SFBU Customer Support System - Speech to Text to Speech

DS572

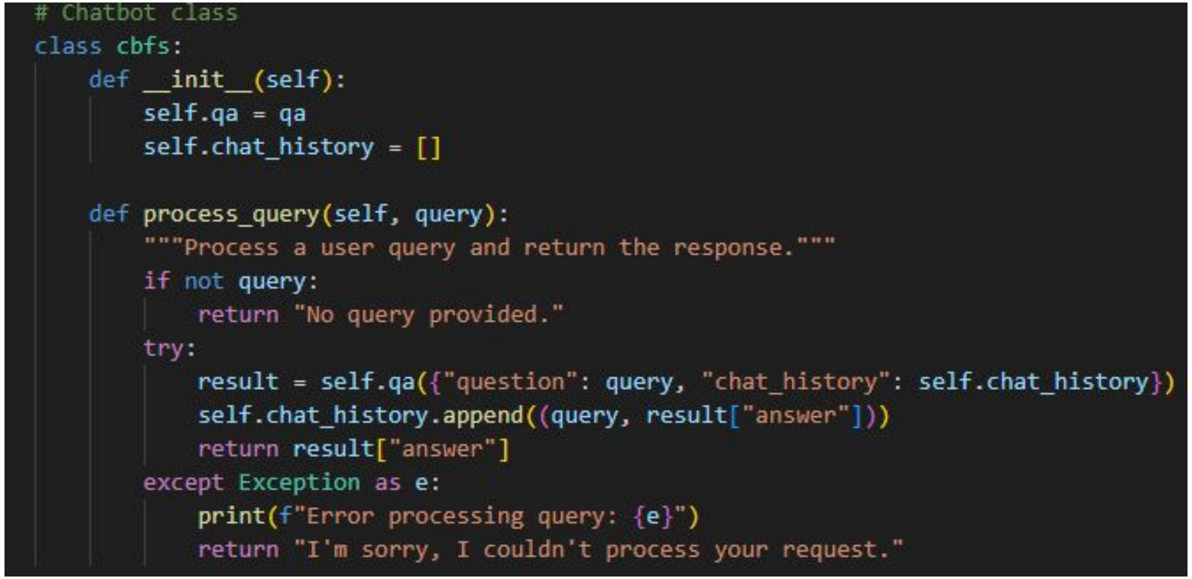
Emily Weng 20016

Table of Contents

Step 1: Implement SFBU Customer Support System - text

Use previous text chatbot from previous Week 5 homework to implement into the whole system.

We would need the cbfs class as well

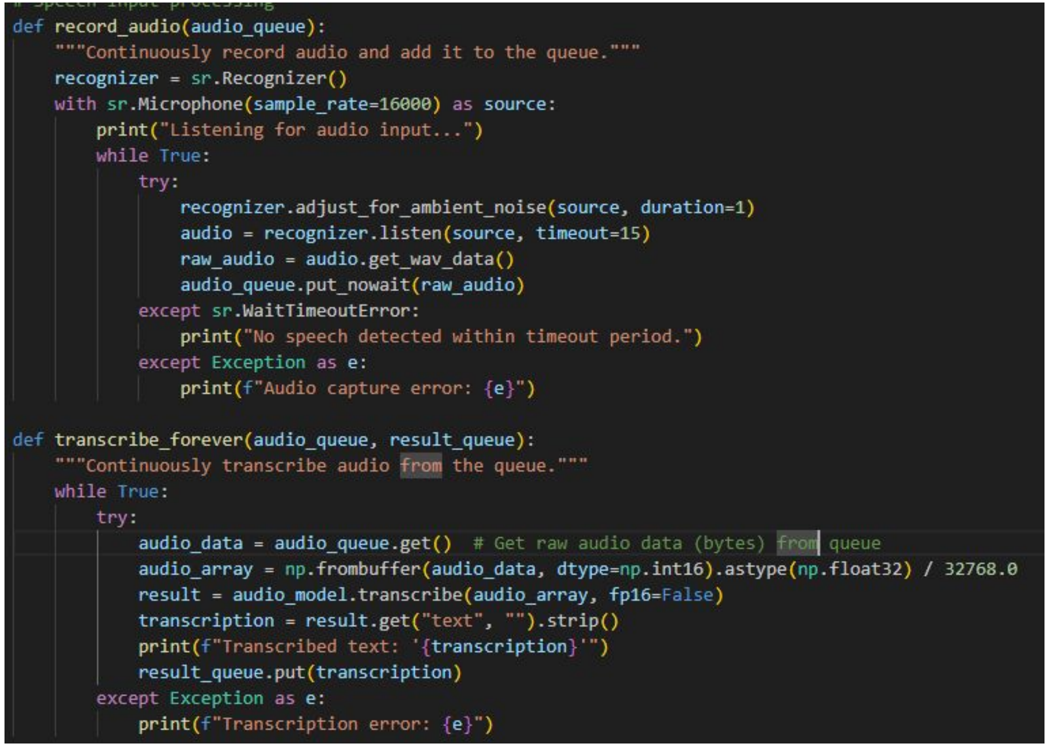


```python
# Chatbot class
class cbfs:
    def __init__(self):
        self.qa = qa
        self.chat_history = []

    def process_query(self, query):
        """Process a user query and return the response."""
        if not query:
            return "No query provided."
        try:
            result = self.qa({"question": query, "chat_history": self.chat_history})
            self.chat_history.append((query, result["answer"]))
            return result["answer"]
        except Exception as e:
            print(f"Error processing query: {e}")
            return "I'm sorry, I couldn't process your request."
```

Step 2: Implement Real-time Speech to Text to Speech

Implement the steps from Speech to Text to Speech model as well and include the function in your code

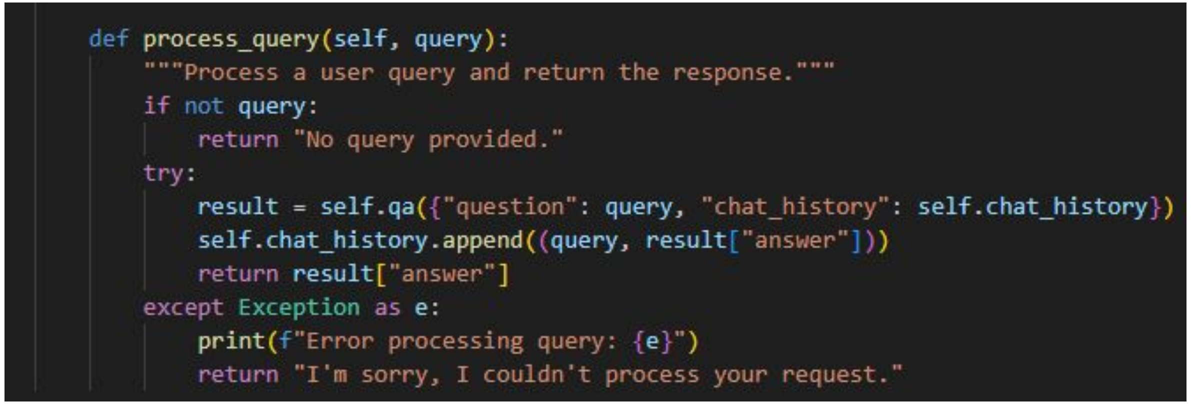


```
# Speech input processing
def record_audio(audio_queue):
    """Continuously record audio and add it to the queue."""
    recognizer = sr.Recognizer()
    with sr.Microphone(sample_rate=16000) as source:
        print("Listening for audio input...")
        while True:
            try:
                recognizer.adjust_for_ambient_noise(source, duration=1)
                audio = recognizer.listen(source, timeout=15)
                raw_audio = audio.get_wav_data()
                audio_queue.put_nowait(raw_audio)
            except sr.WaitTimeoutError:
                print("No speech detected within timeout period.")
            except Exception as e:
                print(f"Audio capture error: {e}")

def transcribe_forever(audio_queue, result_queue):
    """Continuously transcribe audio from the queue."""
    while True:
        try:
            audio_data = audio_queue.get()  # Get raw audio data (bytes) from queue
            audio_array = np.frombuffer(audio_data, dtype=np.int16).astype(np.float32) / 32768.0
            result = audio_model.transcribe(audio_array, fp16=False)
            transcription = result.get("text", "").strip()
            print(f"Transcribed text: '{transcription}'")
            result_queue.put(transcription)
        except Exception as e:
            print(f"Transcription error: {e}")
```

Step 3: Enhance Step 2 by adding the features of the project implemented in Step 1

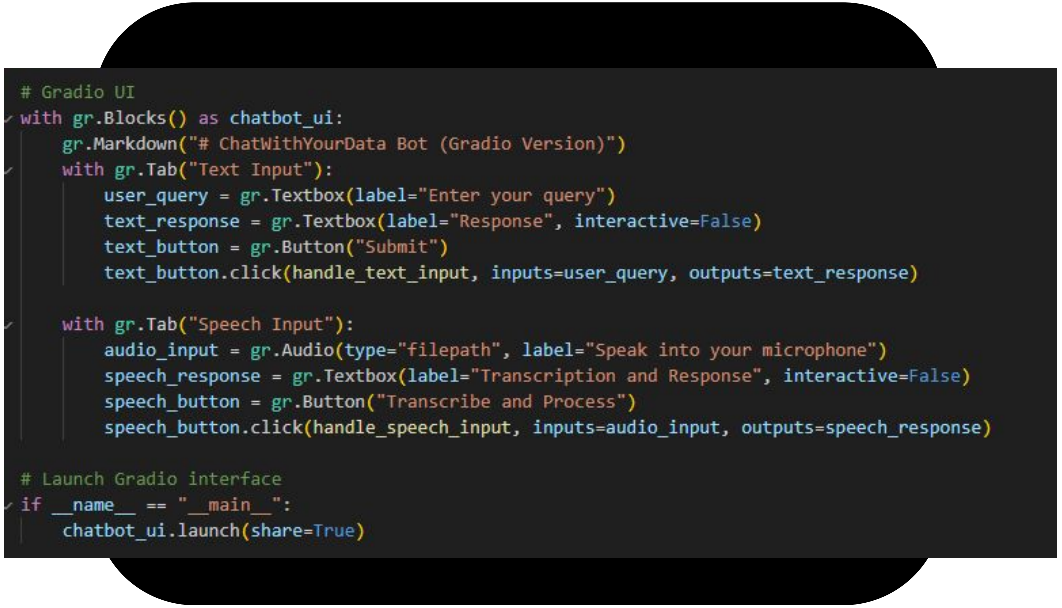
The reply enhancement was implemented in the cbfs class as a process query



```python
    def process_query(self, query):
        """Process a user query and return the response."""
        if not query:
            return "No query provided."
        try:
            result = self.qa({"question": query, "chat_history": self.chat_history})
            self.chat_history.append((query, result["answer"]))
            return result["answer"]
        except Exception as e:
            print(f"Error processing query: {e}")
            return "I'm sorry, I couldn't process your request."
```

Step 4: GUI using Gradio

Created a GUI using Gradio to show test the text and speech I/O.

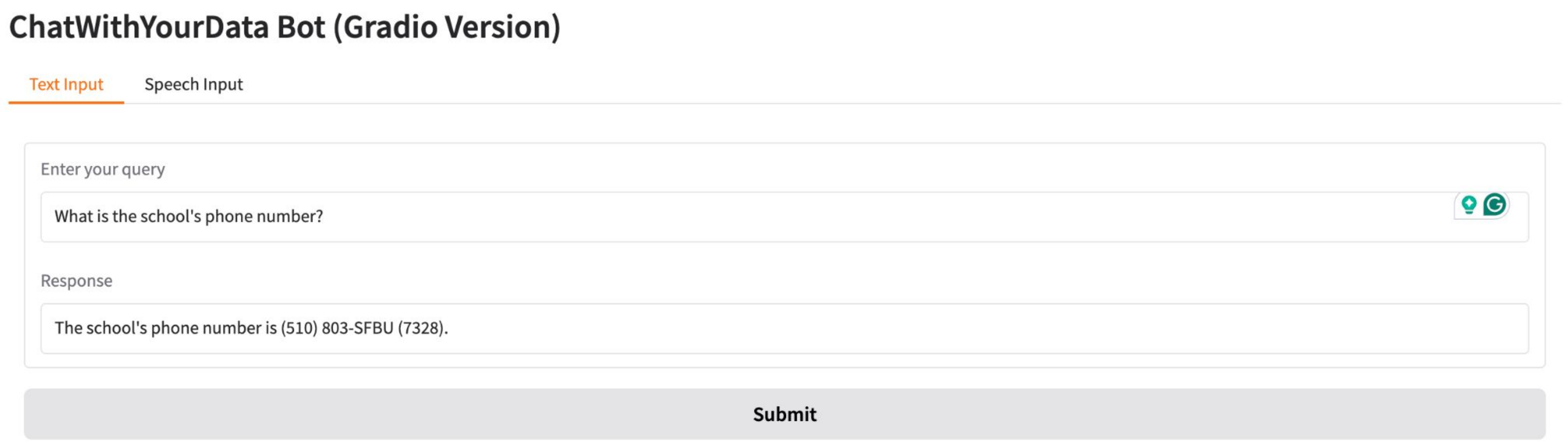


```python
# Gradio UI
with gr.Blocks() as chatbot_ui:
    gr.Markdown("# ChatWithYourData Bot (Gradio Version)")
    with gr.Tab("Text Input"):
        user_query = gr.Textbox(label="Enter your query")
        text_response = gr.Textbox(label="Response", interactive=False)
        text_button = gr.Button("Submit")
        text_button.click(handle_text_input, inputs=user_query, outputs=text_response)

    with gr.Tab("Speech Input"):
        audio_input = gr.Audio(type="filepath", label="Speak into your microphone")
        speech_response = gr.Textbox(label="Transcription and Response", interactive=False)
        speech_button = gr.Button("Transcribe and Process")
        speech_button.click(handle_speech_input, inputs=audio_input, outputs=speech_response)

# Launch Gradio interface
if __name__ == "__main__":
    chatbot_ui.launch(share=True)
```

Output:



ChatWithYourData Bot (Gradio Version)

Enter your query

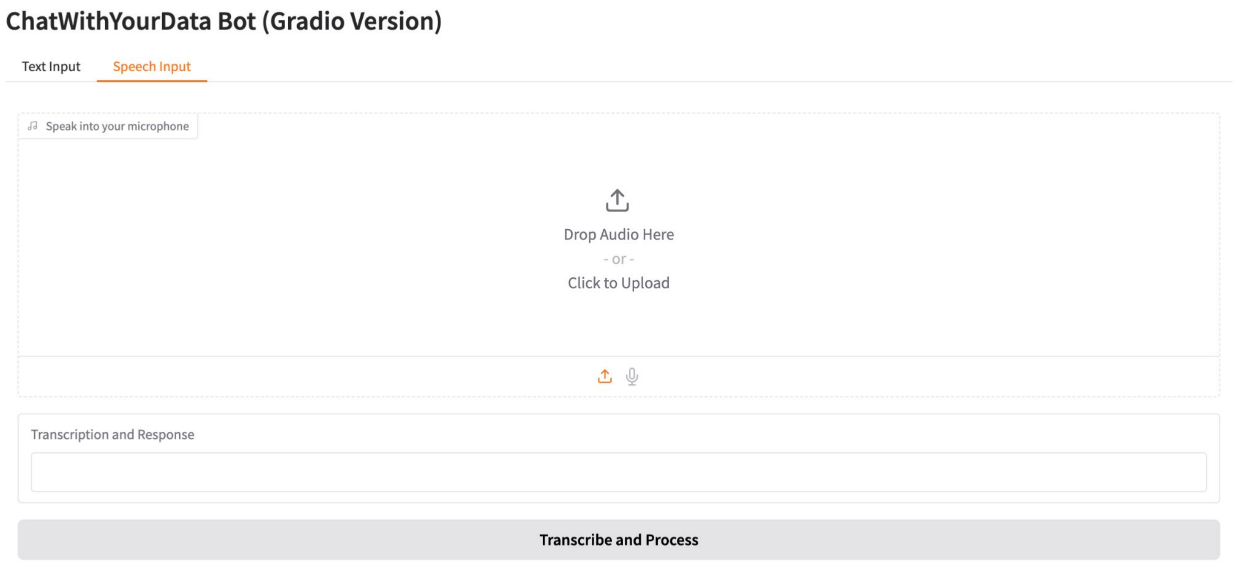
What is the school's phone number?

Response

The school's phone number is (510) 803-SFBU (7328).

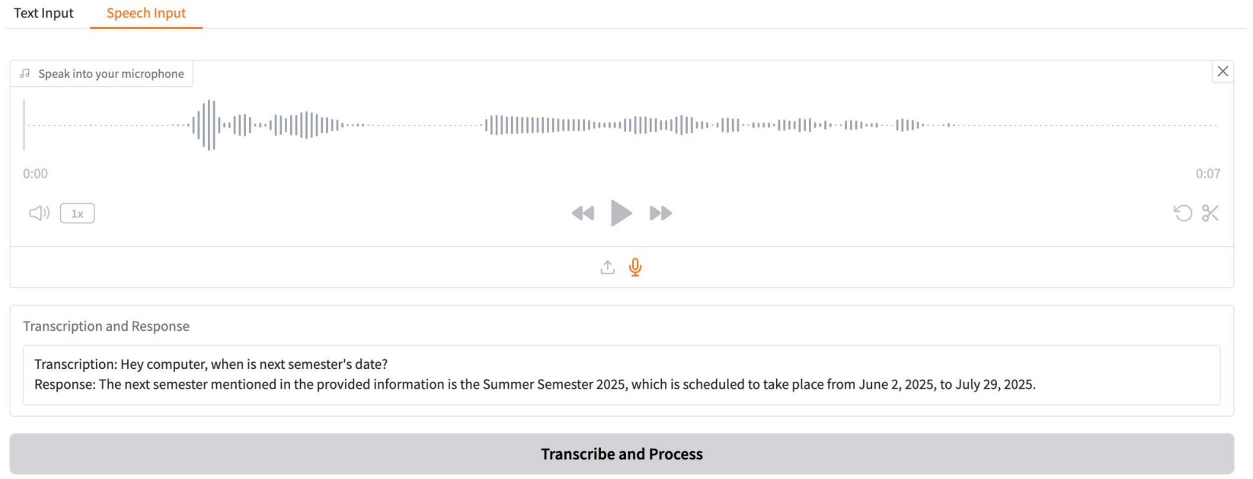
Text to text output

Output


ChatWithYourData Bot (Gradio Version)
Text Input
Speech Input
Speak into your microphone
Drop Audio Here
- or -
Click to Upload
Transcription and Response
Transcribe and Process

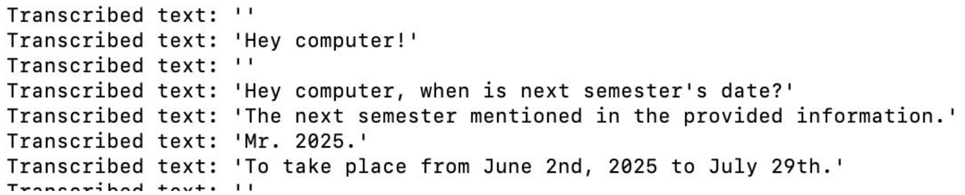
Option to upload audio file

Output



Transcription: Hey computer, when is next semester's date?
Response: The next semester mentioned in the provided information is the Summer Semester 2025, which is scheduled to take place from June 2, 2025, to July 29, 2025.

Output for speech to text to speech, the output would be printed on terminal as well.



```
Transcribed text: ''
Transcribed text: 'Hey computer!'
Transcribed text: ''
Transcribed text: 'Hey computer, when is next semester's date?'
Transcribed text: 'The next semester mentioned in the provided information.'
Transcribed text: 'Mr. 2025.'
Transcribed text: 'To take place from June 2nd, 2025 to July 29th.'
Transcribed text: ''
```

Conclusion

- More functions could be enhanced to make it into a more complete chatbot
- Could also have the option of loading more data
- The speech to text to speech takes more time to process.

Github

https://github.com/emilywengster/sfbu/tree/main/Machine%20Learning/ChatGPT/Customer%20Support%20System/Moderation%2C%20Classification%2C%20Checkout%20and%20Evaluation

THANK YOU

COMPANY NAME