# Text to Speech & Speech to Text Converter

## Project Report

Author: [Himanshu Bisht]

Date: [12/01/2025]

# Table of Contents

1. Introduction

2. Technology Stack

3. Features

4. Implementation

5. Speech-to-Text Details

6. Text-to-Speech Details

7. User Interface

8. Testing

9. Challenges Faced

10. Future Scope

11. Conclusion

12. References

## 1. Introduction

The purpose of this project is to provide a seamless interface for converting text to speech and speech to text using browser technologies. This tool facilitates accessibility for users with disabilities and enhances communication efficiency.

## 2. Technology Stack

Core Technologies Used:  
- HTML: Structure of the web application  
- CSS: Styling the application  
- JavaScript: Adding interactivity  
- APIs: SpeechRecognition and SpeechSynthesis  
  
APIs Utilized:  
- SpeechRecognition API  
- SpeechSynthesis API

## 3. Features

1. Text to Speech Conversion:  
- Input text and listen to it being read aloud  
- Supports multiple voice options  
2. Speech to Text Conversion:  
- Speak and transcribe speech into text  
- Provides language selection for accurate transcription.

## 4. Implementation

Workflow:  
1. User selects Text-to-Speech or Speech-to-Text.  
2. Relevant section is displayed.  
3. User interacts with input fields/buttons.

## 5. Speech-to-Text Details

Workflow:  
- Record voice and transcribe speech into text.  
- Supports multiple languages for transcription.  
- Handles background noise with configuration.

## 6. Text-to-Speech Details

Workflow:  
- Input text and convert it to speech.  
- Choose different voices from a dropdown menu.  
- Adjust language and tone as needed.

## 7. User Interface

- Clean and intuitive layout.  
- Responsive design for various devices.  
- Dropdowns for language and voice selection.

## 8. Testing

Test Cases:  
- Validate SpeechRecognition and SpeechSynthesis functionalities.  
- Test UI responsiveness.  
- Verify compatibility across multiple browsers.

## 9. Challenges Faced

Challenges:  
1. Browser Compatibility:  
- Solution: Provided fallback mechanisms.  
2. Background Noise:  
- Solution: Adjusted SpeechRecognition configurations.  
3. Voice Selection:  
- Solution: Dynamically updated available voices.

## 10. Future Scope

- Enhance voice recognition accuracy using AI.  
- Add support for more languages and dialects.  
- Integrate features for saving and sharing transcripts.  
- Provide speed adjustment for Text-to-Speech.

## 11. Conclusion

This project demonstrates effective use of web technologies for text-to-speech and speech-to-text functionalities. It addresses accessibility and interaction needs effectively.

## 12. References

- Mozilla Developer Network: SpeechRecognition API  
- Mozilla Developer Network: SpeechSynthesis API  
- Youtube: JavaScript Basics  
- Stack Overflow: Troubleshooting SpeechRecognition  
- Official Browser API Documentation