

HANDBOOK

AI-Powered Text Generation and its Integration



Table of Contents

Table of Contents

.....	0
Table of Contents	1
Course Objectives.....	2
Integration of Gemini	3
Introduction to Gemini	3
Creating API Key.....	4
Installing and Importing Gemini Library	6
Adding API Key	7
Adding a Query	8
Generating Text Response	8
After sending the query, Gemini processes it and sends back a response.....	8
Key Points to Remember	9
References	10

Course Objectives

After completing this handbook, learner will be able to

- Apply modern web development skills using AI-assisted tools like GitHub Copilot and Cursor AI.
- Enable rapid prototyping and data visualization through Streamlit for building interactive web applications.
- Introduce AI integration techniques using tools like Gemini API for smart feature development (e.g., chatbots, content generation).
- Develop proficiency in version control and cloud deployment using Git, GitHub, and Streamlit Community Cloud.
- Empower learner to transfer industry-relevant skills to students, enhancing employability and innovation in academic projects.

Chapter 3: AI-Powered Text Generation and its Integration

Learning Outcomes:

- Explain the purpose and key capabilities of the Gemini generative AI model and its API.
- Create and manage API keys required for secure access to the Gemini API.
- Install the Gemini Python client library and import it into a programming environment.
- Configure an application by adding the API key securely for authenticated communication with Gemini services.
- Formulate and send well-structured text queries (prompts) to the Gemini model using the client library.
- Generate, retrieve, and interpret text responses returned by the Gemini API for integration into applications.

Integration of Gemini

The Gemini API is a gateway to use powerful generative AI models that can create text responses based on prompts. By integrating it into a project, a teacher or developer can build applications that automatically generate content such as explanations, answers, lesson outlines, or summaries. This topic explains the process of using the Gemini API in detail.

Introduction to Gemini

Gemini is a generative AI model that is designed to understand text prompts and produce meaningful responses. Through the Gemini API, you can connect your application with this model. The API works like a bridge between your software and the model, ensuring smooth communication. When you interact with Gemini, you send a request (called a query or prompt) to the model through the API, and you receive a response that is automatically generated. This process allows your program to behave in an intelligent way without you writing every piece of content manually.

Analogy: Imagine you are a faculty member conducting a seminar. Instead of answering all the questions yourself, you invite a guest expert. You pose a question to this expert, and they give you an answer based on their knowledge. Gemini works

in a similar way. It is the guest expert that your program asks questions to, and the Gemini API makes sure the question reaches the expert and the answer comes back.

Creating API Key

To use Gemini API, you first need an **API key**. This is a unique code that verifies your identity and authorizes you to make requests.

Steps to create an API key:

1. **Create an account on the Gemini platform:** Sign up on the platform that provides access to Gemini models - <https://gemini.google.com/app>.

Gemini
2.5 Flash ▾

About Gemini

Sign in

Meet Gemini, your personal AI assistant

Source: <https://gemini.google.com/app>

2. Now navigate to - <https://ai.google.dev/gemini-api/docs>
3. **Locate the API section:** On the homepage, there is usually a section named "Get a Gemini API Key."

Home > Gemini API > Models

Gemini Developer API

Get a Gemini API Key

Get a Gemini API key and make your first API request in minutes.

```
Python  JavaScript  Go  Java  REST

from google import genai

client = genai.Client()

response = client.models.generate_content(
    model="gemini-2.5-flash",
    contents="Explain how AI works in a few words",
)

print(response.text)
```

Source: Screenshot

Generate a new key: Click the option to generate a new key. The system will create a long string of letters and numbers. This is your API key.

It's time to build

Experience the multimodal model from Google DeepMind

Use Google AI Studio

 Try Gemini

Build with the Gemini API

 Get API key

View Gemini API [documentation](#)

Source: Screenshot



Google AI Studio is the fastest way to start exploring and building with Gemini, our next generation family of multimodal generative AI models. Test prompts, get an API key, and go build.

The [Google APIs Terms of Service](#), [Gemini API Additional Terms of Service](#), and the [Google Privacy Policy](#) apply. Prompts and responses may be reviewed and used to train Google AI, so don't submit sensitive or personal information. Learn more about [data use](#). Gemini can make mistakes, so double-check it.

☒ I consent to the [Google APIs Terms of Service](#) and the [Gemini API Additional Terms of Service](#) and acknowledge that I have read the [Google Privacy Policy](#) *

☐ I'd like to receive emails for model updates, offers, useful tips, invitations to participate in research studies, and news about Google AI

I accept

Source: Screenshot

API Keys

[+ Create API key](#)

Quickly test the Gemini API

[API quickstart guide](#)

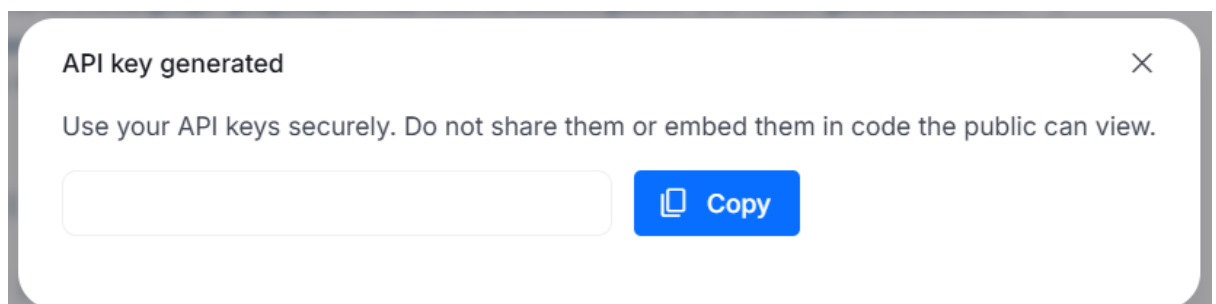
```
curl "https://generativelanguage.googleapis.com/v1beta/models/gemini-2.0-flash:generateContent" \
-H 'Content-Type: application/json' \
-H 'X-goog-api-key: GEMINI_API_KEY' \
-X POST \
-d '{
  "contents": [
    {
      "parts": [
        {
          "text": "Explain how AI works in a few words"
        }
      ]
    }
  ]
}'
```

Use code [with caution](#).

Your API keys are listed below. You can also view and manage your project and API keys in Google Cloud.

Source: Screenshot

4. **Copy and store it securely:** Keep this key safe. You should not share it publicly because anyone with your key can use your allocated usage.



Source: Screenshot

Analogy:

This is like receiving a digital ID card that allows you to enter a restricted research lab. Only those who have an ID can access the lab. The API key works in the same way for software access.

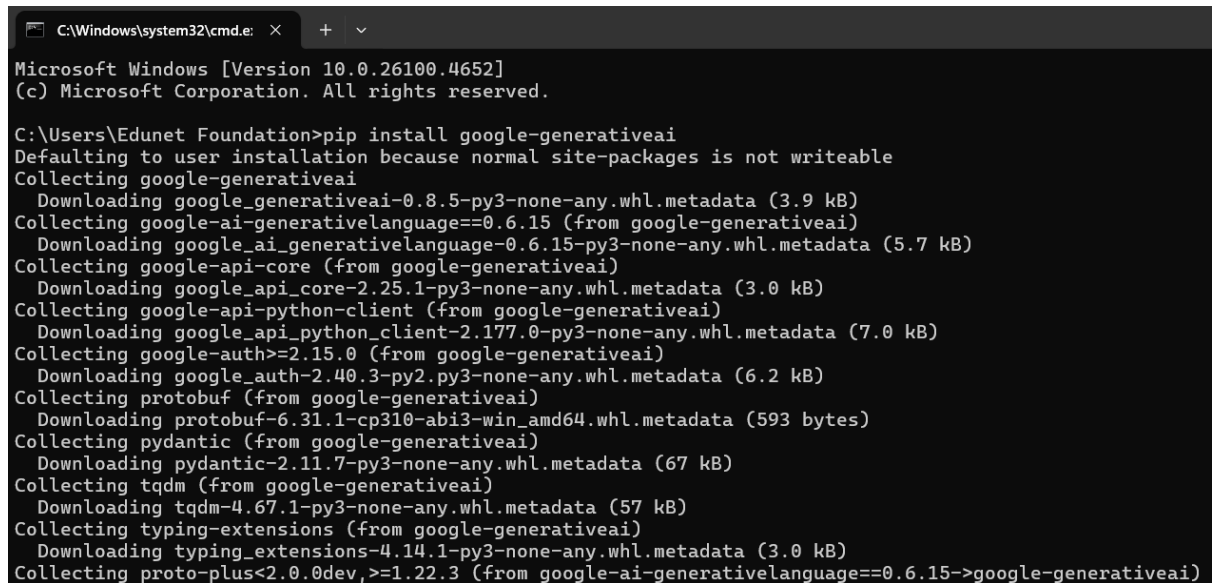
Installing and Importing Gemini Library

To make it easier to use the API, a programming library is provided. Instead of manually writing all the network request code, you can use this library.

Steps to install and import the Gemini library (for example, in Python):

1. **Open your coding environment:** This could be an IDE, Command Prompt or a simple terminal.

2. **Use the package manager to install:** Type a command such as: `pip install google-generativeai`

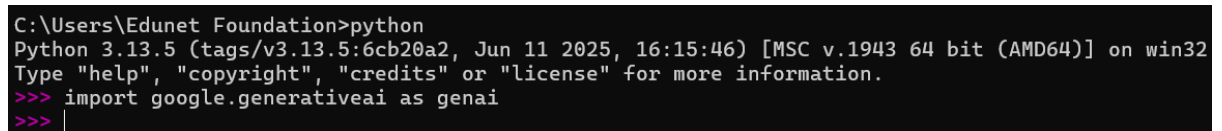


```
C:\Windows\system32\cmd.exe: X + v
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Edunet Foundation>pip install google-generativeai
Defaulting to user installation because normal site-packages is not writeable
Collecting google-generativeai
  Downloading google_generativeai-0.8.5-py3-none-any.whl.metadata (3.9 kB)
Collecting google-ai-generativelanguage==0.6.15 (from google-generativeai)
  Downloading google_ai_generativelanguage-0.6.15-py3-none-any.whl.metadata (5.7 kB)
Collecting google-api-core (from google-generativeai)
  Downloading google_api_core-2.25.1-py3-none-any.whl.metadata (3.0 kB)
Collecting google-api-python-client (from google-generativeai)
  Downloading google_api_python_client-2.177.0-py3-none-any.whl.metadata (7.0 kB)
Collecting google-auth>=2.15.0 (from google-generativeai)
  Downloading google_auth-2.40.3-py2.py3-none-any.whl.metadata (6.2 kB)
Collecting protobuf (from google-generativeai)
  Downloading protobuf-6.31.1-cp310-abi3-win_amd64.whl.metadata (593 bytes)
Collecting pydantic (from google-generativeai)
  Downloading pydantic-2.11.7-py3-none-any.whl.metadata (67 kB)
Collecting tqdm (from google-generativeai)
  Downloading tqdm-4.67.1-py3-none-any.whl.metadata (57 kB)
Collecting typing_extensions (from google-generativeai)
  Downloading typing_extensions-4.14.1-py3-none-any.whl.metadata (3.0 kB)
Collecting proto-plus<2.0.0dev,>=1.22.3 (from google-ai-generativelanguage==0.6.15->google-generativeai)
```

Source: Screenshot

3. **Verify installation:** After installation, check that no error messages are shown.
4. **Import the library in your code:** In your Python file, you write: `import google.generativeai as genai`



```
C:\Users\Edunet Foundation>python
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import google.generativeai as genai
>>> |
```

Source: Screenshot

Analogy:

Installing the library is like bringing a set of ready-made teaching aids to the classroom. Instead of creating everything from scratch, you use these aids to simplify your work.

Adding API Key

The library needs your API key to authenticate. Without it, no request will be processed.

Steps to add your API key:

1. **Store the key securely:** The preferred way is to store it in an environment variable. This is like keeping your ID card in a wallet.
2. **Access the key in your code:** Alternatively, you can directly paste the key in your code (not recommended for production use).
3. **Configure the library to use the key:** Example in Python: `genai.configure(api_key="YOUR_API_KEY")`

Adding a Query

Once authentication is complete, you can start interacting with Gemini by sending a prompt.

Steps to add a query:

1. Think about the task clearly: Decide what you want from Gemini: a summary, explanation, creative story, or structured output.
2. Frame the query: Write the prompt in a clear and specific way.
3. Send the prompt to the model: Use the library's function to send this prompt.

Code:

```
import google.generativeai as genai

# Step 1: Configure API key
genai.configure(api_key="YOUR_API_KEY")

# Step 2: Choose the model
model = genai.GenerativeModel("models/gemini-1.5-flash-latest")

# Step 3: Provide a prompt
prompt = "Explain the process of machine learning in simple terms suitable for B.Tech students."

# Step 4: Generate the response
response = model.generate_content(prompt)

# Step 5: Display the output
print(response.text)
```

Generating Text Response

After sending the query, Gemini processes it and sends back a response.

Steps to handle the response:

1. Access the response content: The API returns the text output in a response object.
2. Display or use the response: You can print the result on screen, use it in a web application, or save it in a file.
3. Post-process if needed: If the text needs formatting or further use, your program can process it.

```
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Edunet Foundation>python
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import google.generativeai as genai
>>> genai.configure(api_key="AIzaSyDIGjwXy7m00W4eYZWpp_wqIcWraMrV85k")
>>> model = genai.GenerativeModel("models/gemini-1.5-flash-latest")
>>> prompt = "Explain the process of machine learning in simple terms for B.Tech students."
>>> response = model.generate_content(prompt)
>>> print(response.text)
Imagine you're teaching a dog a new trick, like fetching a ball. You don't explicitly program the dog's brain with instructions; instead, you show it examples, reward good behavior (fetching the ball), and correct mistakes. Machine learning is similar.

Instead of a dog, we have a computer program. Instead of a ball, we have data. And instead of rewards and corrections, we have algorithms.

The process generally follows these steps:

1. Data Collection: Gather a large amount of relevant data. This is like showing the dog many examples of fetching a ball. For example, to build a spam filter, you'd need a huge dataset of emails labeled as "spam" or "not spam."

2. Data Preparation: Clean and organize the data. This is like making sure the ball is always the same kind and readily available for the dog to fetch. This step includes handling missing values, converting data into a usable format, and potentially splitting the data into training, validation, and testing sets.

3. Model Selection: Choose an appropriate algorithm. This is like deciding what training method to use for the dog (positive reinforcement, clicker training, etc.). Different algorithms are better suited for different types of problems
```

Source: Screenshot

Key Points to Remember

1. The entire process has to be done in sequence: get the API key, install the library, add the key in your code, create a prompt, and get a response.
2. The clarity of your prompt determines the usefulness of the response.
3. Always secure your API key to prevent misuse.
4. These responses can be integrated into tools like lecture note generators, chatbots for students, and digital teaching assistants.

References

- About Gemini. (n.d.). Learn about Gemini, the everyday AI assistant from Google. Gemini. <https://gemini.google/about/>
- Caballar, R., & Stryker, C. (2025, June 3). Google Gemini. What is Google Gemini? <https://www.ibm.com/think/topics/google-gemini>
- Pichai, S. (2024, September 27). Introducing Gemini: our largest and most capable AI model. Google. <https://blog.google/technology/ai/google-gemini-ai/>