Green University of Bangladesh
Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering

Semester: (Spring, Year: 2021), B.Sc. in CSE (Evening)

LAB PROJECT PROPOSAL

Course Title: Artificial Intelligence Lab

Course Code: CSE-404 Section: ED

Student Details

Name		ID
1.	Tanvir Ahmed	191015102
2.	Shahajady Khatun Juthy	191015105
3.	Sanzida Yesmin	191015123

Submission Date : 12/1/2021

Course Teacher's Name : Md. Moshiur Rahman

[For Teachers use only: Don't Write Anything inside this box]

Project Proposal Status

Marks: …… Signature: .....

Comments: ..... Date: .....

Project Name: Speech Recognition Web Application.

Introduction:

Speech recognition refers to the process of enabling a computer to identify and respond to the sounds produced in human speech.

It was first introduced at Bell Laboratories in 1952 and this version could only recognize numbers but not words. Few years later, speech recognition had grown from just recognizing numbers to recognizing text, grammars and even detecting noise.

This technology was developed as an alternative to typing on keyboard, we will only have to talk to our computer and our words appear on our computer screen.

In year 2012, the Web Speech API was introduced with the aim of enabling speech recognition and also converting text to speech on modern web browsers. Speech recognition is not currently supported on all browsers.

Project Explanation:

This is a Web Based Project, here we used JavaScript to represent artificial intelligence the things we can do in this project are:

a) Auto voice replay system
b) Voice to text convert
c) Text to voice convert

Used:

a) HTML5
b) CSS3
c) Bootstrap
d) JavaScript

Code Explanation:

The SpeechRecognition interface of the Web Speech API is the controller interface for the recognition service,this also handles the SpeechRecognitionEvent sent from the recognition service.

Constructor: SpeechRecognition.SpeechRecognition() Creates a new SpeechRecognition object.

Properties:
SpeechRecognition also inherits properties from its parent interface, EventTarget.

SpeechRecognition.grammars:
Returns and sets a collection of SpeechGrammar objects that represent the grammars that will be understood by the current SpeechRecognition.

SpeechRecognition.lang:
Returns and sets the language of the current SpeechRecognition. If not specified, this defaults to the HTML lang attribute value, or the user agent's language setting if that isn't set either.

SpeechRecognition.continuous:
Controls whether continuous results are returned for each recognition, or only a single result. Defaults to single (false.)

SpeechRecognition.interimResults:
Controls whether interim results should be returned (true) or not (false.) Interim results are results that are not yet final (the SpeechRecognitionResult.isFinal property is false.)

SpeechRecognition.maxAlternatives:
Sets the maximum number of SpeechRecognitionAlternatives provided per result. The default value is 1.

SpeechRecognition.serviceURI:
Specifies the location of the speech recognition service used by the current SpeechRecognition to handle the actual recognition. The default is the user agent's default speech service.

Method:
SpeechRecognition also inherits methods from its parent interface, EventTarget.

SpeechRecognition.abort():
Stops the speech recognition service from listening to incoming audio, and doesn't attempt to return a SpeechRecognitionResult.

SpeechRecognition.start():
Starts the speech recognition service listening to incoming audio with intent to recognize grammars associated with the current SpeechRecognition.

SpeechRecognition.stop():
Stops the speech recognition service from listening to incoming audio, and attempts to return a SpeechRecognitionResult using the audio captured so far.

Plugging the grammar into our speech recognition:
The next thing to do is define a speech recogntion instance to control the recognition for our application. This is done using the SpeechRecognition() constructor. We also create a new speech grammar list to contain our grammar, using the SpeechGrammarList() constructor.

Events:
Listen to these events using addEventListener() or by assigning an event listener to the oneventname property of this interface.

Audiostart:
Fired when the user agent has started to capture audio. Also available via the onaudiostart property.

Audioend:
Fired when the user agent has finished capturing audio. Also available via the onaudioend property.

End:
Fired when the speech recognition service has disconnected. Also available via the onend property.

Error:
Fired when a speech recognition error occurs. Also available via the onerror property.

Nomatch:
Fired when the speech recognition service returns a final result with no significant recognition. This may involve some degree of recognition,

which doesn't meet or exceed the confidence threshold. Also available via the onnomatch property.

Result:
Fired when the speech recognition service returns a result — a word or phrase has been positively recognized and this has been communicated back to the app. Also available via the onresult property.

Soundstart:
Fired when any sound — recognisable speech or not — has been detected. Also available via the onsoundstart property.

Soundend:
Fired when any sound — recognisable speech or not — has stopped being detected. Also available via the onsoundend property.

Speechstart:
Fired when sound that is recognized by the speech recognition service as speech has been detected. Also available via the onspeechstart property.

Speechend:
Fired when speech recognized by the speech recognition service has stopped being detected. Also available via the onspeechend property.

Start:
Fired when the speech recognition service has begun listening to incoming audio with intent to recognize grammars associated with the current SpeechRecognition. Also available via the onstart property.

PROBLEM DOMAIN & MOTIVATIONS:

a. The speech recognition part of the Web Speech API allows authorized Web applications to access the device's microphone and produces a transcript of the voice being recorded. This allows Web applications to use voice as one of the input & control method, similar to touch or keyboard.
b. It is a JavaScript API that enables web developers to incorporate speech recognition and synthesis into their web pages. It enables developers to use scripting to generate text-to-speech output and to use speech recognition as an input for forms, continuous dictation, and control.

OBJECTIVES/AIMS:

a. Speech Recognition is the process of receiving a voice through a microphone and making it enable a computer to identify and respond, Thus allowing for further actions to be initiated as a result.
b. Speech synthesis is an artificial simulation of human speech with a computer.
c. The speech recognition object can either stop listening after the user stops speaking or it can keep listening until the user stops it. If we only want to recognize a phrase or a word, we can set this to false. For this tutorial, let’s set it to true.

Browser support:

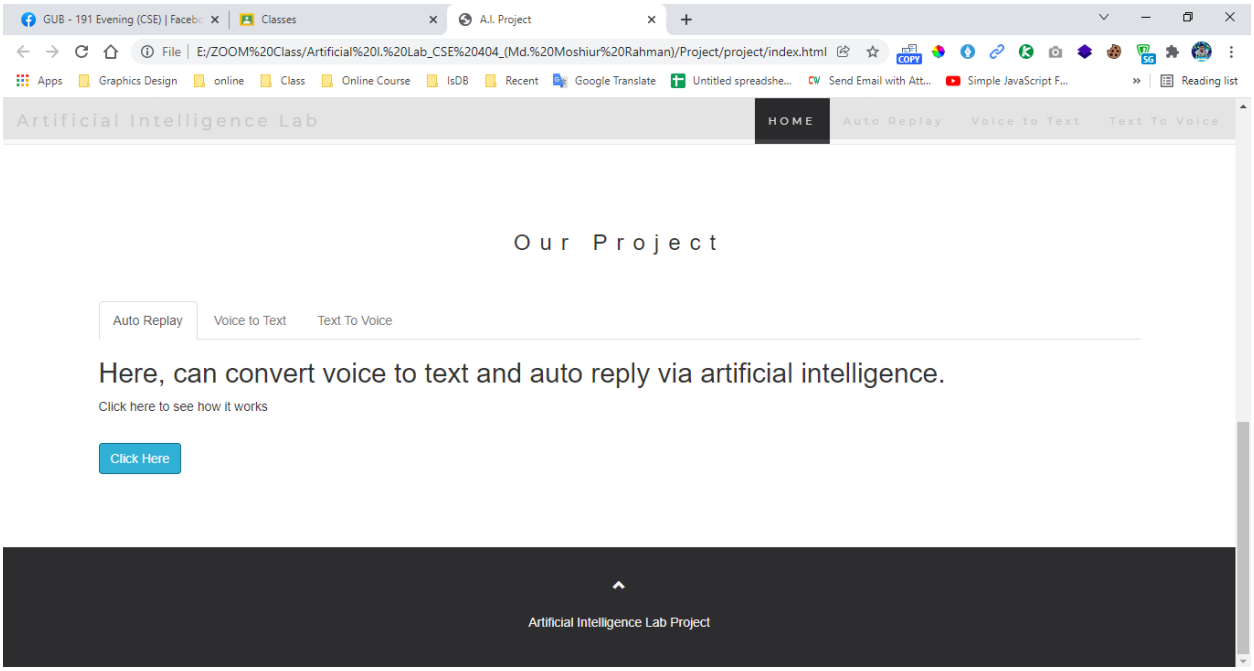
Support for Web Speech API speech recognition is currently limited to Chrome for Desktop and Android — Chrome has supported it since around version 33 but with prefixed interfaces, so you need to include prefixed versions of them, webkitSpeechRecognition.

HTML and CSS:

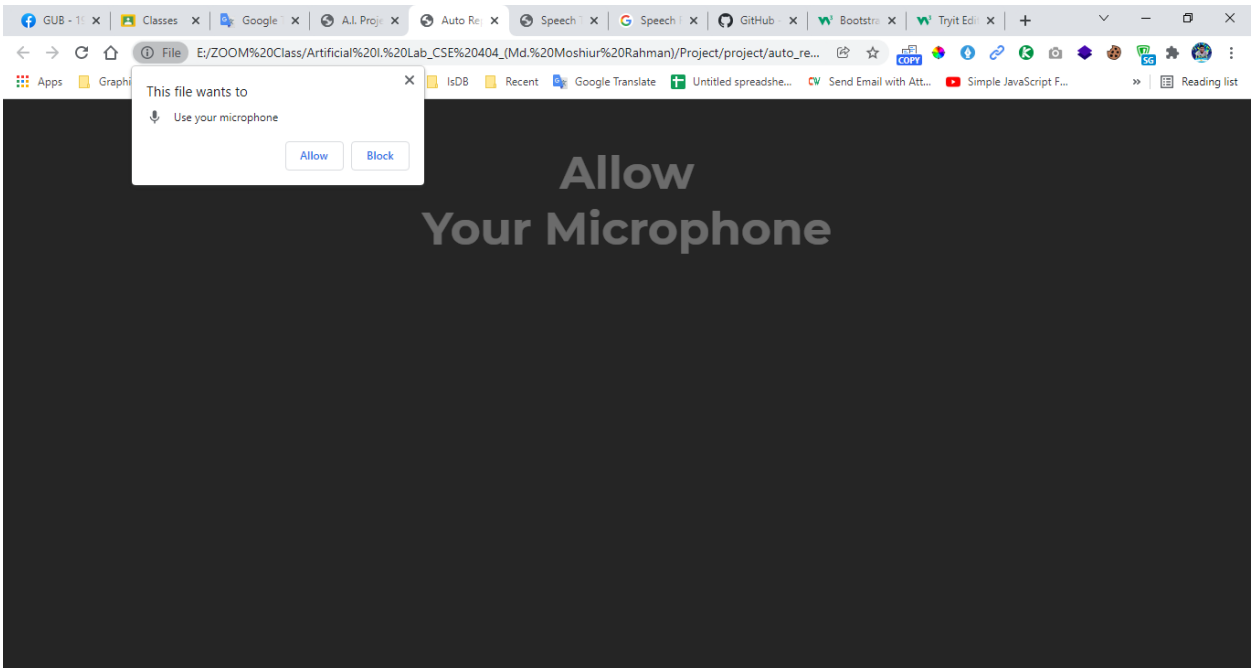
The HTML and CSS for the app is really trivial. We have a title, instructions paragraph, and a div into which we output diagnostic messages.

Output:

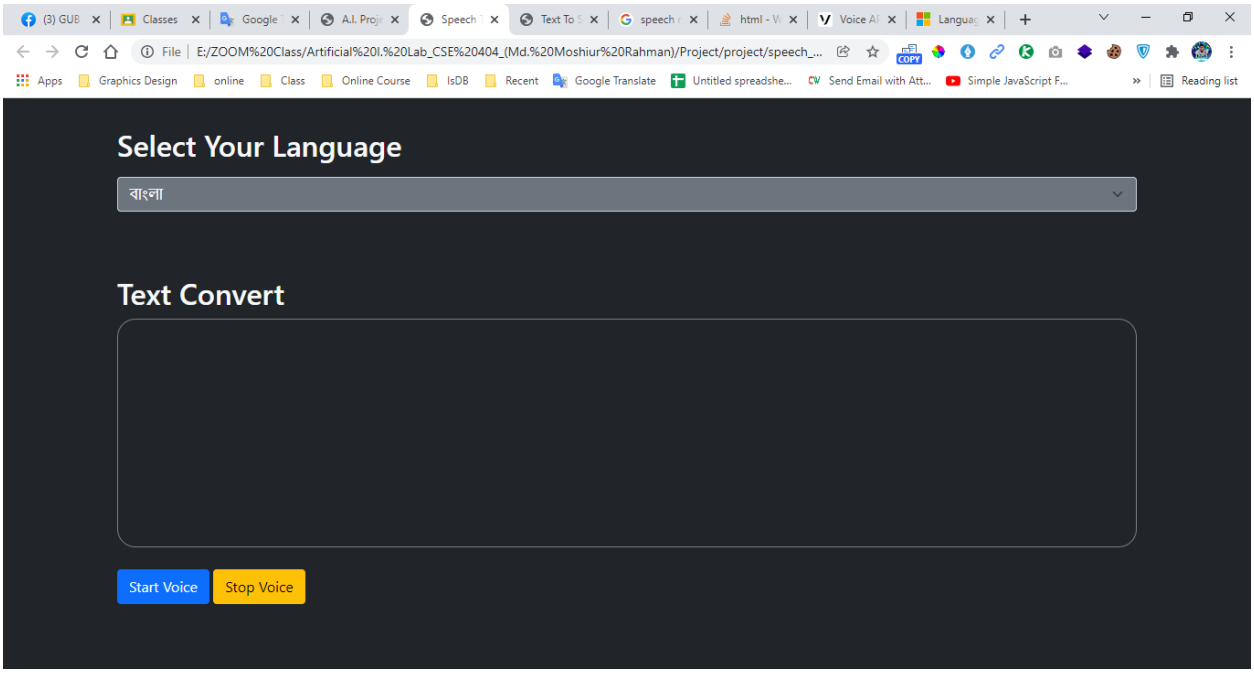
Project Display



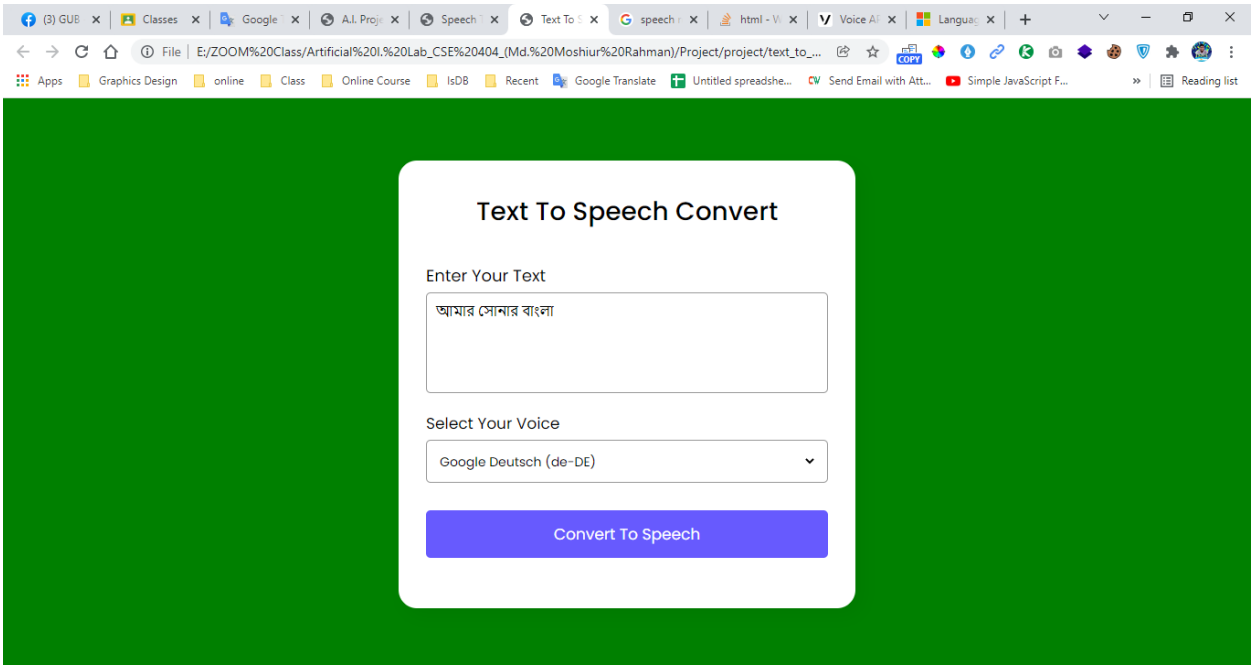
Auto Replay:



Voice to Text Convert:



Text to Voice Convert:



1. CONCLUSION:

a. Speech recognition refers to the process of enabling a computer to identify and respond to the sounds produced in human speech.
b. If someone's hand is disabled, they will be able to communicate through voice here.
c. Both of these APIs play a great role in accessibility over the past few years, most especially for the visually impaired, people with an injured arm, and many more.

Thank You