

DS2002 Data Project 2: Flask Chatbot Assistant

Total: 55 Points (+10 Bonus)

Due: May 16 at 11:59 PM

Deployment: Must be hosted on your own GCP instance and accessible from the public internet

You may work alone or in groups of up to 4 people

Goal

Build a Flask-based chatbot assistant that can answer user questions using two distinct datasets:

- One from a live API call
- One from a local data source, such as CSV, JSON, or SQL

Your assistant can focus on anything: weather, recipes, sports stats, pop culture, or custom topics.

The local data must come from an external source and be processed through a Python-based ETL pipeline.

Once built, your chatbot must be deployed and publicly accessible via your GCP instance.

Assignment Requirements

Chatbot Assistant

- Flask app with a /chat POST route
- Chatbot must respond to user questions based on a real dataset
- One data source must be from a live public API
- The other must be a locally stored data file (CSV, JSON, or SQL) generated by your ETL script

ETL Function

You must write a Python-based ETL (Extract, Transform, Load) script that:

- Extracts raw data from an external source
- Transforms the data: format conversion, column pruning/augmentation, cleansing
- Loads the final result into a new file or SQL database that you Bot uses, in addition to any api calls it may need.

This local file/database is then used by your chatbot to answer questions.

Error Handling

Your chatbot and ETL pipeline must fail gracefully with clear and informative error messages.

GCP Deployment

- Host your Flask chatbot on a Google Cloud Platform VM or App Engine
- It must be publicly accessible via the web
- Include the public URL in your GitHub README.md

BONUS (10 Points): Discord Bot Integration and/or Gemini LLM Integration

- Integrate Gemini LLM to enhance chatbot intelligence (can be used instead of or in addition to Discord bot)
- Add a Discord bot that uses the same assistant logic
- Reuse Flask logic or connect directly in Python using discord.py

Submission Checklist

- GitHub repo with all code
- Flask /chat route works
- ETL script that fetches, transforms, and loads external data
- Local dataset used in chatbot comes from your ETL pipeline
- Integration with public API
- GCP instance is publicly accessible
- 2-page reflection must be included. It should cover:
- Bonus: Discord bot (optional)

Grading Rubric

| Criteria | Points |
|--|--------|
| Flask chatbot works and is publicly hosted | 10 |
| API and local data are integrated | 10 |
| ETL pipeline processes external data correctly | 10 |

Chatbot uses both sources to answer questions 10

Creativity + polish 5

Documentation + GitHub clarity+Reflection Paper 10

- How you selected your datasets

- Challenges or problems faced during the process

- Key learnings and discoveries

- What enhancements or features you would have added with more time

BONUS: Discord Bot Integration/Gemini 10

-