Use Case: The Chatbot Blueprint: Imagine, Build, Solve

Introduction

“Tools don’t solve problems. People do.”

At NeoStats, we believe that real innovation happens when technology meets imagination. In this challenge, we’re giving you the opportunity to act as an AI engineer on a mission to build an intelligent chatbot that doesn’t just answer; it understands, retrieves, and what not!

You’ll start with a working chatbot template built in Streamlit designed to be flexible and extensible; with built-in support for integration with major LLM providers like OpenAI, Groq, and Google Gemini. But here’s what makes this challenge different: we’re not giving you a problem to solve. We’re giving you the **foundation of a solution** and it’s up to you to define the problem and build the solution accordingly.

Think creatively. Identify a domain, use case, or real-world challenge where intelligent, contextual conversations can make a difference. Then use the tools and features provided to **build a chatbot that solves it** and if you think it needs more than what’s provided, extend it further.

This is your opportunity to combine technical skill with design thinking, to not just build a chatbot, but to architect a smart, usable solution from the ground up.

Tasks to Complete

1. RAG Integration (Retrieval-Augmented Generation)

- Add the ability for the chatbot to reference local documents or knowledge bases.
- Use vector embeddings to retrieve relevant chunks from documents.
- Keep your embedding models inside models/embeddings.py, and invoke it inside app.py.
- Keep your embedding logic inside utils.

2. Live Web Search Integration

- Add a tool/functionality to perform **real-time web searches** when the LLM lacks knowledge.
- Integrate this logic in utils and manage API keys in the config/ folder.

3. Response Modes: Concise vs Detailed

- Implement in the UI to switch between:
    - **Concise**: Short, summarized replies
    - **Detailed**: Expanded, in-depth responses

Recommended Project Structure

Ensure you **follow the folder structure provided**. You may add new files, but adhere to modularity:

```
project/
├──── config/
|     └──── config.py       ← All API keys, settings
|
├──── models/
|     ├──── llm.py          ← LLM models (OpenAI/ Groq / Gemini)
|     └──── embeddings.py   ← RAG embedding models
|
├──── utils/                ← Functions
|
├──── app.py                ← Main Streamlit UI logic
|
├──── requirements.txt      ← Update with required packages
└──── <your additional files if needed>
```

Development Guidelines

- Put your project in a **GitHub repository**.
- **Do NOT commit API keys** directly — keep them inside config/config.py as environment variables or constants.
- **Reuse code** smartly by creating reusable functions in the appropriate modules (e.g., utils/).
- **Wrap all functional code in try/except blocks** to catch and log errors.
- **Debug any issues** in the existing chatbot code if you find bugs.
- **Avoid code generation tools** like Cursor or GitHub Copilot.
- Be **original and creative** — avoid copying others' work. Think of your own solution strategies.
- You are free to use other frameworks such as Flask.
- Ensure your GitHub repository, Streamlit chatbot, and presentation (PPT) are accessible or shared with appropriate access levels.

Deployment Instructions

1. Finalize your app in app.py
2. Deploy your app to **Streamlit Cloud**: https://streamlit.io/cloud
3. Make sure it works end-to-end with the added features
4. Include the **Streamlit Cloud link** in your final **PPT deck**

Final Deliverables

You should submit the following:

- A working project zip (without the virtual environment folder) with your completed app.
- Deployed **Streamlit Cloud app link**
- Short and impactful **PPT presentation deck** containing:
  - Use Case Objective
  - How you approached the problem
  - Solution
  - Features implemented
  - Any challenges faced
  - Deployment link

We're looking for creative minds who can not only solve technical challenges but also think strategically to design impactful solutions in a fast-paced, deadline-driven environment.

This is your opportunity to showcase your technical expertise, problem-solving skills, and design thinking. The foundation is provided, but **how you define the problem and build the solution (with mandatory + additional features) is where you'll shine**. We are excited to see how you combine technology and imagination to create something that truly matters.

We believe that great things happen when technology meets creativity — and we can't wait to see what you build.

All the best!