**Project Report: Custom Chatbot with GPT-3.5 Turbo and Gradio**

**Table of Contents**

1. \*\*Introduction\*\*

- Project Overview

- Purpose

- Key Objectives

- Team Members

2. \*\*Methodology\*\*

- Project Architecture

- Technologies Used

- System Design

- Data Flow

3. \*\*Implementation\*\*

- OpenAI Integration

- Gradio Interface

- Custom ChatGPT Function

- User Messages Handling

- Assistant Responses

- System Initialization

4. \*\*User Guide\*\*

- How to Interact with the Chatbot

- Examples of User Input

5. \*\*Results and Discussion\*\*

- Chatbot's Performance

- User Feedback

- Limitations

- Future Enhancements

6. \*\*Conclusion\*\*

- Summary of Achievements

- Lessons Learned

7. \*\*Appendices\*\*

- Source Code

- API Key Security

- References

---

**1. Introduction**

**Project Overview**

This project involves the development of a custom chatbot using OpenAI's GPT-3.5 Turbo model and Gradio to provide conversational interactions with users. The chatbot is designed to respond to user queries and engage in conversation as a doctor.

**Purpose**

The purpose of this project is to demonstrate the capabilities of the GPT-3.5 Turbo model in a practical application, providing users with a conversational AI experience. Additionally, it serves as an example of how to integrate OpenAI models with Gradio for creating user interfaces.

**Key Objectives**

- Develop a chatbot that simulates conversation with a doctor.

- Implement the chatbot's core logic using the GPT-3.5 Turbo model.

- Create a user-friendly web interface using Gradio for easy interaction.

- Ensure the chatbot's ability to generate meaningful responses.

**2. Methodology**

Project Architecture

The project follows a client-server architecture with Gradio serving as the interface to collect user input and display chatbot responses. The GPT-3.5 Turbo model, provided by OpenAI, is used to generate responses based on user input.

Technologies Used

- OpenAI's GPT-3.5 Turbo

- Gradio

- Python

System Design

The system is designed to manage a conversation between the user and the chatbot, with messages being passed back and forth. The `CustomChatGPT` function handles user input and response generation. System initialization is set to "You are a doctor."

Data Flow

1. User inputs a message via the Gradio interface.

2. The message is sent to the `CustomChatGPT` function.

3. The function sends the user's message and the chatbot's messages history to GPT-3.5 Turbo.

4. The model generates a response, which is then displayed to the user through Gradio.

---

**3. Implementation**

OpenAI Integration

The OpenAI API is integrated with the project using the `openai` Python library, and the API key is set to authenticate requests.

Gradio Interface

Gradio is used to create a user-friendly interface, allowing users to interact with the chatbot through a web page.

Custom ChatGPT Function

The `CustomChatGPT` function handles user input and response generation. It appends user messages to the conversation history, sends the conversation history to the GPT-3.5 Turbo model, and processes the assistant's response.

User Messages Handling

User messages are collected via Gradio and added to the `messages` list with a "user" role.

Assistant Responses

The assistant's response is generated by the GPT-3.5 Turbo model and added to the `messages` list with an "assistant" role.

System Initialization

The chatbot is initialized with a system message, "You are a doctor."

---

**4. User Guide**

How to Interact with the Chatbot

1. Visit the web interface at [URL].

2. Enter your message in the text input field.

3. Press "Submit" or "Enter."

4. The chatbot will respond with a doctor-like message.

Examples of User Input

- User: "Can you tell me about the symptoms of COVID-19?"

- User: "I have a headache. What should I do?"

- User: "What's the best way to prevent the flu?"

---

**5. Results and Discussion**

Chatbot's Performance

The chatbot effectively engages in conversations with users, providing responses that align with the persona of a doctor. The GPT-3.5 Turbo model demonstrates strong language understanding and generation capabilities.

User Feedback

[Include user feedback if available.]

Limitations

- The chatbot may generate incorrect or nonsensical responses.

- Limited to a predefined persona (doctor).

- The chatbot may not handle highly technical or specialized medical queries well.

Future Enhancements

- Improve response quality and accuracy.

- Expand the chatbot's persona options.

- Handle more complex medical inquiries.

---

**6. Conclusion**

This project successfully demonstrates the creation of a custom chatbot using OpenAI's GPT-3.5 Turbo model and Gradio. The chatbot provides an interactive conversational experience with users, showcasing the potential of AI-driven chatbots in various domains. While there are limitations, continuous improvements can be made to enhance the chatbot's performance and capabilities.

---