Project name: College Enquiry Chatbot

Project ID: collegeenquirybot-tgmk

Project Overview

The **College Enquiry Chatbot** is a web-based conversational agent designed to assist users with common questions related to college admission, fees, courses, hostel facilities, and contact details and many more. It integrates **Google Dialogflow** for natural language processing(NLP) and **Flask** (Python's web framework) for backend handling.

It offers:
- 24/7 automated responses
- Reduced manual workload for college staff
- Fast, intelligent, and consistent query resolution

Objective

To simplify and automate student support by:
- Providing instant responses to FAQs
- Enabling easy access to college-related information
- Reducing human intervention and manual repetitive tasks

Technologies Used

Technology	Purpose
Dialogflow	NLP engine for intents, entity extraction, responses
Flask (Python)	Backend framework to route and serve data .
Google Cloud Service Account	Securely access Dialogflow API
HTML/CSS/JavaScript	Frontend design and dynamic message interaction
.env	Store API credentials and environment configurations
Railway	For hosting publically

Functional Workflow:
Step 1: Dialogflow Setup

- **Go to** Dialogflow Console**.**
- **Create a new agent** (e.g., "CollegeEnquiryBot").
- **Create Intents** for FAQs:

Example Intent: admission_info

Training Phrases: Tell me about the admission process, How to apply for admission?

Responses: Admissions open from June 1st. Visit our website for details.

- **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
```

Step 3: Create a Flask App to Connect with Dialogflow Webhook

Create a file app.py

Step 4: Get Dialogflow Credentials

1. **Go to** Google Cloud Console**.**
2. Enable **Dialogflow API**.
3. Create a **Service Account Key (JSON format)** and download it.
4. Rename the JSON file to "my-chatbot-key.json" and place it in your project directory.
5. Add collegeenquirybot-tgmk in the PROJECT_ID variable.

Step 5: Create a Simple Frontend UI

- Create templates/index.html for a basic chatbot UI
- Create static/style.css and static/script.js and images in separate static folder.

Step 6: Run the Chatbot

- Start the Flask server: python app.py
- Open your browser and visit: http://127.0.0.1:5000/

Project structure:

```
Folder name: college_enquiry_chatbot
          |-- app.py            (flask backend)
          |-- my-chatbot-key.json       (service-account-file)
          |-- static  (folder)
                    |-- script.js
                    |--style.css
                    |-- logo.jpg (all images)
          |-- templates
                    |-- index.html   (all html files)
```

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.
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. **Rename it to**: my-chatbot-key.json
6. Move it to your **Python project directory**.

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

Deploy Your Flask Chatbot on Railway (Free Hosting)

GitHub Account: Push your chatbot project to GitHub

Step-by-Step Setup

1. Go to Railway

Visit https://railway.app

Sign up with your **GitHub account**

2. Create a New Project

Click "New Project"

Choose **"Deploy from GitHub repo"**

Select your chatbot repo

Click **"Deploy"**

3 Set up the environment

Railway needs to know about your credentials (like the Dialogflow JSON file):

Instead of uploading the file, we'll copy-paste its content as an environment variable.

Open your my-chatbot-key.json file

Copy all the JSON content

In Railway, go to:

Project → Variables tab → click **"New Variable"**

Name: GOOGLE_APPLICATION_CREDENTIALS_JSON

Value: *Paste the full JSON content.*

4 Modify Your Code to Use This Variable

In your Python code, **replace this line**:

```
os.environ["GOOGLE_APPLICATION_CREDENTIALS"] = "my-chatbot-key.json"
```

With this:

```
import json
from google.oauth2 import service_account
```

```
creds_json =
json.loads(os.getenv("GOOGLE_APPLICATION_CREDENTIALS_JSON"))
credentials = service_account.Credentials.from_service_account_info(creds_json)
session_client = dialogflow.SessionsClient(credentials=credentials)
```

This loads the credentials directly from the Railway environment variable instead of a file.

5 Make sure you're listening on the correct port

Change: app.run(debug=True)

To:

```
import os
port = int(os.environ.get("PORT", 5000))
app.run(host="0.0.0.0", port=port)
```

Create a file named Procfile (no extension!)

Inside the file, add this line:

```
web: python app.py
```

Replace app.py with whatever your Flask app's main file is

Redeploy or Restart the Service

- Once you've made these changes and committed them to GitHub:
- Railway will redeploy automatically
- We'll get a **public URL** like https://yourapp.up.railway.app

DONE! Chatbot is Live