Here's a **step-by-step guide with code** for an **easy, pre-trained chatbot using Dialogflow + Flask**. 🚀

## ****🔹 Step 1: Setup Dialogflow Agent****

1. **Go to** [Dialogflow Console](https://dialogflow.cloud.google.com/)**.**
2. **Create a new agent** (e.g., "CollegeEnquiryBot").
3. **Create Intents** for FAQs:
   1. Example Intent: admission\_info
      1. Training Phrases:
         1. "Tell me about the admission process."
         2. "How to apply for admission?"
      2. Responses:
         1. "Admissions open from June 1st. Visit our website for details."
4. **Enable Webhook** under the **Fulfillment** tab.

## ****🔹 Step 2: Install Required Packages****

📌 Install Flask and Dialogflow SDK:

pip install flask dialogflow google-auth google-auth-oauthlib google-auth-httplib2

## ****🔹 Step 3: Create a Flask App****

📌 **Flask App to Connect with Dialogflow Webhook**

Create a file app.py:

from flask import Flask, request, jsonify

import dialogflow\_v2 as dialogflow

import os

from google.oauth2 import service\_account

app = Flask(\_\_name\_\_)

# Load Dialogflow Credentials

GOOGLE\_AUTHENTICATION\_FILE = "your-service-account-file.json"

credentials = service\_account.Credentials.from\_service\_account\_file(GOOGLE\_AUTHENTICATION\_FILE)

dialogflow\_session\_client = dialogflow.SessionsClient(credentials=credentials)

PROJECT\_ID = "your-dialogflow-project-id"

def detect\_intent(session\_id, text):

session = dialogflow\_session\_client.session\_path(PROJECT\_ID, session\_id)

text\_input = dialogflow.types.TextInput(text=text, language\_code="en")

query\_input = dialogflow.types.QueryInput(text=text\_input)

response = dialogflow\_session\_client.detect\_intent(session=session, query\_input=query\_input)

return response.query\_result.fulfillment\_text

@app.route('/')

def home():

return "College Enquiry Chatbot is Running!"

@app.route('/chat', methods=['POST'])

def chat():

data = request.get\_json()

user\_message = data.get("message", "")

if not user\_message:

return jsonify({"response": "Please enter a valid question."})

response\_text = detect\_intent("123456", user\_message)

return jsonify({"response": response\_text})

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

### ****🔹 Step 4: Get Dialogflow Credentials****

1. **Go to** [Google Cloud Console](https://console.cloud.google.com/)**.**
2. Enable **Dialogflow API**.
3. Create a **Service Account Key (JSON format)** and download it.
4. Rename the JSON file to "your-service-account-file.json" and place it in your project directory.
5. Replace "your-dialogflow-project-id" in the PROJECT\_ID variable.

## ****🔹 Step 5: Create a Simple Frontend UI****

📌 Create templates/index.html for a basic chatbot UI:

<!DOCTYPE html>

<html lang="en">

<head>

<title>College Chatbot</title>

</head>

<body>

<h2>Ask me anything about the college!</h2>

<input id="userInput" type="text" placeholder="Type your question...">

<button onclick="sendMessage()">Send</button>

<div id="chatOutput"></div>

<script>

function sendMessage() {

let userMessage = document.getElementById("userInput").value;

fetch('/chat', {

method: 'POST',

headers: {'Content-Type': 'application/json'},

body: JSON.stringify({'message': userMessage})

})

.then(response => response.json())

.then(data => document.getElementById("chatOutput").innerHTML += `<p>${data.response}</p>`);

}

</script>

</body>

</html>

## ****🔹 Step 6: Run the Chatbot****

📌 Start the Flask server:

python app.py

📌 Open your browser and visit:

http://127.0.0.1:5000/

Try asking:

Tell me about the admission process.

## ****🔹 Step 7: Deploy on Cloud****

1. **Use ngrok** to expose Flask server to the internet:

pip install ngrok

ngrok http 5000

1. **Deploy on Heroku** or **AWS (EC2, Lambda)** for a public chatbot.

### ****Why Do We Need a Service Account Key (JSON)?****

The **Service Account Key (JSON file)** is required to **authenticate** your Python application (Flask) with **Google Dialogflow**. It allows your chatbot to communicate with **Dialogflow's API** securely.

Now, let's go through the step-by-step process to create and download the **Service Account Key (JSON format).**

## ****🔹 Step 1: Open Google Cloud Console****

1. Go to [Google Cloud Console](https://console.cloud.google.com/).
2. **Sign in** with your Google account.

## ****🔹 Step 2: Select or Create a Google Cloud Project****

1. Click on the **"Select a Project"** button at the top.
2. If you already have a **Dialogflow project**, select it.
3. If not, click **"New Project"** and:
   1. Give it a name (e.g., college-chatbot).
   2. Click **"Create"** and wait for it to be ready.

## ****🔹 Step 3: Enable Dialogflow API****

1. In the **Google Cloud Console**, go to the **Navigation Menu** (☰) on the top-left.
2. Click **"APIs & Services" → "Library"**.
3. Search for **Dialogflow API**.
4. Click **"Enable"**.

## ****🔹 Step 4: Create a Service Account****

1. In the **Google Cloud Console**, go to **Navigation Menu (☰) → IAM & Admin → Service Accounts**.
2. Click **"Create Service Account"**.
3. **Enter a name** (e.g., dialogflow-access).
4. Click **"Create"**.

## ****🔹 Step 5: Assign a Role****

1. Under **Grant this service account access to the project**, click **"Select a role"**.
2. Choose:
   1. **"Dialogflow API Admin"** (Full access)
   2. OR **"Dialogflow API Client"** (Limited access, but enough for chatbot)
3. Click **"Continue"**.

## ****🔹 Step 6: Create and Download JSON Key****

1. Scroll down to the **"Create Key"** section.
2. Click **"Add Key" → "Create New Key"**.
3. Choose **JSON format**.
4. Click **"Create"**.
5. The JSON file will be downloaded automatically.
6. **Rename it to**: your-service-account-file.json
7. Move it to your **Python project directory**.

## ****🔹 Step 7: Set Up Your Python Script to Use the JSON Key****

Modify your **Flask app (**app.py**)** to load the key:

from google.oauth2 import service\_account

GOOGLE\_AUTHENTICATION\_FILE = "your-service-account-file.json"

credentials = service\_account.Credentials.from\_service\_account\_file(GOOGLE\_AUTHENTICATION\_FILE)

## ****✅ Final Check****

* Your project folder should now contain the JSON key:

/your-project-folder/

|-- app.py

|-- your-service-account-file.json ✅

* Your Flask app can now authenticate with **Dialogflow API**.

## ****🎯 Next Steps****

Now, try running your **Flask chatbot** again:

python app.py

Your chatbot should now be able to **send queries to Dialogflow and get responses**.

Your error **"PERMISSION\_DENIED"** confirms that your service account (dialogflow-access@college-chatbot-1.iam.gserviceaccount.com) **does not have the necessary permissions** on the project **collegeenquirybot-tgmk**.

## ✅ ****Fixing the Issue: Grant IAM Permissions Manually****

### ****Step 1: Log in with Your Personal Account****

Your active account is currently:

dialogflow-access@college-chatbot-1.iam.gserviceaccount.com

🔹 **Switch back to your personal account (**shrutiharayan06@gmail.com**)**:

gcloud auth login

Then set your account as active:

gcloud config set account shrutiharayan06@gmail.com

### ****Step 2: Grant Permissions to Your Service Account****

Now that you're using your owner account, **give the service account the necessary permissions**.

#### ✅ ****Add Dialogflow Roles****

Run the following command:

gcloud projects add-iam-policy-binding collegeenquirybot-tgmk \

--member="serviceAccount:dialogflow-access@college-chatbot-1.iam.gserviceaccount.com" \

--role="roles/dialogflow.admin"

#### ✅ ****Add Service Usage Permissions****

gcloud projects add-iam-policy-binding collegeenquirybot-tgmk \

--member="serviceAccount:dialogflow-access@college-chatbot-1.iam.gserviceaccount.com" \

--role="roles/serviceusage.serviceUsageConsumer"

#### ✅ ****Add Editor Role (If Needed)****

gcloud projects add-iam-policy-binding collegeenquirybot-tgmk \

--member="serviceAccount:dialogflow-access@college-chatbot-1.iam.gserviceaccount.com" \

--role="roles/editor"

### ****Step 3: Enable the API Again****

After granting permissions, **try enabling the Dialogflow API again**:

gcloud services enable dialogflow.googleapis.com --project collegeenquirybot-tgmk

### ****Step 4: Switch Back to Your Service Account****

Once permissions are fixed, switch back to your service account:

gcloud auth activate-service-account dialogflow-access@college-chatbot-1.iam.gserviceaccount.com --key-file="path-to-your-json-key.json"

Then verify: gcloud auth list

### ****Step 5: Test Again****

Now, test if you can list enabled services:

gcloud services list --enabled --project collegeenquirybot-tgmk

To add **context and follow-up intents** in **Dialogflow**, follow these steps:

### ****1️⃣ Setting Up the Parent Intents****

* **Intent 1:** admission\_process
* **Follow-up Question:** "Would you like the application link?"
  + **Set Output Context:**
    - **Name:** admission\_process\_followup
    - **Lifespan:** 2
* **Intent 2:** admission\_deadline

**Follow-up Question:** "Would you like assistance with the process?"

**Set Output Context:**

* + - **Name:** admission\_deadline\_followup
    - **Lifespan:** 2

### ****2️⃣ Creating Follow-Up Intents****

* **Follow-up for** admission\_process
  + Create a new intent: admission\_process - yes
  + Training phrases:
    - "Yes"
    - "Sure, send me the link"
  + **Set Input Context:** admission\_process\_followup
  + Response: "Here is the application link: [Insert Link]"
* **Follow-up for** admission\_deadline
  + Create a new intent: admission\_deadline - yes
  + Training phrases:
    - "Yes"
    - "Sure, I need help"
  + **Set Input Context:** admission\_deadline\_followup
  + Response:"Great! You can visit our admissions office or check our website for detailed assistance."

Follow up: Admission\_deadline - no

Intent ": "admission\_deadline - no"

trainingPhrases": [No,Not needed ,"I’ll do it myself"

response": [Alright, feel free to reach out if you change your mind!"

"Alright, let me know if you need it later!"

inputContexts": [admission\_deadline\_followup

download ngrok for secure https connection->unzip file ->place ngrok.exe file in chatbot folder

sign-in on ngrok using google/GitHub

open cmd from chatbot folder add your Authtoken: This is your personal Authtoken. Use this to authenticate the ngrok agent that you downloaded.

>ngrok config add-authtoken 2u4dP8O(personal token accessible after login ngrok under your authtoken from left hand side)

>ngrok http 5000(localhost port number accessible after running flask app)

download postman app locally to test api

postman app->collections->new->http->select post method -> Enter http://127.0.0.1:5000/chatbot in the URL field.

Go to Body → Select raw → Choose JSON format.

{

"session\_id": "12345",

"text": "Hello"

}

**My project ID: [collegeenquirybot-tgmk](https://console.cloud.google.com/home/dashboard?project=collegeenquirybot-tgmk" \t "https://dialogflow.cloud.google.com/" \l "/editAgent/collegeenquirybot-tgmk/_blank)**

In cmd for testing: curl -X POST http://127.0.0.1:5000/chatbot -H "Content-Type: application/json" -d "{\"message\": \"What courses are available?\"}"

Webhook Url <https://7d2c-103-182-130-44.ngrok-free.app/>

provide a **ZIP file** containing the intents in the correct format for direct import into Dialogflow.

Say this in 5 different ways “how is the admission process”

Give me synonyms of entity “admission”