KWADWO AMPONSAH AMPOFO

J00946242@students.jsums.edu

**A SPEECH TRANSCRIPTION AND EVALUATION TOOL WITH CANVAS LTI INTEGRATION**

Abstract

This report documents the development of a web-based platform that transcribes, evaluates, and delivers feedback on spoken language responses using SYSTRAN's FasterWhisper model

Contents

[**Executive Summary** 3](#_Toc202965999)

[Background and Motivation 4](#_Toc202966000)

[1. Introduction 5](#_Toc202966001)

[2. Project Overview 5](#_Toc202966002)

[3. Files and Structure 5](#_Toc202966003)

[4. Dependencies and Setup 6](#_Toc202966004)

[- Configuration 6](#_Toc202966005)

[5. Code Overview and Explanation 7](#_Toc202966006)

[5.1 app.py 7](#_Toc202966007)

[5.2 whisper\_eval.py 7](#_Toc202966008)

[5.3 lti\_config.py 7](#_Toc202966009)

[5.4 LTI Configuration Files 7](#_Toc202966010)

[5.5 Frontend Assets (static/ and templates/) 8](#_Toc202966011)

[6. Usage Instructions 8](#_Toc202966012)

[7. Testing Approach 9](#_Toc202966013)

[Functional Testing 9](#_Toc202966014)

[Model Validation 9](#_Toc202966015)

[LTI Integration Testing 9](#_Toc202966016)

[UI Testing 9](#_Toc202966017)

[Results 9](#_Toc202966018)

[8. Limitations and Future Work 10](#_Toc202966019)

[9. Security Considerations 11](#_Toc202966020)

[10. Appendices 12](#_Toc202966021)

[Appendix A: app.py 12](#_Toc202966022)

[Appendix B: whisper\_eval.py 17](#_Toc202966023)

[Appendix C: lti\_config.py 18](#_Toc202966024)

[Appendix D: tool\_config.json 18](#_Toc202966025)

[Appendix E: tool\_config\_backend.json 19](#_Toc202966026)

[Appendix F: tool\_config\_template.json 19](#_Toc202966027)

[Appendix G: requirements.txt 20](#_Toc202966028)

[Appendix H: jwk\_public.json 20](#_Toc202966029)

[Appendix I: CSS (static/css/style.css) 20](#_Toc202966030)

[Appendix J: JavaScript (static/js/main.js) 21](#_Toc202966031)

[Appendix K: HTML Template (templates/index.html) 21](#_Toc202966032)

[References 22](#_Toc202966033)

# Executive Summary

This report documents the development of a web-based platform that transcribes, evaluates, and delivers feedback on spoken language responses using SYSTRAN's FasterWhisper model. Designed to integrate seamlessly into educational platforms like Canvas LMS via the Learning Tools Interoperability (LTI) 1.3 standard, the system offers students a streamlined way to submit oral assessments and receive immediate, rubric-based evaluation.

The project addresses a key challenge in modern education: the need for scalable, automated, and intelligent tools that can assess spoken responses fairly and efficiently. By leveraging FasterWhisper for high-accuracy transcription and a customizable evaluation rubric, this system enables instructors to evaluate clarity, relevance, organization, and grammar in student submissions - quickly and consistently.

The backend is implemented in Python using Flask and is capable of handling multiple audio file formats and real-time recordings. Users can either upload pre-recorded audio files or capture responses directly via the web interface. The transcribed content is processed, evaluated against predefined rubric keywords, and presented to the user along with qualitative feedback.

From a systems integration perspective, the application supports secure LTI launches, allowing educational institutions to embed the tool directly within Canvas assignments. Authentication, session management, and public key handling follow industry best practices as defined by the IMS Global Learning Consortium (IMS Global Learning Consortium., 2025).

This report presents the entire project lifecycle - from design rationale to implementation details - along with source code, deployment instructions, and an in-depth walkthrough of the system’s frontend and backend components. Future improvements will focus on expanding the evaluation model, bundling it into a mobile application, and enhancing the user interface for accessibility and performance.

# Background and Motivation

In the evolving landscape of digital education, the ability to assess spoken responses has become increasingly important. Oral assessments play a critical role in language learning, communication skills development, and reflective evaluations. However, traditional methods of evaluating oral submissions - such as manual listening and scoring - are time-consuming and difficult to scale, especially in large classrooms or online learning environments.

The rapid advancement of automatic speech recognition (ASR) and natural language processing (NLP) has opened new opportunities for improving how educational institutions handle oral submissions. Tools powered by ASR can now provide real-time transcription of spoken language with high accuracy. However, integrating these technologies into Learning Management Systems (LMS) like Canvas remains a challenge due to compatibility, authentication requirements, and lack of standardization.

This project, Oral Response Evaluator, was conceived to address these challenges by providing:

* A lightweight, secure, and accessible interface for students to record or upload speech.
* A high-accuracy transcription backend using Systran’s FasterWhisper (SYSTRAN, 2025), an optimized and efficient implementation of the Whisper speech model (OpenAI, 2025).
* An automatic rubric-based evaluation of responses, providing immediate feedback on important criteria like clarity, organization, relevance, and grammar.
* Seamless integration with Canvas LMS using LTI 1.3, ensuring secure authentication, context awareness, and compatibility with academic workflows.

By combining open-source tools and standardized protocols, this system bridges the gap between modern AI-based evaluation and classroom implementation. It reduces the burden on educators while promoting a more interactive and responsive learning experience for students.

Ultimately, this work is motivated by the vision of creating a smart, scalable, and transparent tool for academic speech assessment - one that supports inclusive education and enhances feedback loops in both in-person and virtual classrooms.

# 1. Introduction

This project implements an Oral Response Evaluator that transcribes and evaluates speech responses using Systran’s Faster Whisper model. It integrates with Canvas LMS through LTI 1.3 to provide seamless authentication and course context, allowing instructors and students to access oral transcription and evaluation directly within Canvas.

# 2. Project Overview

The project consists of:

* A Flask backend server that hosts the speech transcription and evaluation model.
* LTI integration endpoints to support secure launches from Canvas.
* A frontend interface for recording or uploading audio.
* Evaluation logic that scores responses based on keyword rubrics.

# 3. Files and Structure

| **File** | **Description** |
| --- | --- |
| app.py | Main Flask application with routes for LTI, transcription, upload. |
| whisper\_eval.py | Transcription and evaluation logic using Faster Whisper. |
| lti\_config.py | Alternative Flask app handling LTI login and launch endpoints. |
| tool\_config.json | Public LTI tool configuration JSON for Canvas integration. |
| tool\_config\_backend.json | Backend LTI configuration with client secrets (sample). |
| tool\_config\_template.json | Template config for easy deployment setup. |
| requirements.txt | Python dependencies required to run the app. |
| jwk\_public.json | Public key JSON Web Key Set (JWKS) file for OAuth security. |
| static/ | Static frontend assets (CSS, JS, images). |
| templates/ | HTML templates for frontend UI. |
| keys/ | RSA keys for JWT signing and verification. |

# 4. Dependencies and Setup

**-** Required Python Packages

* Flask
* pylti1p3 (for LTI 1.3 support)
* python-dotenv (for environment variable support)
* requests
* openai-whisper (Faster Whisper transcription model)

You can install dependencies using:

*bash*

pip install -r requirements.txt

## Configuration

* Provide your Canvas Client ID, Deployment ID, and RSA keys in tool\_config\_backend.json and the keys/ directory.
* Update URLs in tool\_config.json to match your deployment endpoint (e.g., your public ngrok or server URL).

# 5. Code Overview and Explanation

## 5.1 app.py

* **Purpose:** Flask app implementing routes for LTI login, launch, transcription API, and upload UI.
* Uses pylti1p3 (Viskov, 2025) for LTI 1.3 OIDC login flow and launch validation.
* Limits transcription requests with Flask-Limiter.
* Saves uploaded audio files and passes them to the transcription and evaluation module.
* Provides an HTML interface allowing users to upload or record audio.

## 5.2 whisper\_eval.py

* Contains WhisperTranscriber class to load and run Faster Whisper transcription on audio.
* SimpleEvaluator class checks for rubric keywords and provides a score and feedback.
* transcribe\_and\_evaluate() combines both to return a transcript and evaluation metrics.

## 5.3 lti\_config.py

* A simpler Flask app (Pallets, 2025) focusing on LTI login and launch endpoints.
* Stores session launch info.
* Useful for testing LTI flows separately.

## 5.4 LTI Configuration Files

* tool\_config.json and tool\_config\_backend.json define the tool’s integration points with Canvas, including OAuth URLs and keys.
* tool\_config\_template.json serves as a deployment starter template.

## 5.5 Frontend Assets (static/ and templates/)

* CSS file styles the upload and recording interface.
* JavaScript handles audio recording and upload asynchronously.
* HTML template renders transcription results or upload UI depending on context.

# 6. Usage Instructions

* Clone the project and install Python dependencies.
* Configure LTI keys and Canvas client IDs.
* Run the Flask app (python app.py).
* Expose your local server via a public URL (ngrok or cloud).
* Configure Canvas to recognize your LTI tool with the public URLs.
* Use the Canvas assignment selection to launch the oral response evaluator.
* Upload or record audio responses.
* Receive transcript and rubric-based evaluation feedback instantly.

# 7. Testing Approach

## Functional Testing

* + Verified upload and recording functionality across browsers.
  + Ensured compatibility with multiple audio formats: .wav, .mp3, .webm, .m4a, .flac, and .mp4.
  + Validated real-time transcription and scoring with sample responses.

## Model Validation

* + Used sample speech inputs with known expected transcripts.
  + Compared Whisper outputs against manual transcripts for accuracy.

## LTI Integration Testing

* + OIDC login tests and LTI 1.3 launches via Canvas LMS Developer Keys yet to be conducted.
  + Verified secure launch payloads, deployment IDs, and role parsing yet to be conducted.

## UI Testing

* + Ensured form validation, real-time status updates, and responsive rendering across devices.

## Results

* + The system achieves highly reliable transcription on clean audio.
  + Evaluation scores correctly reflect rubric term presence.
  + Real-time recording works in Chromium-based browsers.

# Limitations and Future Work

Despite successful implementation, the project has several limitations and opportunities for expansion:

**8.1** Current Limitations

* **Model Limitations:** The base FasterWhisper model performs well but may misrecognize speech in low-quality recordings or with heavy accents.
* **Deployment Fragility:** Hosting via Ngrok is not suited for production. A more robust deployment (e.g., on GCP, AWS, or Azure) is required.
* **No Authentication for Frontend Uploads:** Currently, uploads outside Canvas LTI are unauthenticated.

**8.2** Future Enhancements

* Upgrade to **larger Whisper models** for more accurate transcription.
* Support **CSV/JSON exports** for instructor-gradebooks.
* Secure and scale the app using **Docker**, **HTTPS**, and **production-grade databases**.
* Extend LTI scoring logic to **push grades directly to Canvas** using the LTI AGS (Assignment and Grade Services) spec.

# Security Considerations

Given the system interacts with student responses and educational platforms, it adheres to several security principles:

* **File Upload Restrictions:** Only whitelisted audio formats are accepted. Filenames are sanitized using secure\_filename.
* **Size Limitations:** Flask is configured with a 10MB file size cap to prevent Denial of Service (DoS) via large uploads.
* **Session Protection:** HTTP-only and SameSite cookie settings are enforced to mitigate Cross-Site Request Forgery and Cross-Site Scripting (CSRF/XSS).
* **LTI Security:**
  + The tool authenticates launches via OpenID Connect (OIDC**)** with JSON Web Key Sets (JWKS).
  + Deployment ID and client validation ensure only known LMS instances can launch the tool.
* **Temporary File Handling:** Uploaded files are saved using tempfile and removed after evaluation.
* **No Persistent Storage of Transcripts or Audio** (currently): All processing is ephemeral unless extended in future builds.

# 10. Appendices

## Appendix A: app.py

Python

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

## Appendix B: whisper\_eval.py

Python

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

## Appendix C: lti\_config.py

python

A screenshot of a computer

AI-generated content may be incorrect.

## Appendix D: tool\_config.json

json

A screenshot of a computer

AI-generated content may be incorrect.

## Appendix E: tool\_config\_backend.json

json

A screenshot of a computer

AI-generated content may be incorrect.

## Appendix F: tool\_config\_template.json

json

A screenshot of a computer

AI-generated content may be incorrect.

## Appendix G: requirements.txt

**A screenshot of a computer

AI-generated content may be incorrect.**

## Appendix H: jwk\_public.json

json

A screenshot of a computer

AI-generated content may be incorrect.

## Appendix I: CSS (static/css/style.css)

css

A screenshot of a computer

AI-generated content may be incorrect.

## Appendix J: JavaScript (static/js/main.js)

js

A screenshot of a computer

AI-generated content may be incorrect.

## Appendix K: HTML Template (templates/index.html)

html

A screenshot of a computer

AI-generated content may be incorrect.

# References

IMS Global Learning Consortium. (2025, July 9). *Learning Tools Interoperability (LTI)® Core Specification*. Retrieved from https://www.imsglobal.org/spec/lti/v1p3/

OpenAI. (2025, July 9). *Github.* Retrieved from https://github.com/openai/whisper

Pallets. (2025, July 9). Retrieved from Flask: https://flask.palletsprojects.com/en/stable/

SYSTRAN. (2025, July 9). Retrieved from Github: https://github.com/SYSTRAN/faster-whisper

Viskov, D. (2025, July 9). Retrieved from Github: https://github.com/dmitry-viskov/pylti1.3