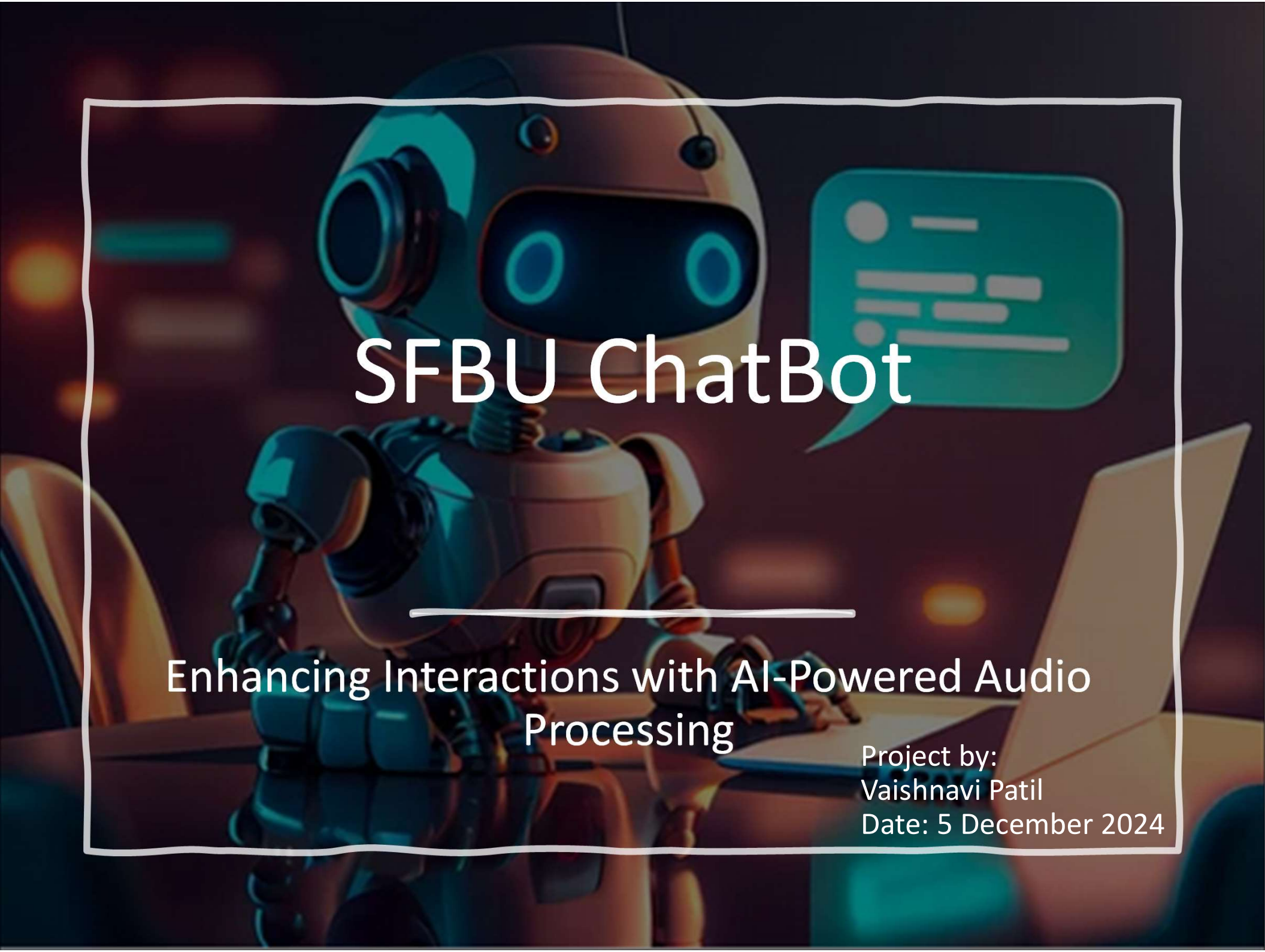


SFBU ChatBot

Enhancing Interactions with AI-Powered Audio Processing

Project by:
Vaishnavi Patil
Date: 5 December 2024

Table of Contents

1. Introduction
2. Design
    - Why was this approach taken?
    - Problem Identification and Understanding
    - Investigating and Comparing Solutions
3. Implementation
4. Testing
5. Enhancement Ideas
6. Conclusion
7. References
8. Appendix

Introduction

Objective: Develop a chatbot that converts user audio inputs into meaningful interactions using AI.

Key Features:

- Audio transcription using OpenAI Whisper
- GPT-powered responses
- AI-generated audio responses for an interactive experience

Significance: Improves accessibility and user engagement in educational platforms.

Design - Why This Approach?

Problem Identification:

- Lack of effective audio-based interactions in educational tools.
- Accessibility challenges for non-text inputs.

Investigation:

- Explored existing solutions (voice assistants, transcription tools).
- Considered open-source vs. proprietary tools for flexibility and scalability.

Comparison and Selection:

- OpenAI APIs offered the best trade-off between accuracy, ease of integration, and scalability.

Implementation - How This Was Done?

Frameworks Used:

- ✓ Flask for web server and backend processing.
- ✓ Bootstrap for user-friendly front-end design.
- ✓ OpenAI Whisper for transcription and GPT for response generation.

Steps:

1. **Frontend**: Designed a simple interface with start/stop recording buttons and response sections.
2. **Backend**:
   - Captures audio, transcribes it, and retrieves responses from GPT.
   - Converts GPT responses to audio using TTS.
3. **Vector Embedding**: Utilized LangChain for document context retrieval.

Testing

Functional Testing:

- Tested transcription accuracy with varied audio inputs.
- Verified GPT responses for relevance and context.

Performance Testing:

- Evaluated API response times under different loads.

Usability Testing:

- Conducted tests with a sample user group for interface and interaction feedback.

Enhancement Ideas

Future Improvements:

- Add multilingual support for broader accessibility.
- Incorporate real-time transcription and response feedback.
- Enhance document retrieval with larger datasets for more context-aware responses.
- Integrate with existing SFBU tools for seamless experience.

Conclusion

SUCCESSFULLY DEVELOPED A USER-FRIENDLY CHATBOT FOR SFBU.

COMBINED CUTTING-EDGE AI TECHNOLOGIES FOR TRANSCRIPTION, CONTEXT RETRIEVAL, AND RESPONSE GENERATION.

PROVIDES A SCALABLE AND INTERACTIVE PLATFORM FOR EDUCATIONAL ENGAGEMENT.

References

OpenAI API Documentation: (https://platform.openai.com/docs)

LangChain Documentation: (https://docs.langchain.com)

Flask Framework: (https://flask.palletsprojects.com/)

Bootstrap CSS Framework: (https://getbootstrap.com/)

Research on AI-powered accessibility in education: (https://scholar.google.com)

SFBU_Chatbot.py 5

Week 10 > SFBU ChatBot > SFBU_Chatbot.py > ...

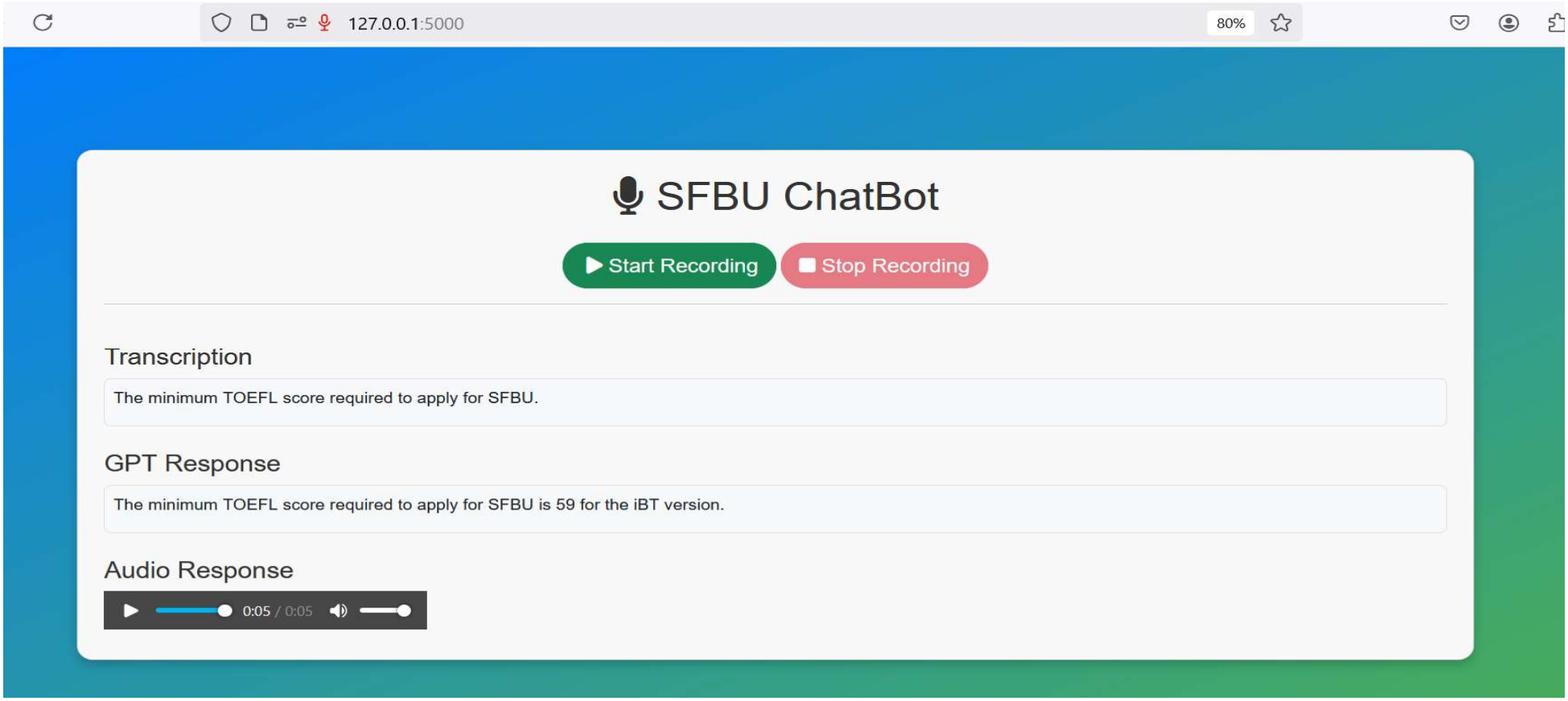
```python
import os
import openai
import datetime
from dotenv import load_dotenv, find_dotenv
from flask import Flask, request, jsonify, send_from_directory,
from werkzeug.utils import secure_filename
from langchain_community.document_loaders import PyPDFLoader
from langchain_openai import OpenAIEmbeddings
from langchain_chroma import Chroma
from langchain_openai import ChatOpenAI
from langchain.memory import ConversationBufferMemory
from pathlib import Path

# Load environment variables
load_dotenv(find_dotenv())
openai.api_key = os.getenv("OPENAI_API_KEY")

# Initialize Flask app
app = Flask(__name__)

# Directories
PDF_DIRECTORY = 'docs/cs229_lectures/'
PERSIST_DIRECTORY = 'docs/chroma/'

# Embedding function
embedding = OpenAIEmbeddings()

# Function to load PDFs and generate vector embeddings
def load_pdfs_and_store_embeddings(pdf_directory, persist_direct
    pdf_files = [f for f in os.listdir(pdf_directory) if f.endsw
    all_docs = []

    for pdf_file in pdf_files:
        pdf_path = os.path.join(pdf_directory, pdf_file)
        loader = PyPDFLoader(pdf_path)
        all_docs.extend(loader.load_and_split())

    return Chroma.from_documents(all_docs, embedding=embedding,

# Initialize vector store
vectordb = load_pdfs_and_store_embeddings(PDF_DIRECTORY, PERSIST

# Initialize LLM
current_date = datetime.datetime.now().date()
llm_name = "gpt-3.5-turbo" if current_date >= datetime.date(202
llm = ChatOpenAI(model=llm_name, temperature=0)

# Create retriever and memory
memory = ConversationBufferMemory(memory_key="chat_history", ret
retriever = vectordb.as_retriever()

# Helper Functions
def transcribe_audio(audio_file):
    """Transcribe audio file to text using OpenAI Whisper."""
```

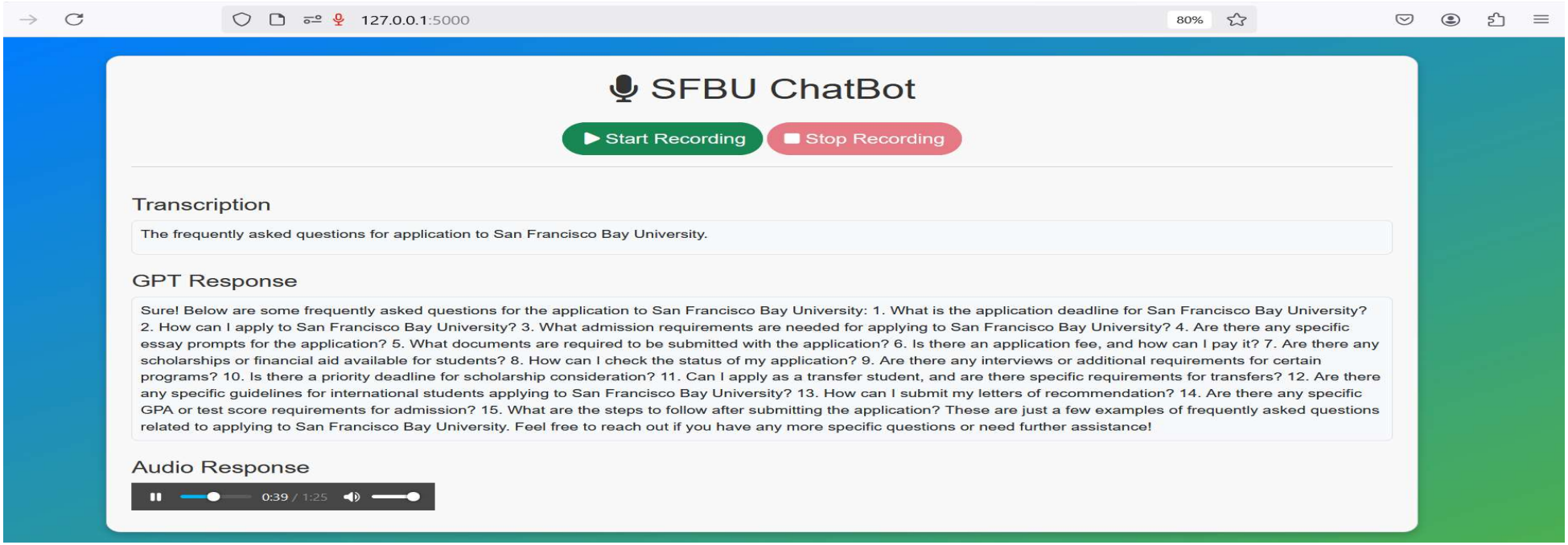
SFBU_Chatbot.py 5

Week 10 > SFBU ChatBot > SFBU_Chatbot.py > ...

```python
def generate_gpt_response(prompt):
    """Generate a GPT response for the given prompt."""
    response = openai.chat.completions.create(
        model="gpt-3.5-turbo-0125",
        messages=[
            {"role": "system", "content": "You are a helpful assistant."},
            {"role": "user", "content": prompt}
        ]
    )
    return response.choices[0].message.content

def text_to_speech_openai(text, output_path):
    """Convert text to speech using OpenAI's TTS."""
    try:
        response = openai.audio.speech.create(
            model="tts-1",  # Hypothetical model
            voice="nova",
            input=text
        )
        response.stream_to_file(output_path)
    except Exception as e:
        print(f"TTS processing failed: {e}")

# Routes
@app.route('/')
def index():
    """Serve the HTML file."""
    return send_from_directory('templates', 'index.html')

@app.route('/process-audio', methods=['POST'])
def process_audio():
    if 'audio' not in request.files:
        return jsonify({'error': 'No audio file provided'}), 400

    audio_file = request.files['audio']
    audio_path = "temp_audio_input.wav"
    audio_file.save(audio_path)

    try:
        transcription_text = transcribe_audio(open(audio_path, "rb"))
        docs = vectordb.similarity_search(transcription_text)
        context = "\n".join([doc.page_content for doc in docs])
        prompt = f"Use the following context to answer the question:\n\n{con
        gpt_response = generate_gpt_response(prompt)

        output_speech_path = Path("response_audio.mp3")
        text_to_speech_openai(gpt_response, output_speech_path)

        return jsonify({
            'transcription': transcription_text,
            'gptResponse': gpt_response,
            'audioUrl': str(output_speech_path)
        })
    except Exception as e:
        print(f"Error processing audio: {e}")
        return jsonify({'error': 'Processing failed'}), 500

@app.route('/response_audio.mp3')
def serve_audio():
    """Serve the generated audio file."""
    audio_file = "response_audio.mp3"
    try:
        return send_file(audio_file, as_attachment=False)
    except FileNotFoundError:
        return "Audio file not found", 404
```

Appendix


127.0.0.1:5000
80%
SFBU ChatBot
Start Recording
Stop Recording
Transcription
The minimum TOEFL score required to apply for SFBU.
GPT Response
The minimum TOEFL score required to apply for SFBU is 59 for the iBT version.
Audio Response
0:05 / 0:05


127.0.0.1:5000
80%
SFBU ChatBot
Start Recording
Stop Recording
Transcription
The frequently asked questions for application to San Francisco Bay University.
GPT Response
Sure! Below are some frequently asked questions for the application to San Francisco Bay University: 1. What is the application deadline for San Francisco Bay University? 2. How can I apply to San Francisco Bay University? 3. What admission requirements are needed for applying to San Francisco Bay University? 4. Are there any specific essay prompts for the application? 5. What documents are required to be submitted with the application? 6. Is there an application fee, and how can I pay it? 7. Are there any scholarships or financial aid available for students? 8. How can I check the status of my application? 9. Are there any interviews or additional requirements for certain programs? 10. Is there a priority deadline for scholarship consideration? 11. Can I apply as a transfer student, and are there specific requirements for transfers? 12. Are there any specific guidelines for international students applying to San Francisco Bay University? 13. How can I submit my letters of recommendation? 14. Are there any specific GPA or test score requirements for admission? 15. What are the steps to follow after submitting the application? These are just a few examples of frequently asked questions related to applying to San Francisco Bay University. Feel free to reach out if you have any more specific questions or need further assistance!
Audio Response
0:39 / 1:25

URLs

Google Slide:
https://docs.google.com/presentation/d/1s6f3ZYjvlFQuZjxeg58Q-m0cjkhWN38v-wjgFekmPQA/edit?usp=sharing

GitHub URL:

https://github.com/vaishnavi477/Machine-Learning/tree/main/AI-Based%20Alexa/Real-time%20Speech-to-Text-to-Speech/SFBU%20ChatBot

Thank You