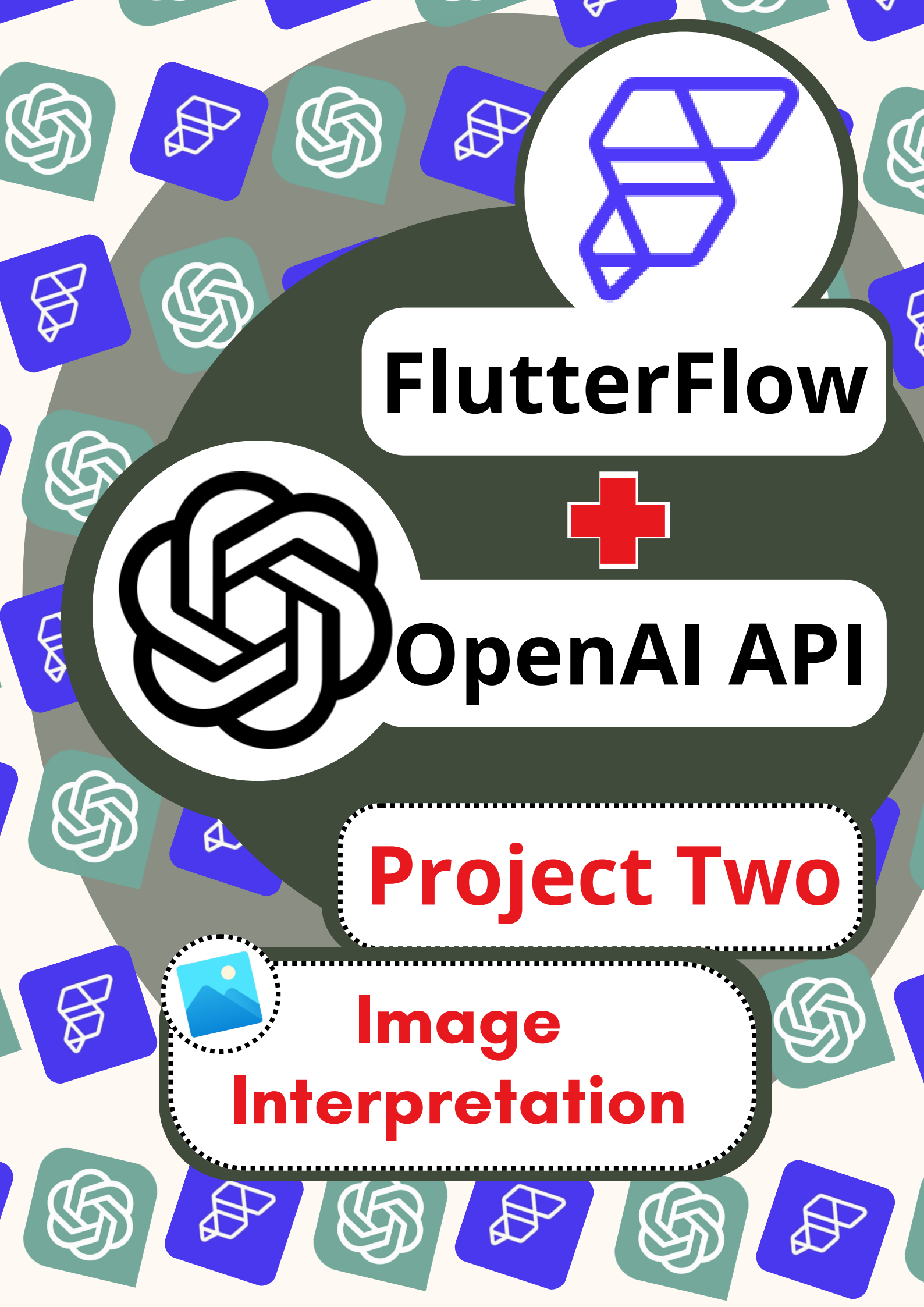

FlutterFlow
OpenAI API
Project Two
Image Interpretation

The Image Interpretation project is an intelligent application built with FlutterFlow and powered by the OpenAI Image & Vision API

Its purpose is to act as a visual interpreter for users by analyzing provided image URLs to generate detailed descriptions, explanations, and contextual insights regarding* the visual content.

*Following the OpenAI Image Analizer norms

1. Context and Overview

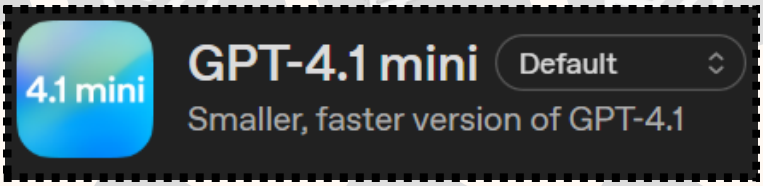
The Image Interpretation project is an AI-powered visual assistant that brings computer “vision”. It integrates FlutterFlow (low-code UI) with the OpenAI Image API (using gpt-5mini). This fusion lets users upload any image URL, ask natural questions about it, and receive accurate, detailed answers instantly demonstrating seamless low-code and multimodal AI integration.

2. Understanding the OpenAI Image API

OpenAI's Image API allows:

This categorie of models:

- Image interpretation
- Object detection
- Scene description
- Reading embedded text (OCR-like)
- Safety-aware analysis



Specifically for this project

```
"input": [
  { "role": "user",
    "content": [
        { "type": "input_text", "text":
"PROMPT" },
        { "type": "input_image",
"image_url": "URL_IMAGE" }
    ]
  }
]
```

The default of image inputs utilize the gtp.4.1.mini

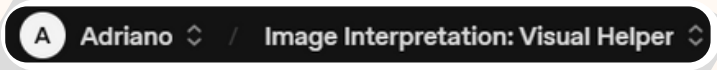
This flexible format allows combining a user prompt with any online-hosted image.

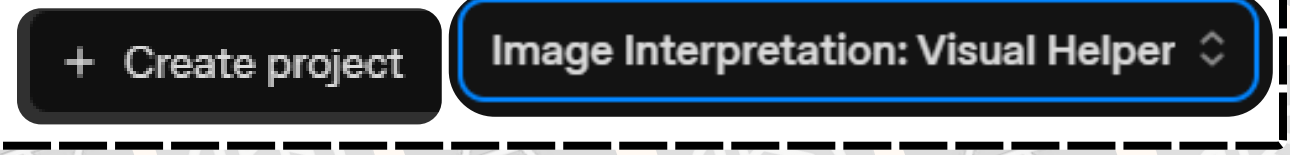
3. Creating the Project and API Keys

In the OpenAI dashboard

Create the project named

Image Interpretation: Visual Helper



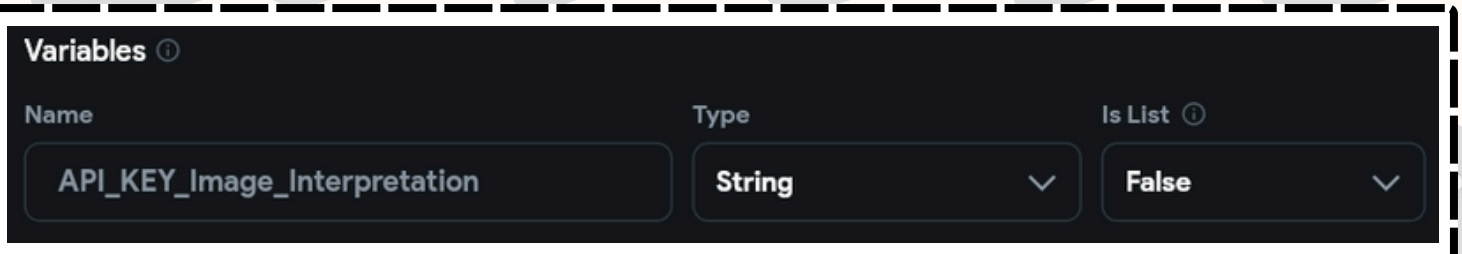


Generate an API Key

Don't Share your API Key

NAME	STATUS	SECRET KEY	CREATED	LAST USED	CREATED BY
API_Key_Image_Interpretation	Active	sk-...1o0A	1 de dez. de 2025	1 de dez. de 2025	Lazy Cap

Store the key securely (FlutterFlow → variables)



4. Screen Design in FlutterFlow

Container 1

Title: **Image Interpretation: Visual Helper**

Text: Image Interpretation: Visual Helper

URL_IMG

Image URL

Text: Image URL

Button

Button: ASK AI

Image URL

URL_Text_Field: Image Adress

Question_ Text_Field

Write a Question Related to the image

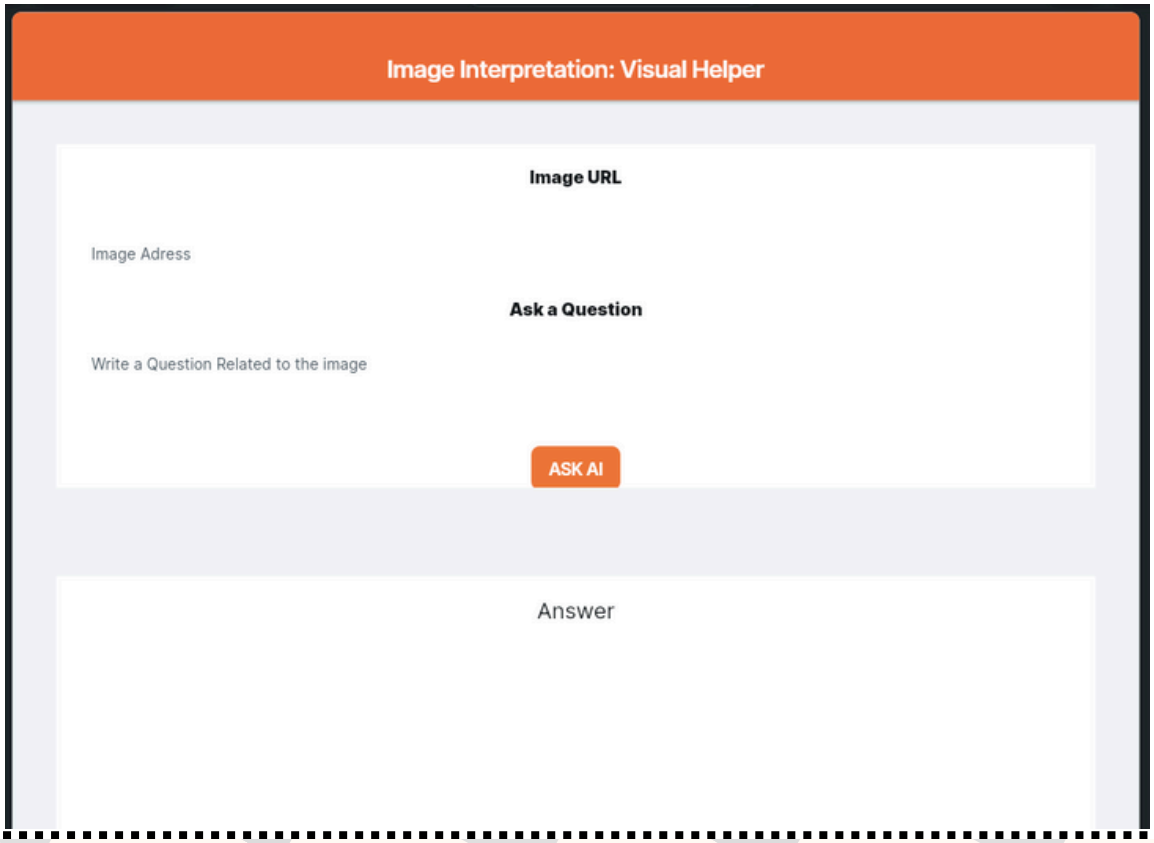
Question_Text_Field: Write a Question Related to the image

Container 2

Answer

Answer_Text: Answer

Preview



5. Creating the API Call Inside FlutterFlow

API Calls + Add

New API Call

App Image Interpretation Synced
main Production

Name: "Image Interpretation"

Set as POST

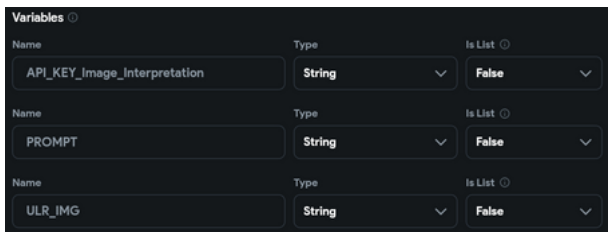
Image Interpretation
api.openai.com/v1/responses
POST

Add headers:

- Content-Type: application/json
- Authorization: Bearer [API_KEY_Image_Interpretation]

Headers
Content-Type: application/json
Authorization: Bearer [API_KEY_Image_Interpretation]

Create variables



- API_KEY_Image_Interpretation → string
- PROMPT → string (input from user)
- URL_IMG → string (input from user)

Define Response Varable:

Response & Test

$.output[:]content[:].text

Answer = JSON Path

JSON Path	Name	Response Preview	Type
$.output[:].content[:].text	Answer	*[Click "Test API Call" to view preview]*	String

In JSON Body, add:

the OpenAI API **doc** Docs

Call Definition


```
"input": [
{ "role": "user",
"content": [
{ "type": "input_text", "text": "PROMPT" },
{ "type": "input_image", "image_url": "URL_IMAGE" }
]
}
]
```

Insert

6. Applying the Image API Inside the App

In the "Send Question" button:

Add Action → **API Call**

Pass: API_KEY_Image_Interpretation PROMPT

Button ASK AI

On Tap 3 actions

On Success → Show Snackbar **"Request Successful"**

Action 2 Show Snack Bar Value Successful Request

On Error → Show Snackbar **"Error Detected"**

Action 3 Show Snack Bar Value Error

On the answer container:

- Set the text to show the API variable **Answer**

Text Answer

Set from Variable
Type: String

Variable
apiResultpws
Action Output Predefined Path

API Response Options
JSON Body

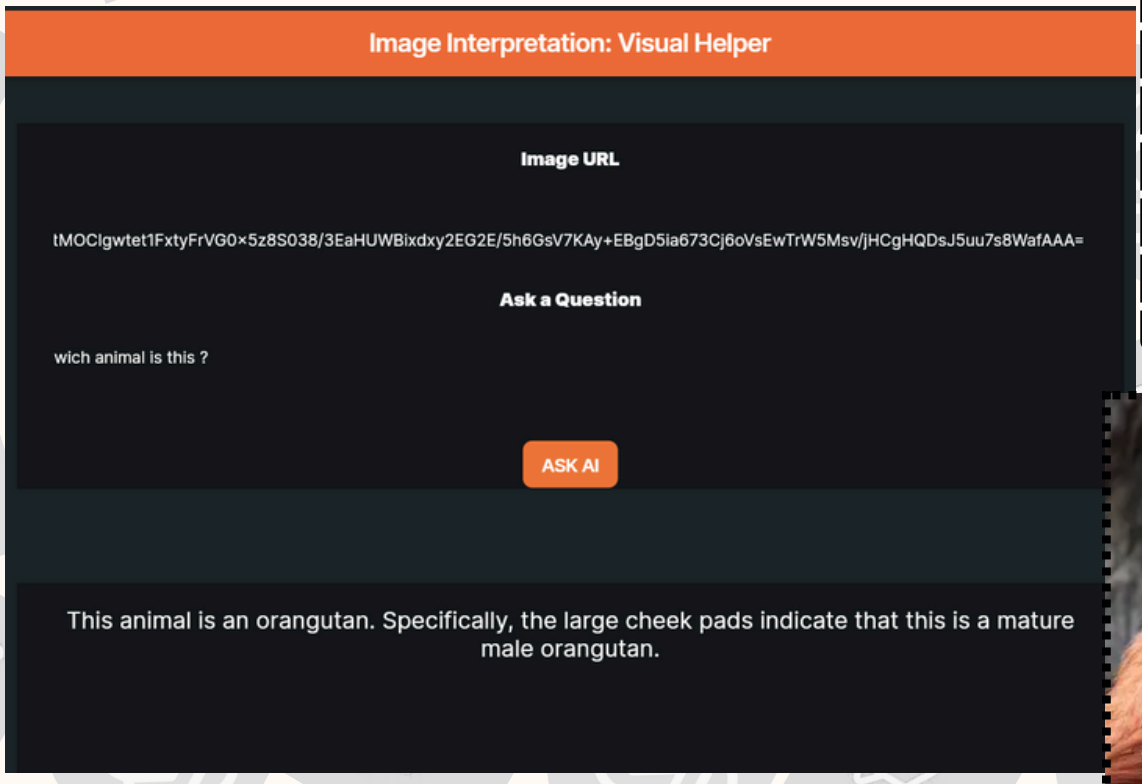
Available Options
Predefined Path

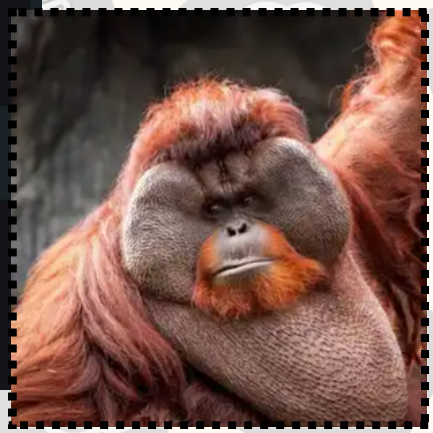
Predefined Path Name
Answer

Default Variable Value
Answer

Testing

Ask a Question





7. Conclusion

The Image Interpretation: Visual Helper project demonstrates how to

- Integrate Flutterflow with the OpenAI Image & Vision API
- Send dynamic prompts and image URLs to perform detailed image analysis
- Capture structured responses with JSON paths
- Build a clean, user-friendly interface for visual understanding tasks
- Enable real-time scene description, object identification, and contextual image insights

It is a practical, fully functional example of how low-code development + multimodal AI instantly unlocks advanced computer vision features (object detection, scene description, OCR, and contextual reasoning) without writing a single line of backend or ML code, making sophisticated image-understanding apps accessible to anyone in minutes

Next Project

Image Regerator, project demonstrates in practice how to:

- Integrate Flutterflow with the OpenAI Image Generation API
- Allow users to generate images by simply describing what they want
- Bind dynamic variables to API requests
- Handle responses and display generated images
- Build fast, smart, low-code creative applications

Image Regerator

Quick Short Project

FlutterFlow

+

OpenAI API