Comprehensive Project Documentation

Project Title: PantherBot - GSU Smart Program Assistant 🎓🤖

---

1. Introduction

PantherBot is an AI-powered chatbot designed to assist students in exploring undergraduate and graduate programs at Georgia State University (GSU). It leverages OpenAI's GPT-4 Turbo model and real-time data from GSU program listings to provide fast, intelligent, and user-friendly responses. The chatbot enhances the student's decision-making experience by offering a conversational interface that retrieves and personalizes program information efficiently.

This project was developed as part of the MSA 8700 course, aimed at designing and deploying real-world AI business solutions.

---

2. Business Problem Addressed

Students often find it challenging to navigate the vast amount of information available on university websites when searching for the right academic program. The information is typically fragmented, hard to compare, and not easily accessible in summarized formats.

PantherBot solves these problems by:

- Summarizing detailed program information.
- Providing quick answers through a conversational interface.
- Offering real-time search fallback when internal data is insufficient.
- Enhancing user experience with multi-turn memory, allowing context-rich conversations.

---

3. Solution Architecture

Frontend

- **Framework**: Streamlit
- **Features**: Responsive web-based interface, user-friendly chat experience, integrated program selection and feedback collection.

AI Backend

- **Model**: OpenAI GPT-4 Turbo
- **Capabilities**: Handles multi-turn conversation, contextual memory, and advanced prompt chaining.

Data Sources

- **Graduate Programs**: Excel dataset (.xlsx format)
- **Undergraduate Programs**: Excel dataset (.xlsx format)

External Search

- **API**: Serper.dev Google search API
- **Purpose**: To retrieve additional program-related information when internal datasets are insufficient.

Hosting

- **Platform**: Streamlit Community Cloud
- **Deployment**: Publicly accessible URL for student, faculty, and instructor access.

---

4. Key Features

- Intelligent and personalized search of GSU programs.
- Retrieval of key program details (tuition, faculty, career outcomes).
- Real-time web search fallback.
- Multi-turn conversation memory for improved user experience.
- Avatars and fade-in animation for enhanced visual appeal.
- Integrated live feedback collection system.
- Offline-capable architecture for future integration with Ollama APIs.

---

5. Setup Instructions

Prerequisites

- Python 3.8+
- Git installed
- OpenAI API key and Serper.dev API key

Steps to Set Up Locally

1. **Clone the Repository**

```
git clone https://github.com/Harshalk2002/GENAI_003_PROJECT_02.git
cd GENAI_003_PROJECT_02
```

2. **Install Required Packages**

```
pip install -r requirements.txt
```

3. **Set Up Environment Variables**

- Configure your OpenAI and Serper.dev API keys either through a `.env` file or system environment variables.

4. **Run the Application**

```
streamlit run Panther_bot.py
```

5. **Interact with PantherBot**

- Open your browser at `http://localhost:8501`
- Begin exploring programs and chatting with PantherBot!

---

6. Team Roles & Responsibilities

Team Member	Role	Responsibilities
Harshal Kamble	Project Lead / LLM Engineer	Developed chatbot logic, integrated local LLM capabilities, managed interaction logging, oversaw GitHub repository and documentation.
Vishnu Sankhyan	Full Stack Developer	Designed Streamlit UI, developed chat interface, deployed app on Streamlit Cloud, managed CI/CD pipelines.
Abhay Prabhakar	UX / Feedback & Testing Lead	Designed conversational UX, conducted user testing, analyzed logs for improvements, and optimized the interface based on feedback.

The team maintained weekly meetings, clear documentation, and collaborative integration checkpoints to ensure smooth progress.

---

7. Challenges Faced

Performance Optimization

- Integrating multi-turn memory while maintaining low response latency was a major challenge. The team optimized session storage and prompt formatting to achieve optimal performance.

Session State Management

- Handling complex user journeys (program selection, conversation history, feedback) while maintaining a seamless experience demanded robust session management techniques.

Local LLM Integration

- Although initial deployment used OpenAI's hosted models, future-proofing the architecture for Ollama's offline local LLM deployments involved rethinking the API abstraction layer.

Deployment Scalability

- Ensuring that the Streamlit application remained performant even when multiple users interacted simultaneously required careful management of memory usage and session initialization.

---

8. Future Improvements

- Integration with Ollama for full offline chatbot functionality.
- Sentiment analysis on collected feedback to drive chatbot improvements.
- Expansion to include admission deadlines, scholarship information, and real-time advisor scheduling.
- Visual redesign of program cards for better mobile responsiveness.

---

9. Conclusion

PantherBot successfully addresses the critical problem of information overload and fragmentation faced by prospective students. With a seamless interface, intelligent search capabilities, and thoughtful engineering, PantherBot elevates the student decision-making experience. The project demonstrates practical AI integration, robust backend optimization, and effective team collaboration, meeting all academic and industry-level expectations.