Cloud Computing Lab Mini Project Report

Audio To Text Conversion

Group No: 32

1604031	Juhi Gianani
1604032	Priya Gianchandani
1604051	Ankita Kar

1. Problem Definition

In modern civilized societies for communication between human, speech is one of the common methods. Different ideas formed in the mind of the speaker are communicated by speech in the form of words, phrases, and sentences by applying some proper grammatical rules. The speech is primary mode of communication among human being and also the most natural and efficient form of exchanging information among human in speech. Speech to Text conversion take input from microphone in the form of speech & then it is converted into text form which is display on desktop. Speech processing is the study of speech signals, and the various methods which are used to process them. In this process various applications such as speech coding, speech synthesis, speech recognition and speaker recognition technologies, speech processing is employed.

2. Requirements

This section gives detailed information about the functional requirements, non functional requirements, hardware requirements and software requirements of the project.

2.1 Functional Requirements

Functional Requirements specifies a function that a system component must be able to perform. It can be documented in various ways. Functional requirements maybe calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. The most common ones are written descriptions in documents and use cases. The functional requirements of this project are:

1. The system should be able to take input in the form of audio through microphone.

2. The system should be able to perform conversion of voice input into text.

2.2 Non Functional Requirements

Non-functional requirements specify the criteria used to judge the operation of the system, rather than the specific behaviors. Non-functional requirements are a description of features, characteristics and attributes of the system as well as any constraint that may limit the boundaries of the proposed system. The Non-Functional requirements are essentially based on the performance, information, economy, control and security efficiency and services. Based on this, the non-functional requirements are as follows:

1. **Performance Requirement: -** The system should be able to convert spoken words to text quickly. For desired performance, fetching audio and processing it, converting it to text, response time, processing speed of the system must be considered.
2. **Reliability: -** The solution should provide reliability to the user that the system will run all the features mentioned in this document are perfectly available and executing perfectly. It should be tested and debugged completely. All exceptions should be well handled.
3. **Accuracy: -** The solution should be able to reach the desired level of accuracy.
4. **Usability: -** Requirements about how difficult it will be to learn and operate the system. The requirements are often expressed in learning time or similar metrics.
5. **Portability: -** The effort required to move the software to a different target platform. The measurement is most commonly person-months or % of modules that need changing.

2.3 Hardware Requirements

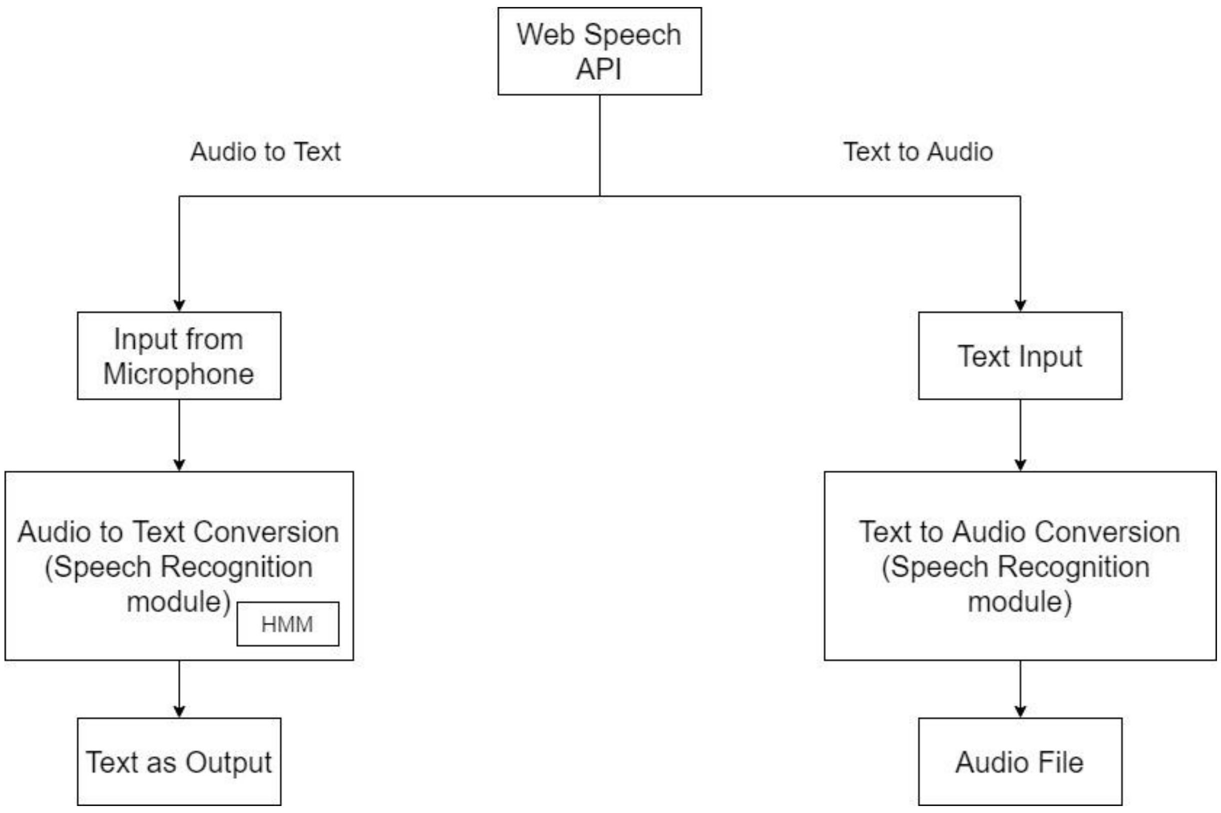
The audio input to the system will be given through a microphone. The microphone should be of good quality which can fetch the audio input quickly. It should have a minimum frequency of 20 Hz and Sound to Noise ratio should be more than 50. This project is implemented as a website, therefore it is independent of the device in which it is being run. The website cam run in desktops as well as in mobiles. The hardware requirement will be same in all the devices.

2.4 Software Requirements

The converted sign language will be shown on a screen. Since this project can be deployed as a website, a mobile application or a desktop application, the software requirement for each deployment method will be different. For the website, any type of browser can be used. For mobile application, Android version should be Lollipop (API Level 21) or higher. For desktop application, the operating system should be Microsoft Windows 7 or higher versions of Microsoft Windows.

3. Architecture

This project includes two-way conversion i.e. audio to text conversion as well as text to audio conversion. We have used Speech Recognition module for the conversions. The following is the block diagram which shows the flow of the process of both the conversions.


Web Speech API
Audio to Text
Text to Audio
Input from Microphone
Text Input
Audio to Text Conversion (Speech Recognition module)
HMM
Text to Audio Conversion (Speech Recognition module)
Text as Output
Audio File

1. **Audio to Text Conversion:**

   The audio input is taken from the user through the microphone. The Speech Recognition module does everything internally. The audio is captured, processed for noise removal and then converted to text. This module internally used Hidden Markov Model (HMM) algorithm for converting audio to text. The converted text is shown on the screen.

2. **Text to Audio Conversion**

   The text input is taken from the user. The process of converting text to audio is reverse of that of audio to text. Using Speech Recognition module, the text is recognized and converted to audio. The final audio file is displayed on the screen which is playable.

4. Services Provided by Cloud for Titled Project

The website is hosted on Firebase, a cloud platform providing many services for backend and hosting. Our project is using Software as a Service (SaaS) of cloud, since this website can be accessed from anywhere at any time.

5. Main Code

```
try {
varSpeechRecognition = window.SpeechRecognition ||
window.webkitSpeechRecognition;
var recognition = new SpeechRecognition();
}
catch(e) {
  console.error(e);
  $('.no-browser-support').show();
  $('.app').hide();
```

```
}

var noteTextarea = $('#note-textarea');
var instructions = $('#recording-instructions');
var notesList = $('ul#notes');
var noteContent = '';

// Get all notes from previous sessions and display them.
var notes = getAllNotes();
renderNotes(notes);

/*-----------------------------
      Voice Recognition
------------------------------*/

// If false, the recording will stop after a few seconds of silence.
// When true, the silence period is longer (about 15 seconds),
// allowing us to keep recording even when the user pauses.
recognition.continuous = true;

// This block is called every time the Speech APi captures a line.
recognition.onresult = function(event) {

  // event is a SpeechRecognitionEvent object.
  // It holds all the lines we have captured so far.
  // We only need the current one.
  var current = event.resultIndex;

  // Get a transcript of what was said.
  var transcript = event.results[current][0].transcript;

  // Add the current transcript to the contents of our Note.
  // There is a weird bug on mobile, where everything is repeated twice.
  // There is no official solution so far so we have to handle an edge case.
  var mobileRepeatBug = (current == 1 && transcript ==
event.results[0][0].transcript);

  if(!mobileRepeatBug) {
    noteContent += transcript;
    noteTextarea.val(noteContent);
  }
};

recognition.onstart = function() {
  instructions.text('Voice recognition activated. Try speaking into the
microphone.');
}
recognition.onspeechend = function() {
  instructions.text('You were quiet for a while so voice recognition turned itself
off.');
}
recognition.onerror = function(event) {
  if(event.error == 'no-speech') {
    instructions.text('No speech was detected. Try again.');
  };
}

/*-----------------------------
      App buttons and input
------------------------------*/

$('#start-record-btn').on('click', function(e) {
  if (noteContent.length) {
    noteContent += ' ';
  }
  recognition.start();
});

$('#pause-record-btn').on('click', function(e) {
  recognition.stop();
  instructions.text('Voice recognition paused.');
});

// Sync the text inside the text area with the noteContent variable.
noteTextarea.on('input', function() {
  noteContent = $(this).val();
})
```

```
$('#save-note-btn').on('click', function(e) {
  recognition.stop();
  if(!noteContent.length) {
    instructions.text('Could not save empty note. Please add a message to your
note.');
  }
else {
    // Save note to localStorage.
    // The key is the dateTime with seconds, the value is the content of the note.
    saveNote(new Date().toLocaleString(), noteContent);

    // Reset variables and update UI.
    noteContent = '';
    renderNotes(getAllNotes());
    noteTextarea.val('');
    instructions.text('Note saved successfully.');
  }

})

notesList.on('click', function(e) {
  e.preventDefault();
  var target = $(e.target);

  // Listen to the selected note.
  if(target.hasClass('listen-note')) {
    var content = target.closest('.note').find('.content').text();
    readOutLoud(content);
  }

  // Delete note.
  if(target.hasClass('delete-note')) {
    var dateTime = target.siblings('.date').text();
    deleteNote(dateTime);
    target.closest('.note').remove();
  }
});

/*-----------------------------
```

```
    Speech Synthesis
------------------------------*/

function readOutLoud(message) {
        var speech = new SpeechSynthesisUtterance();

  // Set the text and voice attributes.
        speech.text = message;
        speech.volume = 1;
        speech.rate = 1;
        speech.pitch = 1;

        window.speechSynthesis.speak(speech);
}

/*-----------------------------
    Helper Functions
------------------------------*/

function renderNotes(notes) {
  var html = '';
  if(notes.length) {
    notes.forEach(function(note) {
      html+= `<li class="note">
        <p class="header">
          <span class="date">${note.date}</span>
          <a href="#" class="listen-note" title="Listen to Note">Listen to 
Note</a>
          <a href="#" class="delete-note" title="Delete">Delete</a>
        </p>
        <p class="content">${note.content}</p>
      </li>`;
    });
  }
  else {
    html = '<li><p class="content">You don\'t have any notes yet.</p></li>';
  }
  notesList.html(html);
}
```

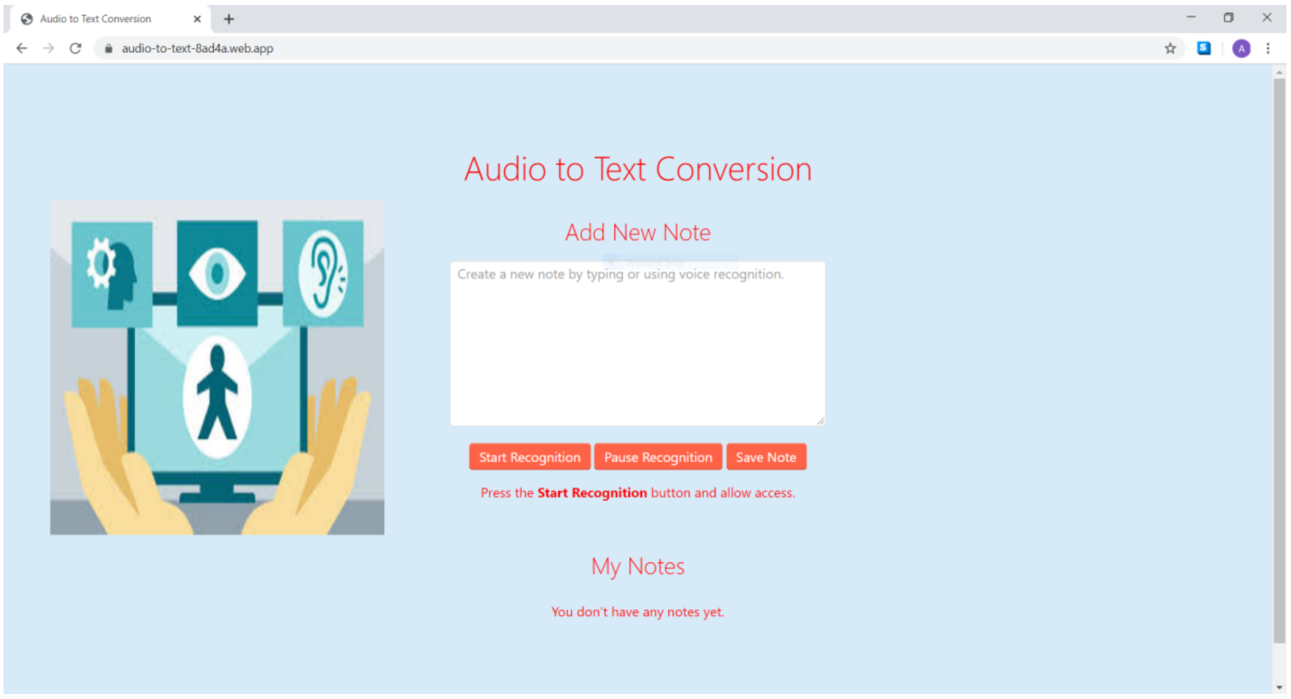
```
function saveNote(dateTime, content) {
  localStorage.setItem('note-' + dateTime, content);
}

function getAllNotes() {
  var notes = [];
  var key;
  for (var i = 0; i < localStorage.length; i++) {
    key = localStorage.key(i);

    if(key.substring(0,5) == 'note-') {
      notes.push({
        date: key.replace('note-',''),
        content: localStorage.getItem(localStorage.key(i))
      });
    }
  }
  return notes;
}

function deleteNote(dateTime) {
  localStorage.removeItem('note-' + dateTime);
}
```

6. Screenshots of Project

This section contains the screenshots of our project. For Audio to Text conversion, the user has to give audio input after clicking Start Recognition button. After some seconds, the text output will be displayed on the screen. The user can also save the text as a note, which will be saved as text as well as audio. For Text to Audio conversion, the user has to type the text input in the text area and click Save note. The audio file will be visible below which is playable. The link of the website is https://audio-to-text-8ad4a.firebaseapp.com/

1. Logo:

2. Landing Page:



Audio to Text Conversion

Add New Note

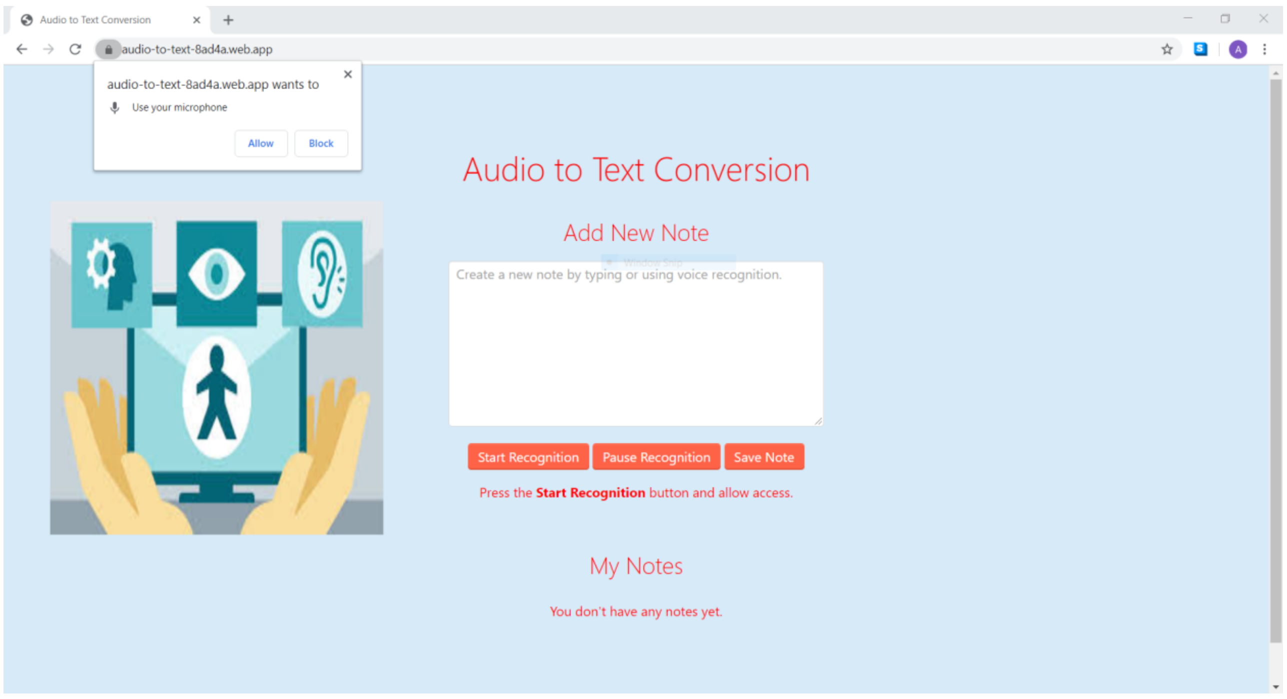
Create a new note by typing or using voice recognition.

Press the **Start Recognition** button and allow access.

My Notes

You don't have any notes yet.

3. Taking input through microphone:


Audio to Text Conversion

audio-to-text-8ad4a.web.app

audio-to-text-8ad4a.web.app wants to

Use your microphone

Allow Block

Audio to Text Conversion

Add New Note

Create a new note by typing or using voice recognition.

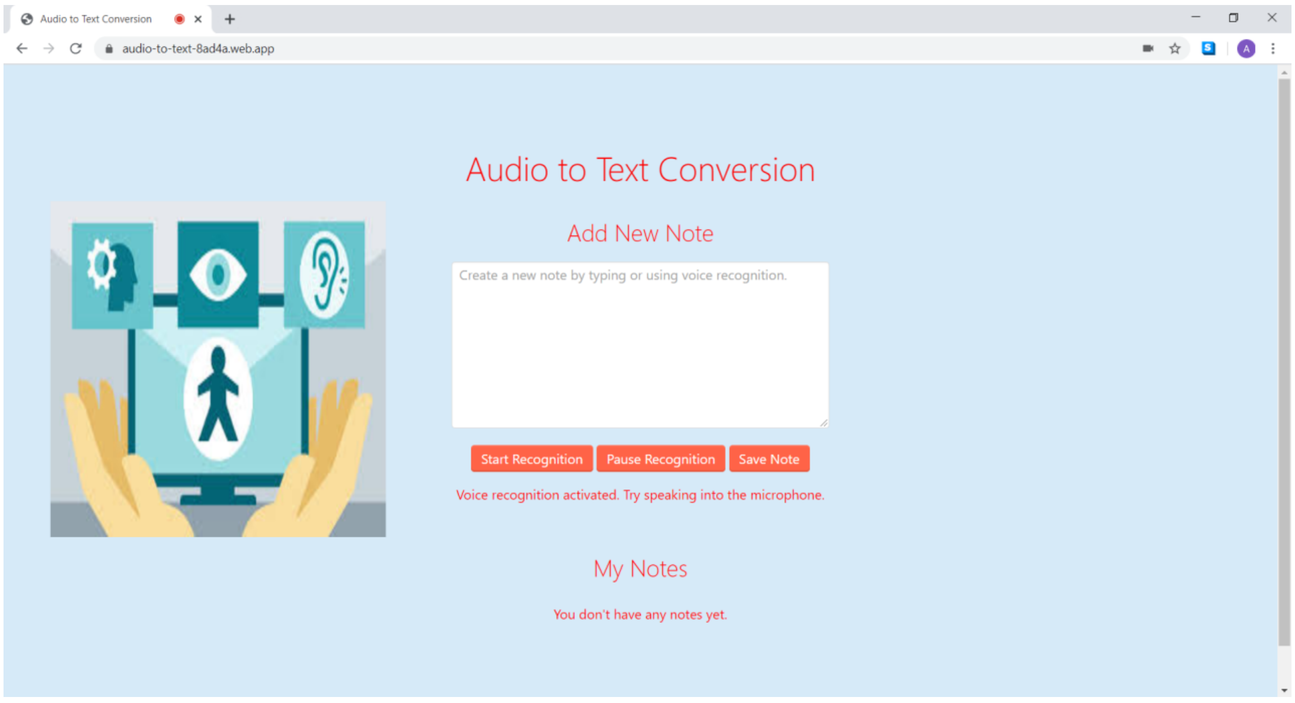
Start Recognition Pause Recognition Save Note

Press the **Start Recognition** button and allow access.

My Notes

You don't have any notes yet.

4. Audio getting converted to text:


Audio to Text Conversion

audio-to-text-8ad4a.web.app

Audio to Text Conversion

Add New Note

Create a new note by typing or using voice recognition.

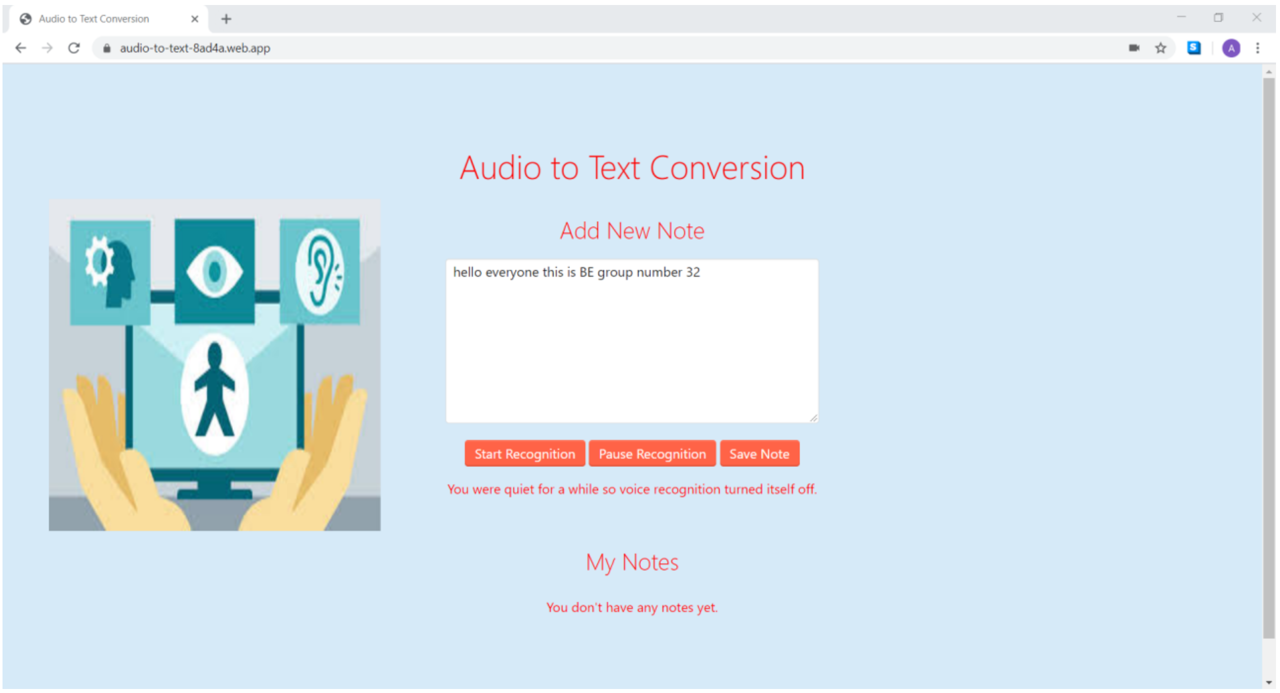
Start Recognition Pause Recognition Save Note

Voice recognition activated. Try speaking into the microphone.

My Notes

You don't have any notes yet.

5. Converted text:



Audio to Text Conversion

Add New Note

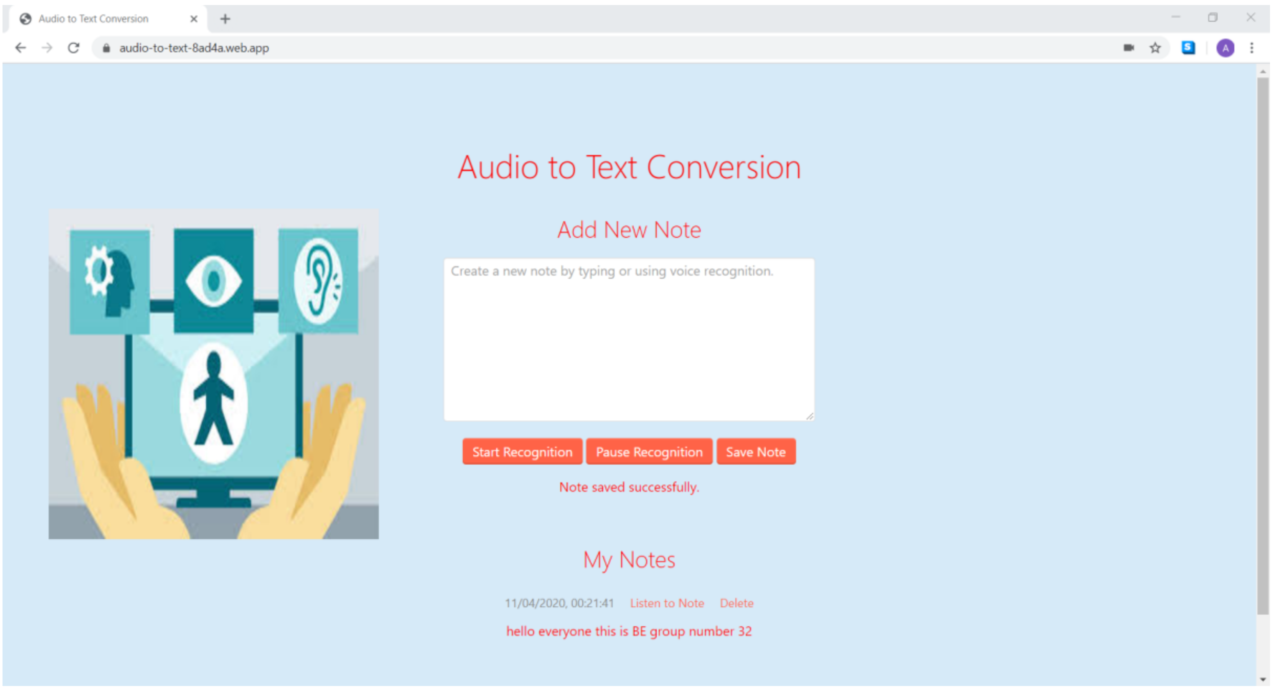
hello everyone this is BE group number 32

You were quiet for a while so voice recognition turned itself off.

My Notes

You don't have any notes yet.

6. Save note:



Audio to Text Conversion

Add New Note

Create a new note by typing or using voice recognition.

Note saved successfully.

My Notes

11/04/2020, 00:21:41 Listen to Note Delete

hello everyone this is BE group number 32

7. Conclusion

The Audio to Text Conversion project is implemented as a website and hosted on Firebase, a cloud platform providing various services for backend and hosting. The user gives the audio input through microphone which is converted to text using Hidden Markov Model (HMM) algorithm internally. The text is displayed on the screen which can also be saved as a note. There is also text to audio conversion service. The user has to give text input which will be converted to audio file which is playable on the website. The project is using Software as a Service (SaaS) of cloud, since the website can be accessed from anywhere and anytime.