Data Management - Research Internship Research and Evaluation of Techniques for Achieving Accessibility in Education.

Author: Aditi Ashok Ingale.
841858

January 2025

UNIVERSITY OF TECHNOLOGY
IN THE EUROPEAN CAPITAL OF CULTURE
CHEMNITZ

Abstract

In a project like the Speech-to-Text Converter, one can really feel the actual realization of Speech Recognition in real time, helped by the Web Speech API, which came to solve many problems for accessibility and productivity with the use of transcription systems. This report describes the state of implementation and its architecture, the challenges that present, speech recognizers, and the evaluation of existing features or enhancements that allow enrichment / extension of the system functionality and accessibility in various dimensions.

In other words, accessibility in education provides equal learning opportunities for all, with and without disabilities. The purpose of this research paper is to present various methods to ensure accessibility to all students with visual or hearing impairments, and to assess them. The current methodology for communicating between the instructor and the students is considered to derive strategies to help modify teaching techniques to achieve effective educational outcomes for all learners.

Contents

1 Introduction

Accessibility in education is essential for fostering inclusive and equal opportunities for all learners. This research investigates whether specific techniques, such as screen readers, text-to-speech technologies, and sign language interpreters, effectively enhance accessibility for students with visual and hearing impairments.

Speech recognition technology has revolutionized human-computer interaction by enabling machines to process and understand spoken language. Its applications range from accessibility tools for people with hearing impairments to productivity-enhancing transcription systems in workplaces. This project implements a web-based Speech-to-Sign Language Converter leveraging the Web Speech API, designed for lightweight, browser-based usage.

The study also evaluates best practices in slide design and other teaching methodologies to ensure clarity and inclusiveness. By addressing the question, Can targeted accessibility techniques improve educational experiences for students with disabilities?, this report aims to provide practical insights for creating more equitable learning environments.

2 Literature Review

Ensuring accessibility in education has been the focus of extensive research, particularly in the context of supporting students with disabilities.

- A Various assistive technologies, such as screen readers and text-to-speech tools, have proven to be critical for visually impaired learners by providing them access to textual content in digital formats.
- sign language interpreters and advancements in speech recognition technologies have revolutionized communication for hearing-impaired students, making educational settings more inclusive.
- Several studies have explored the role of automated systems in accessibility. For example, tools like Speech-to-Text converters, which leverage technologies such as the Web Speech API, highlight the potential of lightweight, browser-based solutions to enhance accessibility in real-time.
- These tools not only assist in overcoming communication barriers but also contribute to productivity by simplifying transcription and interaction processes.
- Best practices in teaching methodologies, including the design of accessible slides, emphasize clarity, high contrast, and simplicity to accommodate diverse learner needs.
- Research has consistently underlined the importance of alternative text for images, descriptive captions, and structured content to improve comprehension for students with visual and cognitive impairments.
- Moreover, systematic reviews of assistive technologies have highlighted their growing impact on fostering inclusive and improving learning outcomes.

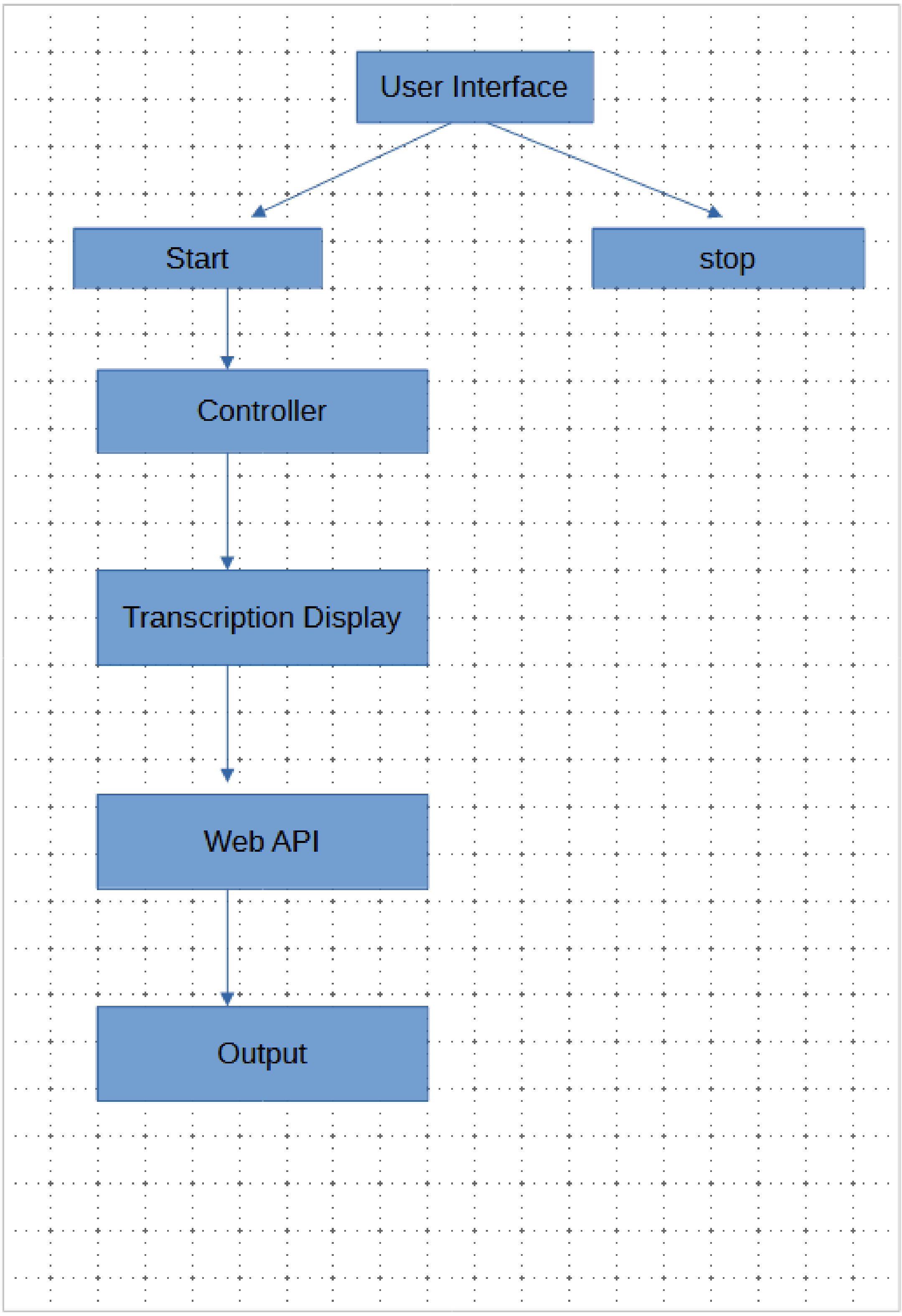
Examples of industry use cases include:

1. Real-time meeting transcription tools.
2. Voice-controlled applications in smart devices.
3. Accessibility software for the differently-able.

3 System Architecture

The system architecture consists of four main components:

- **User Interface (UI)**: This functionality enables users to seamlessly record audio through an intuitive interface and have the spoken content automatically converted into text and sign language. It provides a user-friendly platform for recording, sign language and transcription, making the process accessible and efficient for various applications, such as note-taking, accessibility support, or real-time communication enhancement.
- **Controller**: This feature handles user inputs and actions, seamlessly integrating with the Web Speech API to facilitate speech recognition and transcription. By leveraging the API, it processes spoken language efficiently and ensures smooth interaction between the user and the application, enabling accurate and responsive performance for speech-to-sign language functionality.
- **Web Speech API**: This functionality captures spoken input from the user and processes it to generate accurate transcribed text. By utilizing advanced speech recognition technology, it ensures efficient conversion of audio into text,sign language making it suitable for applications like note-taking, accessibility tools, and real-time communication.
- **Transcription Renderer**: This feature presents the transcribed text alongside a GIF representation of sign language directly on the user interface. By combining text and visual aids, it enhances accessibility and ensures effective communication for users with varying needs.



System Workflow Diagram

4 Implementation

The implementation involves:

4.1 Frontend: HTML and CSS

Defines the structure and styling of the user interface.

```
<main>
<div class="controls-container">
    <p>Click "Start Recording" and speak to see the sign language translation.</p>
    <button id="startButton" class="action-btn">Start Recording</button>
    <button id="stopButton" class="action-btn" disabled>Stop Recording</button>
</div>

<div class="transcript-container">
    <h3>Transcript:</h3>
    <p id="transcript">Your transcription will appear here...</p>
</div>

<div id="signLanguageContainer" class="sign-container">
    <!-- Single sign language video will appear here sequentially -->
</div>
</main>
```

```
body {
    font-family: Arial, sans-serif;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    min-height: 100vh;
    background: linear-gradient(135deg, #6a11cb 0%, #2575fc 100%);
    color: #333;
}

main {
    background-color: #fff;
    border-radius: 15px;
    padding: 30px;
    width: 90%;
    max-width: 600px;
    box-shadow: 0 6px 12px rgba(0, 0, 0, 0.3);
    text-align: center;
    animation: fadeInMain 1s ease-in-out;
}

/* Video Styling with Animation */
```

```
.sign-video {
    width: 500px; /* Increased size for better visibility */
    height: 300px;
    border-radius: 10px;
    box-shadow: 0 8px 16px rgba(0, 0, 0, 0.3);
    animation: fadeIn 0.5s forwards;
}
```

4.2 Backend: JavaScript

Implements real-time transcription using the Web Speech API.

```
// Function to display sign language videos for words and individual letters only if
async function displaySignLanguage(text) {
    // Clear any existing video in the container
    signLanguageContainer.innerHTML = '';

    // Split the text into unique words/numbers, removing punctuation
    const uniqueItems = Array.from(new Set(text.toLowerCase().replace(/[^a-z0-9\s]/g,

    for (const item of uniqueItems) {
        if (!item) continue;

        if (item.length === 1) {
            // Display video for single alphabet letters if they exist
            const letterVideoPath = `sign/${item}.mp4`;
            const letterSuccess = await displayVideo(letterVideoPath);
            if (!letterSuccess) {
                console.log(`Letter video for '${item}' not found.`);
            }
        } else {
            // Try displaying the full word video, skip if not found
            const itemVideoPath = `sign/${item}.mp4`;
            const wordSuccess = await displayVideo(itemVideoPath);
            if (!wordSuccess) {
                console.log(`Word video for '${item}' not found, skipping.`);
            }
        }
    }
}
```

4.3 UI

The user interface is designed to be intuitive, featuring clearly labeled buttons to start and stop recording, along with a dedicated display area for showcasing transcriptions in real-time. This setup ensures a seamless and user-friendly experience for capturing and viewing spoken content.

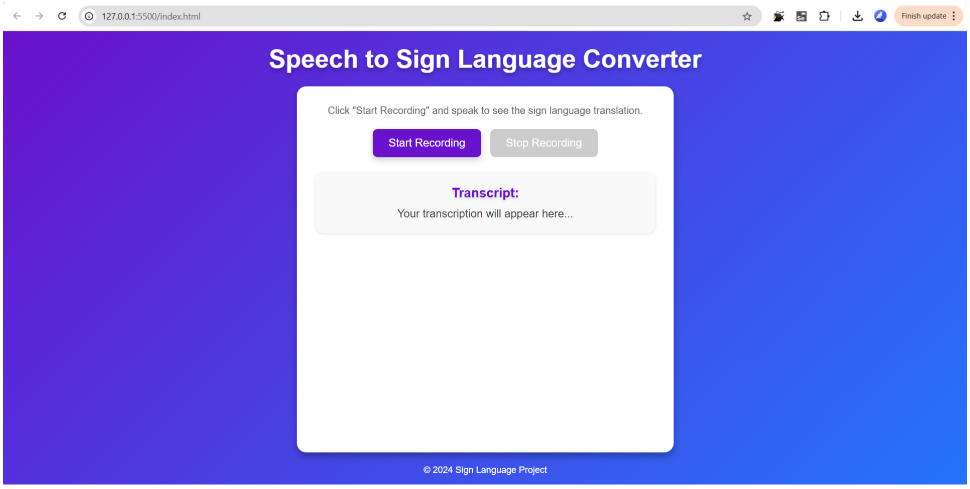


figure:Application User Interface

The figure below provides a detailed representation of the input data converted into sign language, showcasing the visual output designed to enhance accessibility and support effective communication.

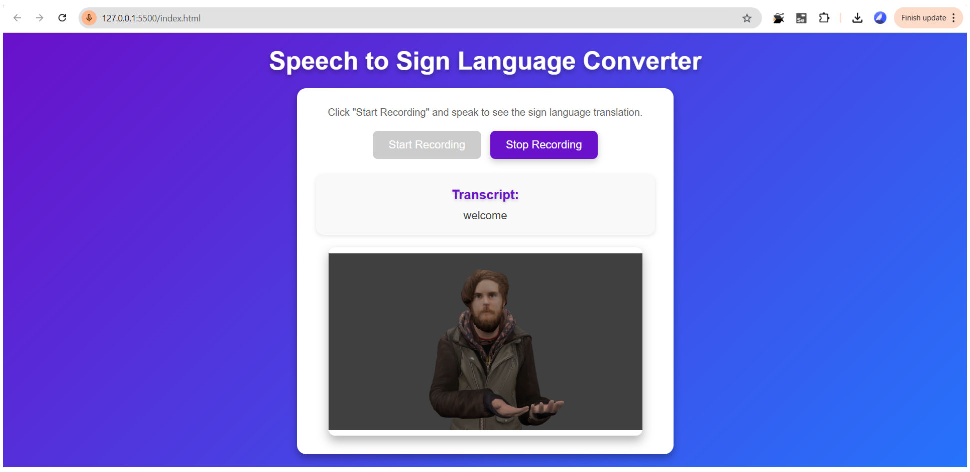


figure:Application User Interface

5 Evaluation and Testing

5.1 Evaluation

The system was evaluated based on:

- **Accuracy**: Measure the percentage of correctly transcribed words compared to the spoken input.
  Best in controlled environments, Good in noisy settings.
- **Latency**: Assess the UI's responsiveness by measuring load time and interaction delay.
- **User Satisfaction**: Conduct surveys to gauge user satisfaction with usability and performance.

5.2 Testing Scenarios

- **Clear Speech in Quiet Room**: Achieved high accuracy with minimal latency.
- **Speech in Noisy Environment**: Accuracy reduced due to background interference.
- **Cross-Browser Testing**: Limited functionality in non-Chromium-based browsers.

6 Challenges and Recommend Solutions

- **Browser Dependency**:
  Challenge: The platform shows compatibility issues in browsers other than Google Chrome, restricting its accessibility for a wider audience.
  Solution: Test and optimize the system for other popular browsers like Firefox, Safari, and Edge by incorporating cross-browser development tools and standards.
- **Noise Interference**:
  Challenge: Variability in speech accents, background noise, or unclear pronunciation can reduce the transcription accuracy.
  Solution: Integrate advanced noise-cancellation algorithms and train the system with diverse datasets to handle different accents and noisy environments.
- **Internet Dependency**:
  Challenge: The system requires an active internet connection to function, which limits usability in offline scenarios.
  Solution: Develop an offline mode using local speech recognition libraries or APIs like TensorFlow.js to provide basic functionality without internet access.
- **Limited Dataset for Speech-to-Sign Language**:
  Challenge: A lack of robust datasets reduces the system's ability to improve accuracy and versatility.
  Solution: Use publicly available datasets or create proprietary datasets by gathering diverse audio samples to train the recognition model.
- **User Adoption**:
  Challenge: Users unfamiliar with technology might find it challenging to use the platform effectively.
  Solution: Provide detailed user guides, tool tips, and video tutorials to simplify onboarding and improve the learning curve.

7 Future Scope

Proposed enhancements include:

- Multi-language transcription support enables the system to process and transcribe spoken content in multiple languages, catering to a diverse range of users. This feature is crucial in breaking down language barriers, allowing individuals to communicate effectively and access information in their native language.

- Offline functionality using WebAssembly (Wasm) allows the system to perform speech recognition and transcription tasks locally on a user's device without requiring an internet connection. WebAssembly is a binary instruction format designed for high-performance applications on web platforms, enabling lightweight and efficient execution of complex algorithms directly in the browser.

- Advanced noise filtering is a critical feature that enhances transcription accuracy in noisy environments, where background sounds can interfere with the recognition of spoken language. This functionality involves the integration of sophisticated audio processing algorithms that can isolate speech from ambient noise, ensuring that only relevant audio signals are processed for transcription and sign language conversion.

- Export options for transcription results in various formats, such as PDF, Video,audio and Word, provide users with flexibility and convenience in utilizing their transcriptions. This functionality allows users to save, share, and use the transcribed text across multiple platforms and applications, catering to diverse personal, educational, and professional needs.

8 Appendix

8.1 Pseudo code for Speech Recognition

```
START
Initialize Web Speech API
WHILE recording is active:
    Capture speech input
    Process input to generate transcription
    Display transcription in real-time
    Display recignized sign language according to transcription
END
```

8.2 Additional Testing Snippets

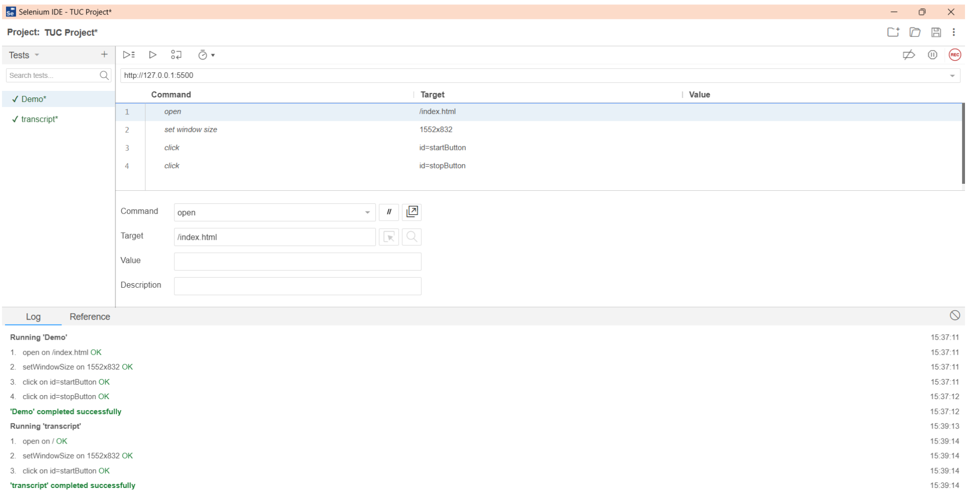


figure:Application User Interface Testing

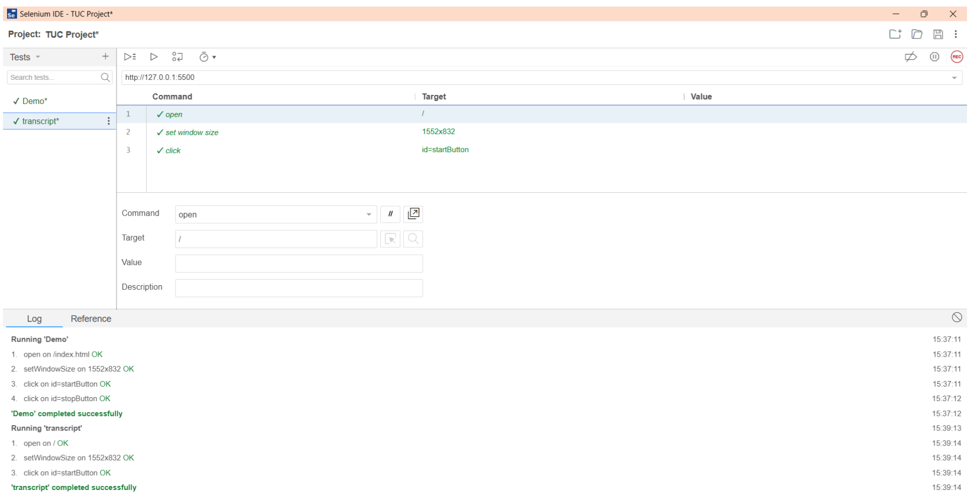


figure:Application User Interface Testing

9 References

- Web Speech API Documentation: https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API
- Automated Speech to Sign language Conversion using Google API and NLP https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3575439
- Audio to Sign Language Translation using NLP https://ieeexplore.ieee.org/document/10397050
- Conversation of Sign Language to Speech with Human Gestures https://www.sciencedirect.com/science/article/pii/S1877050915005050
- Assistive technology for the inclusion of students with disabilities: a systematic review https://link.springer.com/article/10.1007/s11423-022-10127-7
- Accessibility within open educational resources and practices for disabled learners: a systematic literature review. https://slejournal.springeropen.com/articles/10.1186/s40561-019-0113-2