app.py

```python
import os
import logging
from flask import Flask, render_template, request, jsonify
from voice_assistant import VoiceAssistant

# Configure logging
logging.basicConfig(level=logging.DEBUG)
logger = logging.getLogger(__name__)

# Initialize the Flask application
app = Flask(__name__)
app.secret_key = os.environ.get("SESSION_SECRET", "dev_secret_key")

# Initialize the voice assistant
voice_assistant = VoiceAssistant()

@app.route('/')
def index():
    """Render the main page of the voice assistant interface."""
    return render_template('index.html')

@app.route('/start_listening', methods=['POST'])
def start_listening():
    """Start the voice recognition process."""
    try:
        voice_assistant.start_listening()
        return jsonify({"status": "success", "message": "Listening started"})
    except Exception as e:
        logger.error(f"Error starting listening: {e}")
        return jsonify({"status": "error", "message": str(e)})

@app.route('/stop_listening', methods=['POST'])
def stop_listening():
    """Stop the voice recognition process."""
    try:
        voice_assistant.stop_listening()
        return jsonify({"status": "success", "message": "Listening stopped"})
    except Exception as e:
        logger.error(f"Error stopping listening: {e}")
        return jsonify({"status": "error", "message": str(e)})

@app.route('/process_command', methods=['POST'])
def process_command():
    """Process a voice command received from the client."""
    try:
        audio_data = request.json.get('audio_data')
        if not audio_data:
            return jsonify({"status": "error", "message": "No audio data provided"})

        result = voice_assistant.process_audio(audio_data)
        return jsonify({"status": "success", "result": result})
    except Exception as e:
        logger.error(f"Error processing command: {e}")
        return jsonify({"status": "error", "message": str(e)})

@app.route('/get_command_history', methods=['GET'])
def get_command_history():
    """Get the history of commands processed by the voice assistant."""
    try:
        history = voice_assistant.get_command_history()
        return jsonify({"status": "success", "history": history})
    except Exception as e:
        logger.error(f"Error getting command history: {e}")
        return jsonify({"status": "error", "message": str(e)})

@app.route('/execute_text_command', methods=['POST'])
def execute_text_command():
    """Execute a command received as text."""
    try:
        command_text = request.json.get('command')
        if not command_text:
            return jsonify({"status": "error", "message": "No command provided"})

        result = voice_assistant.process_text_command(command_text)
        return jsonify({"status": "success", "result": result})
    except Exception as e:
        logger.error(f"Error executing text command: {e}")
        return jsonify({"status": "error", "message": str(e)})

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000, debug=True)

```