REAL TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

A PROJECT REPORT

SUBMITTED BY

TEAM ID: PNT2022TMID29406

TEAM MEMBERS	REGISTER NUMBER
YAMUNA M	422519205047
ARUNESHWARI S	422519205004
KAVIYA B	422519205019
NANDHINI A	422519205027
ROSHINI A	422519205036

INDUSTRY MENTOR(S) : DIVYA

FACULTY MENTOR(S) : KAVITHA E

In partial fulfilment for the award of

BACHELOR OF TECHNOLOGY

INFORMATION TECHNOLOGY

UNIVERSITY COLLEGE OF ENGINEERING VILLUPURAM

Table Of Contents

12. FUTURE SCOPE

13. APPENDIX

13.1 Source Code

13.2 GitHub & Project Demo Link

REAL TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

1. INTRODUCTION

1.1 PROJECT OVERVIEW

People with disabilities exist in our society. Although technology is constantly evolving, little is being done to improve the lives of these people. It has always been difficult to communicate with someone who is deaf and mute. It is quite challenging for silent people to communicate with non-mute people. because hand-sign language is not taught to the general public. It might be quite challenging for them to communicate at times of crisis. In circumstances where other modes of communication, like speech, are not possible, the human hand has remained a common alternative for information transmission. To have proper communication between a normal person and a handicapped person in any language, a voice conversion system with hand gesture recognition and translation will be very helpful.

1.2 PURPOSE

The project intends to create a system that can translate speech into acceptable sign language for the deaf and dumb, as well as translate sign language into a human voice in the desired language to communicate a message to normal people. A convolutional neural network is being used to build a model that is trained on various hand motions. On the basis of this model, an app is created. With the help of this app, people who are deaf or dumb can communicate using signs that are translated into speech and human-understandable words.

2. LITERATURE SURVEY

2.1 Existing Problem

There are handicapped people in our society. Although technology is constantly evolving, little is being done to improve the lives of these people. It has always been difficult to communicate with

someone who is deaf-mute. It is challenging for mute persons to communicate with hearing people. because hand sign language is not taught to the general public. It might be quite challenging for them to communicate at times of crisis. In circumstances where other forms of communication, like speech, are not possible, the human hand has remained a common choice for information transmission. To have a proper conversation between a normal person and an impaired person in any language, a Voice Conversion System with Hand Gesture Recognition and Translation will come in handy.

2.2 References

- Design of Communication Interpreter for Deaf and Dumb Person was published by Pallavi Verma (Electrical and Electronics Department, Amity University, Greater Noida, Uttar Pradesh, India), Shimi S. L (Assistant Professor, NITTTR, Chandigarh, India), Richa Priyadarshani (Electrical and Electronics Department, Amity University, Greater Noida, Uttar Pradesh, India).

- International Journal of Science and Research (IJSR) · Jan 2013 Development of full duplex intelligent communication system for deaf and dumb people was published in the year January 2017 DOI:10.1109/CONFLUENCE.2017.7943247.

- At 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence (Confluence) by Surbhi Rathi Department of Information Technology, Yeshwantrao Chavan College of Engineering Nagpur, India and Ujwalla Gawande, Department of Information Technology Yeshwantrao Chavan College of Engineering Nagpur, India.

- A Review Paper on Sign Language Recognition for The Deaf and Dumb published by R Rumana(B.E Graduate(IV year), Department of Computer Science and Engineering, SCSVMV, Kanchipuram) , Reddygari Sandhya Rani(B.E Graduate(IV year), Department of Computer Science and Engineering, SCSVMV, Kanchipuram) , Mrs. R. Prema(Assistant Professor, Department of Computer Science and Engineering, SCSVMV, Kanchipuram).

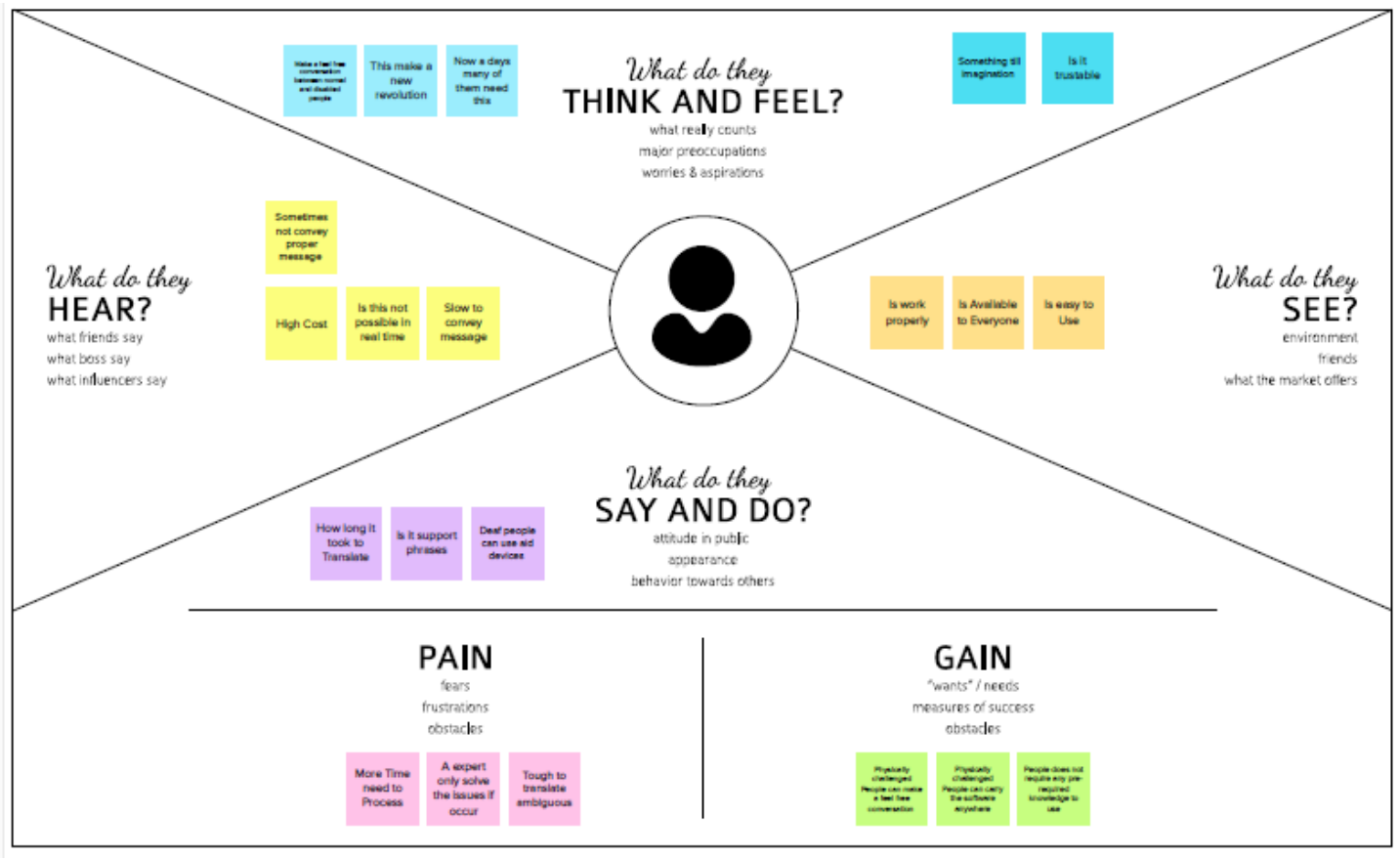
2.3 PROBLEM STATEMENT DEFINITION

This venture is to plan and actualise a framework that can make an interpretation of finger spelling to discourse and the other way around by utilizing acknowledgement and combination methods for every methodology. Such a framework will empower correspondence with the consultation impeded when no other methodology is accessible. Albeit gesture based communication is the fundamental correspondence mode of the consultation weekend, as far as programmed acknowledgment, finger spelling has the benefit of utilizing predetermined number of finder signs, comparing to the letters/sounds in the letters in order.

3. IDEATION AND PROPOSED SOLUTION

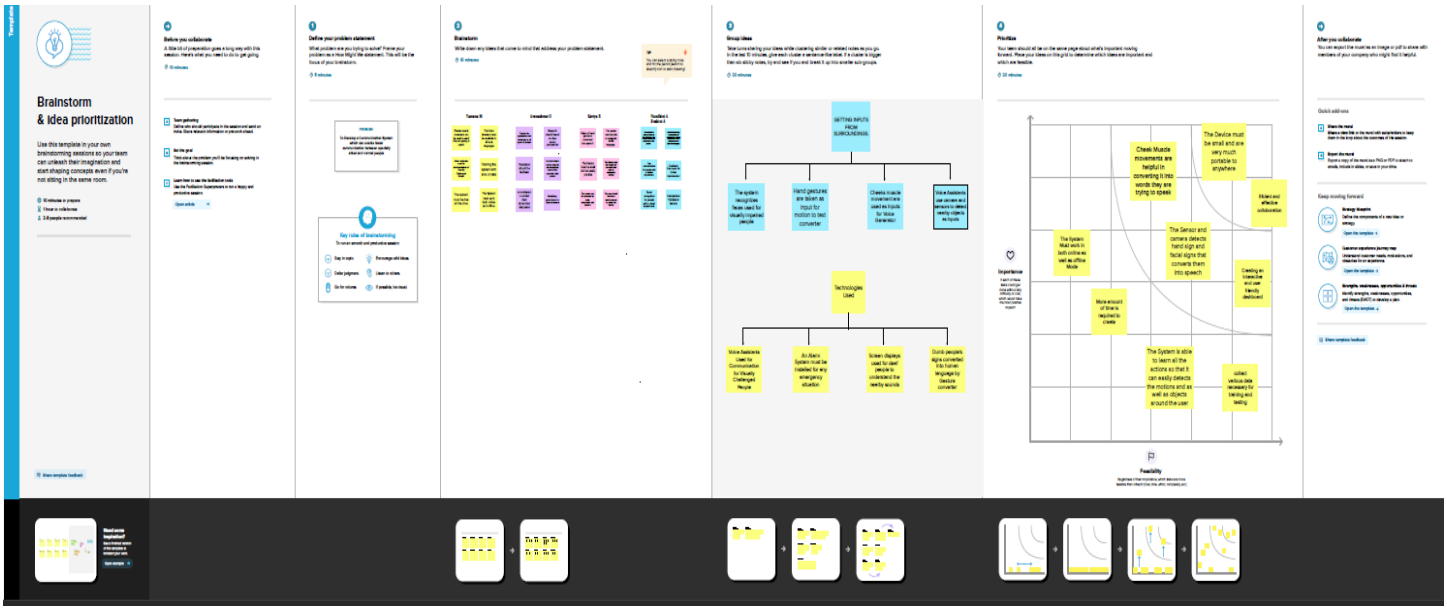
3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visuals that captures knowledge about an user's behaviour and attitude. It is an useful tool to help team build a better understanding for their users. Creating an effective solution requires understanding the true problem he person who is experiencing it.



3.2 IDEATION AND BRAINSTORMING

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.



3.3 PROPOSED SOLUTION

Problem Statement (Problem to be solved):

This programme can facilitate communication between those who have special needs and those
who do not, as well as the other way around.

Idea / Solution description:

By employing AI technology to import the relevant libraries into the Python code, these problems can be fixed. This research presents a prototype assistive system for Deaf-mute people to bridge the communication gap with hearing people. With the help of this tool, a person can communicate through hand gestures to recognise a variety of gesture-based signs. The controller for this assistive system was developed to analyse gesture images using a range of image processing methods and deep learning models to recognise the sign. This sign is converted into voice in real time using a text-to-speech module.

Novelty / Uniqueness:

This project was developed to address the need to translate a variety of modalities, such as images into Natural Language (NL) text, into a language that deaf and blind people can understand and communicate in. The prototype created as a result of this research includes cameras attached to dark-coloured spectacles, along with a travel-sized computer, speaker, and microphone. However, all of the technologies we looked into up until this point were only focused on one parameter or degree of disability among the three: blindness, deafness, and dumbness. We have since discovered a variety of technologies that can help people with disabilities communicate more easily among themselves and with the rest of society. There is currently no technology that is sufficiently developed to act as a general solution to any of these three constraints. Therefore, in order to accomplish this, we propose a generic strategy that anyone with a combination of these three disabilities might use to picture oneself as a part of this magnificent setting.

Social Impact / Customer Satisfaction:

Due to society's poor command of ASL and lack of awareness of the Deaf community, Deaf individuals face challenges in many facets of everyday life, including employment, higher education, healthcare, mental health services, emergency preparedness, technology, and government benefits. The development of expressive and receptive communication skills (speech and language) is hampered. Academic achievement is lowered by language-related learning difficulties. Communication issues usually lead to social isolation and a poor self image.

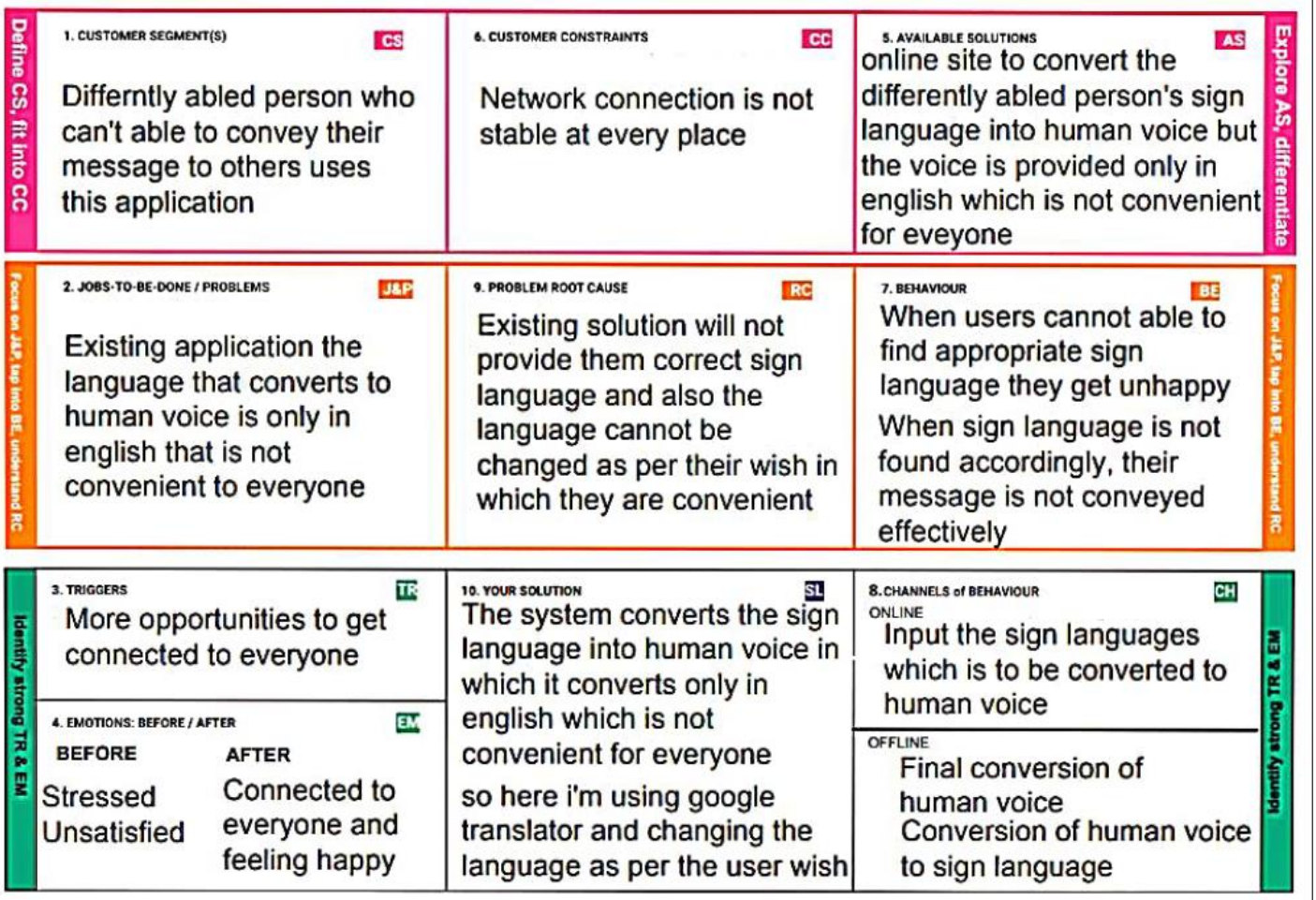
Business Model (Revenue Model):

A business must continuously keep its clients in mind in an environment where the pace is accelerating if it wants to keep their satisfaction and, consequently, their loyalty. Offering ICT based services to people with intellectual disabilities is the case company's main focus. The case firm has concluded that in order to maintain and grow its revenue, it is essential to improve the customer experience because it has already greatly increased its market share and expanded its operations in its primary client categories. The challenge the company is now facing is to develop a business plan and continue offering customers satisfactory service in order to maintain and boost the company's returns.

Scalability of the Solution:

This is typically accepted in the Deaf community, but hearing people might not comprehend it. Directness in communication is regarded to be one aspect of the Deaf culture. Two more indications for communication are touch and physical proximity.

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registrationthrough Gmail
FR-2	User Confirmation	Confirmation via Email Confirmationvia OTP.
FR-3	Create connection	Create a connection between users
FR-4	Access the input	Get the user input through webcam/camera/voice
FR-5	Conversion	Convert the input to voice and voice to sign language

4.2 NON-FUNCTIONAL REQUIREMENTS:

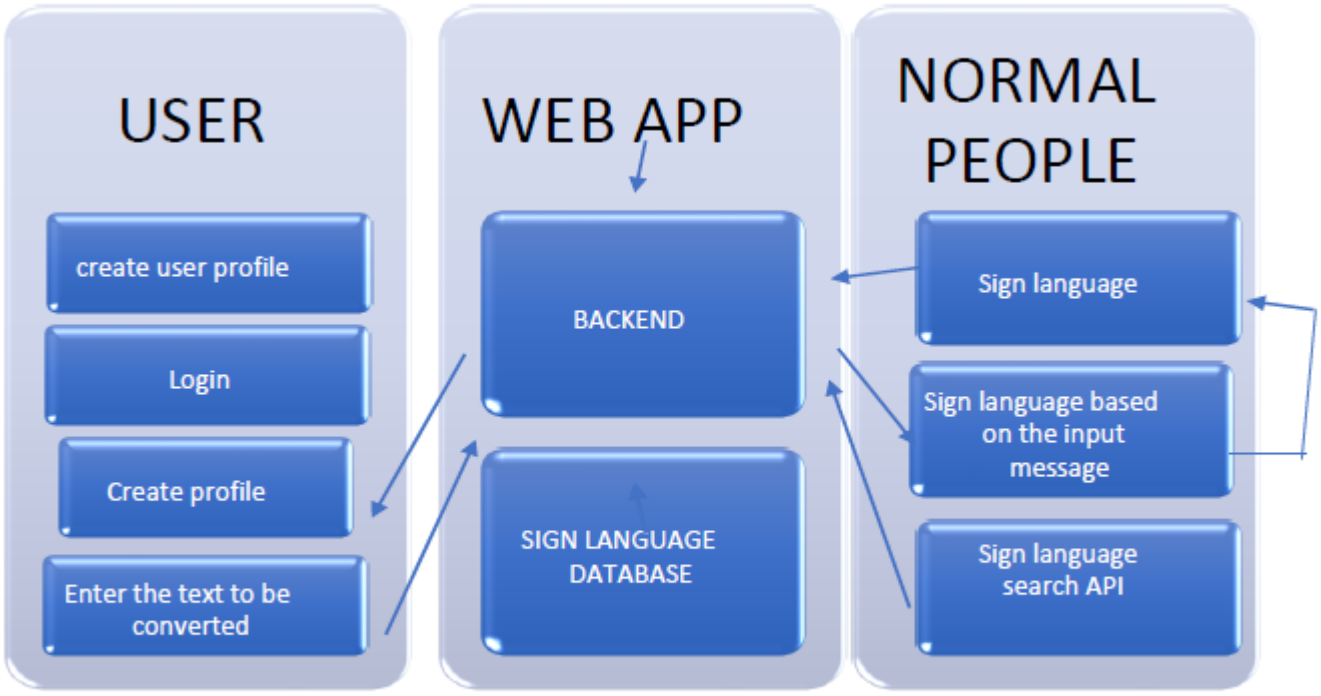
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Making a UI user-friendly creates an easy approach to comprehend it.
NFR-2	**Security**	Making the connection between the users without any interference from the external user.
NFR-3	**Reliability**	The application must run smoothly and without errors.
NFR-4	**Performance**	The speed at which a website's pages load and appear in a web browser is referred to as website performance.
NFR-5	**Availability**	The ability of users to access and use a website or web service is referred to as website availability (also known as website uptime). Usually, a website's availability is expressed as a percentage for a specific period of time.

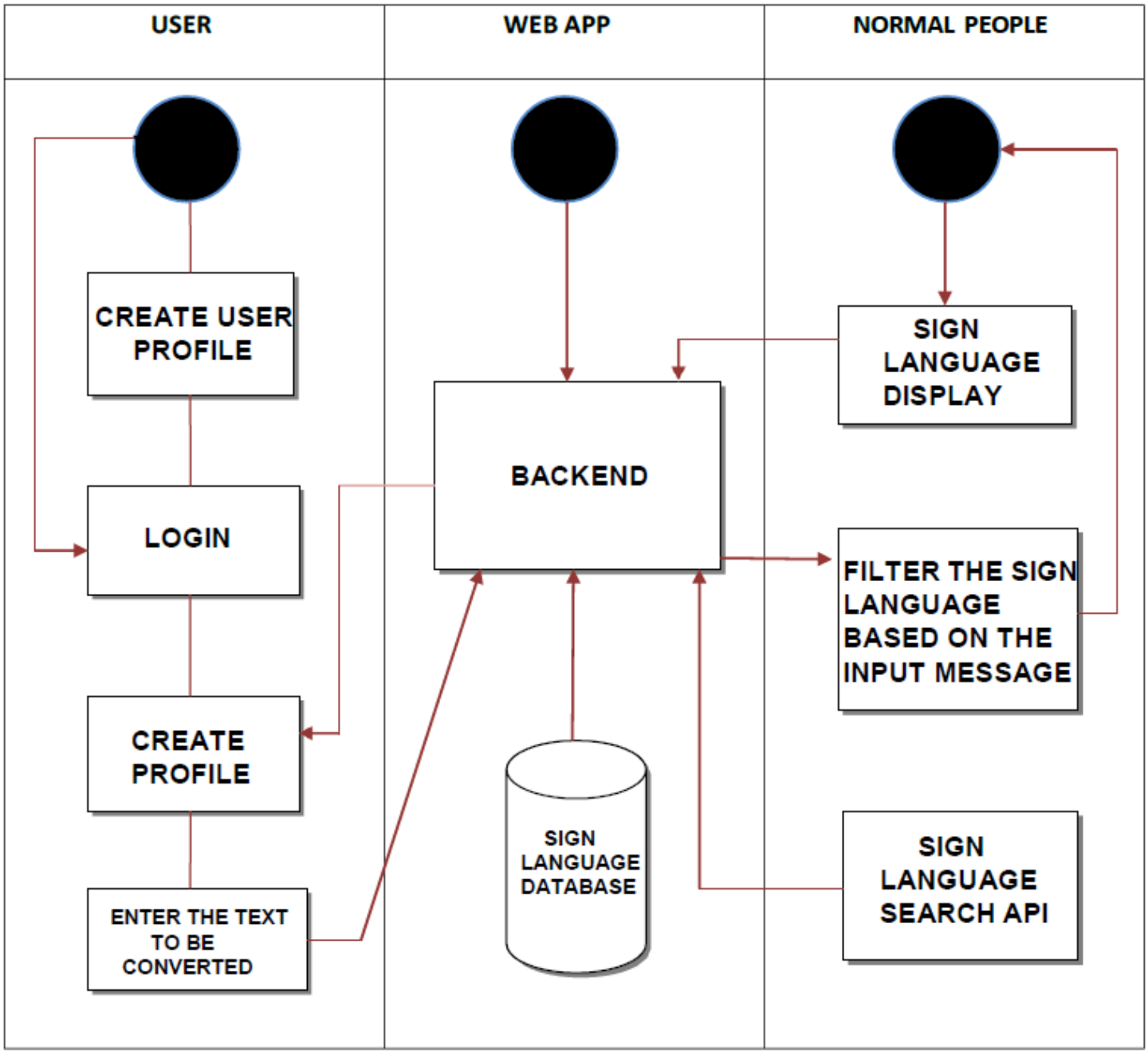
5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Low vision)	Registration	USN-1	As a user, who has trouble reading due to low vision, I want to be able to make the text larger on the screen so that I can read it.	I can access my account / dashboard	High	Sprint-1
Customer (Color blindness)		USN-2	As a user, who is color blind ,I want to have access to information conveyed in color so that, I do not miss anything and I understand the content.	I can receive confirmation email & click confirm	High	Sprint-1
Customer (Impaired user)		USN-3	As a user, who is hearing-mpaired, Iwant a transcript of the spoken audio so that I can have access to all information provided in audio clips	I can register & access the dashboard with Facebook Login	Low	Sprint-2

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

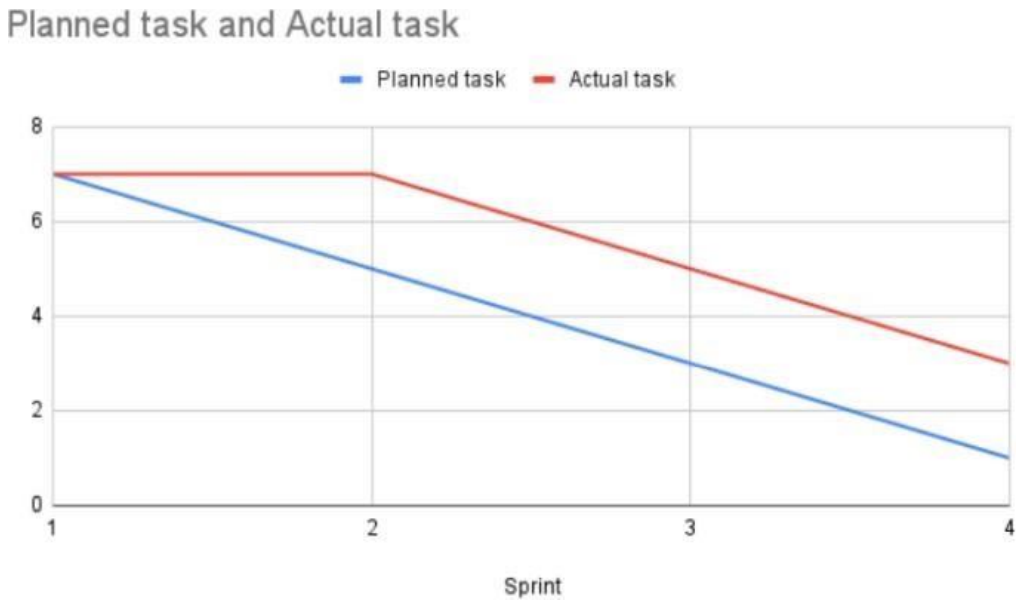
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Data collection	USN-1	Collection of required data, login information from user	2	Low	ARUNESHWARI
Sprint-1		USN-2	Image pre-processing	3	High	YAMUNA
Sprint-2	Model building	USN-3	Import the required libraries, add the necessary layers, and compile the model	2	Low	KAVIYA
Sprint-2		USN-4	Training the image classification model using CNN	3	High	ROSHINI
Sprint-3	Training and testing	USN-5	Training the model and testing the model's performance	3	High	NANDHINI
Sprint-3		USN-6	Converting the input sign language images into English alphabets and save model for deployment	2	Low	ARUNESHWARI
Sprint-4	Implementation and dashboard	USN-7	As a user, I can acknowledge the output of the system by ensuring messages are displayed.	2	Low	YAMUNA
Sprint-4		USN-8	As a user, I can get and give feedback about the system from its output.	3	High	KAVIYA

6.2 SPRINT DELIVERY SCHEDULE

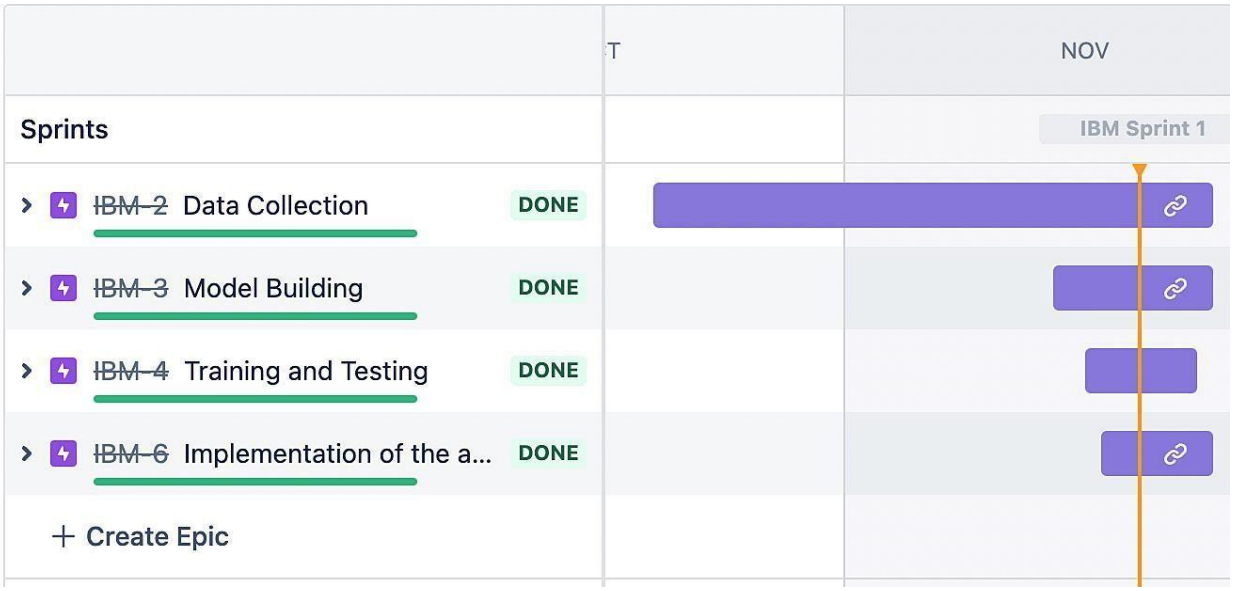
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	05	6 Days	24 Oct 2022	29 Oct 2022	05	05 Nov 2022
Sprint-2	05	6 Days	31 Oct 2022	05 Nov 2022	05	08 Nov 2022
Sprint-3	05	6 Days	07 Nov 2022	14 Nov 2022	05	12 Nov 2022
Sprint-4	05	6 Days	14 Nov 2022	19 Nov 2022	05	19 Nov 2022

Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



6.3 REPORT FROM JIRA



7. Coding and Solutioning

7.1 Libraries to be installed

- ✓ **pip install flask**
- ✓ **pip install opencv-python**
- ✓ **pip install numpy**
- ✓ **pip install keras**
- ✓ **pip install tensorflow**
- ✓ **pip install SpeechRecognition**
- ✓ **pip install moviepy**
- ✓ **pip install scikit-image**
- ✓ **pip install gTTS**
- ✓ **pip install Pillow**
- ✓ **pip install scikit-image**
- ✓ **pip install imutils**
- ✓ **pip install playsound**

7.2 Real time sign to speech

People who are unable to talk typically use sign language to communicate. The majority of people find it extremely challenging to communicate with silent persons since they are unable to grasp Universal Sign Language (unless they have studied it). The core of this project is a tool that allows silent people and others communicate with one another. Our technology makes use of a CNN- built model that can recognise sign languages in real time. It quickly recognises the sign and provides feel-free cover for those with special needs.

img = cv2.imread('static/image.jpg') # read a image

img=resize(img,(64,64,1)) # resize a image

img=image.img_to_array(img) # Convert image into array values

img=np.expand_dims(img,axis=0) if(np.max(img)>1):

```
img=img/255.0 prediction=model.predict(img)# predict the vaule using
trained model prediction=np.argmax(prediction, axis=1)
pred=vals[prediction[0]] # return predicted letter
```

7.3 Real time speech to sign

We can use JavaScript to recognise voice thanks to the Web Speech API. JavaScript makes it very simple to recognise speech in a browser and then extract the text from the speech for use as user input. The voice is translated into text using the Speech Recognition object, which is subsequently shown on the screen as signs. This can be accomplished by our technology in real time. Any language that the user is trying to speak in can be recognised by it. But only the Chrome browser is supported for this API. The live example below will function if you are viewing this example in another browser.

```
r=sr.Recognizer() #listen a auido file=sr.AudioFile("Speech/audio.mp3") with file as source:
    audio_data = r.record(source)#record a source audio text =
  r.recognize_google(audio_data,language='en-IN', show_all=True)
  text=text['alternative']
  text=text[0]
  text=text['transcript']# predict a text using source audio
```

8. Testing

8.1 Test Cases

- **Verify if user can see the options when user clicks the URL**
- **Verify if the UI elements are getting displayed properly**
- **Verify if the user is getting redirected to the sign to speech page**
- **Verify if the application can convert the sign to speech Verify if the user can exit the sign to speech page**
- **Verify if the user is getting redirected to the speech to sign page**
- **Verify if the UI elements are being displayed**
- **Verify if the application can convert speech to text on clicking voice to text button.**
- **Verify if the user can exit the speech to sign page.**

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

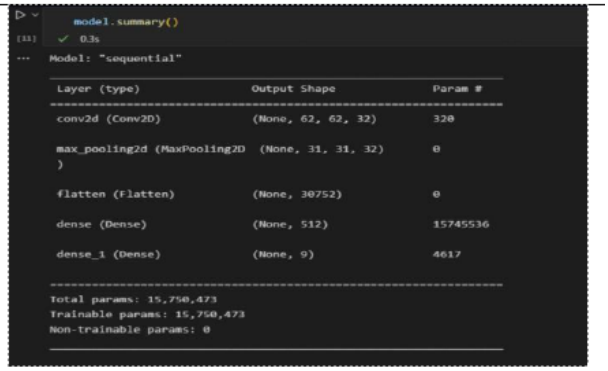
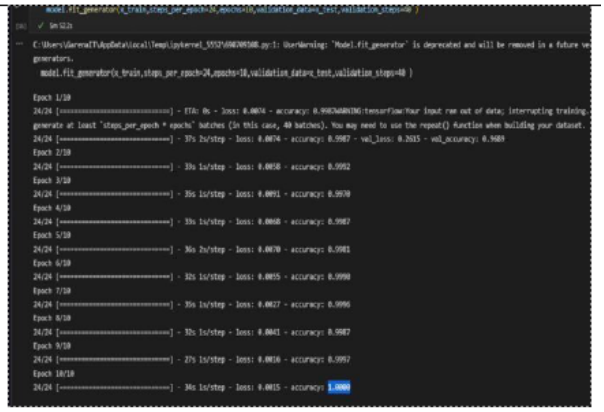
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	10	0	0	10
Security	3	0	0	3

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	4	2	3	17
Duplicate	2	0	3	0	5
External	3	3	0	1	7
Fixed	5	2	4	12	23
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	0	0	0	0
Totals	18	9	11	17	55

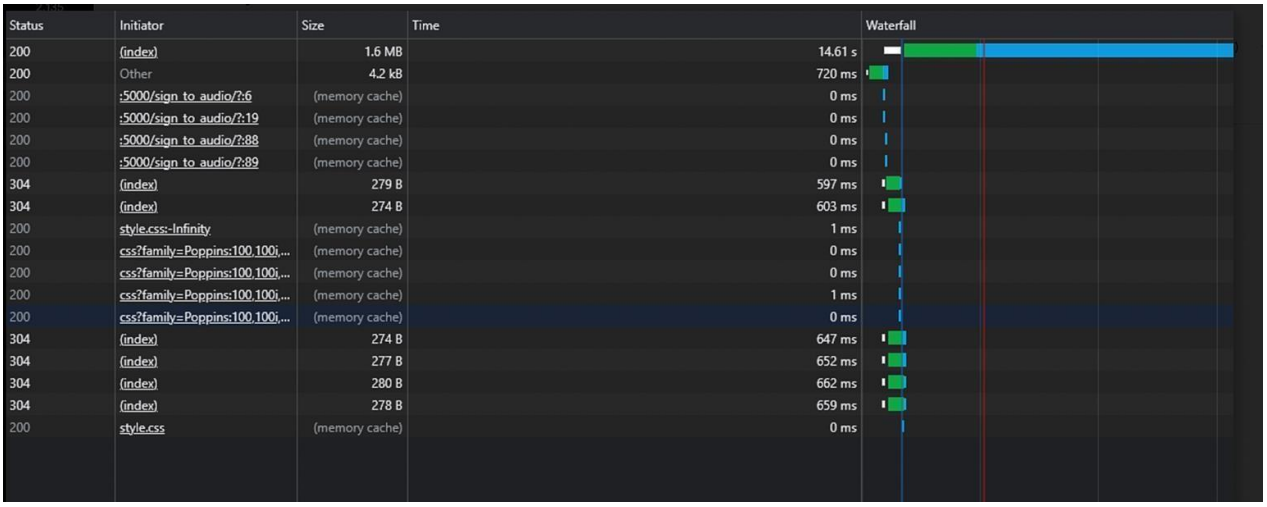
8.2 Performance Testing

S.No	Parameter	Values	Screenshot
1	Model Summary	Total params: 15,750,473 Trainable params: 15,750,473 Non-trainable params: 0	
2	Accuracy	Training Accuracy - 1.0000 Validation Accuracy - 0.9689	

9. Results

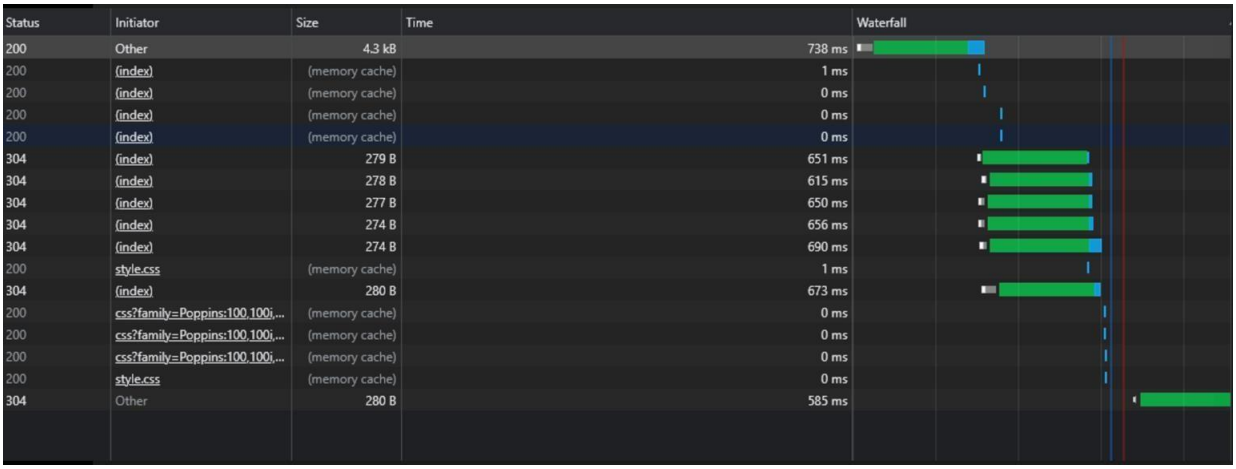
Performance Metrics

For Sign to speech



Status	Initiator	Size	Time	Waterfall
200	(index)	1.6 MB	14.61 s	
200	Other	4.2 kB	720 ms	
200	:5000/sign_to_audio/?:6	(memory cache)	0 ms	
200	:5000/sign_to_audio/?:19	(memory cache)	0 ms	
200	:5000/sign_to_audio/?:88	(memory cache)	0 ms	
200	:5000/sign_to_audio/?:89	(memory cache)	0 ms	
304	(index)	279 B	597 ms	
304	(index)	274 B	603 ms	
200	style.css:-Infinity	(memory cache)	1 ms	
200	css?family=Poppins:100,100i,...	(memory cache)	0 ms	
200	css?family=Poppins:100,100i,...	(memory cache)	0 ms	
200	css?family=Poppins:100,100i,...	(memory cache)	1 ms	
200	css?family=Poppins:100,100i,...	(memory cache)	0 ms	
304	(index)	274 B	647 ms	
304	(index)	277 B	652 ms	
304	(index)	280 B	662 ms	
304	(index)	278 B	659 ms	
200	style.css	(memory cache)	0 ms	

For Speech to sign



Status	Initiator	Size	Time	Waterfall
200	Other	4.3 kB	738 ms	
200	(index)	(memory cache)	1 ms	
200	(index)	(memory cache)	0 ms	
200	(index)	(memory cache)	0 ms	
200	(index)	(memory cache)	0 ms	
304	(index)	279 B	651 ms	
304	(index)	278 B	615 ms	
304	(index)	277 B	650 ms	
304	(index)	274 B	656 ms	
304	(index)	274 B	690 ms	
200	style.css	(memory cache)	1 ms	
304	(index)	280 B	673 ms	
200	css?family=Poppins:100,100i,...	(memory cache)	0 ms	
200	css?family=Poppins:100,100i,...	(memory cache)	0 ms	
200	css?family=Poppins:100,100i,...	(memory cache)	0 ms	
200	style.css	(memory cache)	0 ms	
304	Other	280 B	585 ms	

10. ADVANTAGES AND DISADVANTAGES

Advantages:

1. **Real time sign to speech detection.**
2. **Model provides good accuracy.**
3. **Real time facial emotion detection.**
4. **Language Customization.**
5. **Real time speech to text conversion.**
6. **Friendly UI**
7. **Data privacy**

Disadvantages:

1. **At times the website may lag.**
2. **Model is not tested on a wide set of data set, having all the signs.**
3. **Sign language customization feature is not available.**
4. **User cannot take notes while using the app.**
5. **User cannot make calls using the app.**
6. **Speech recognition works only on google chrome.**

11. CONCLUSION

The ability to express oneself requires communication. Additionally, it satisfies one's needs. Career advancement requires effective communication. By promoting mutual understanding, good
communication skills can simplify your daily life and enhance your connections with others. As part of our effort, a system that converts speech into suitable sign language for the deaf and dumb has been created. In order to converse with regular people, it also transforms sign language into a human voice. A model that has been trained on several hand motions has been created using a convolutional neural network. An app is produced using this idea. This software makes it simple for deaf and dumb people to communicate by using signs that can be translated into speech and human-understandable English.

12. FUTURE SCOPE

The following are the features that can be added in our application:

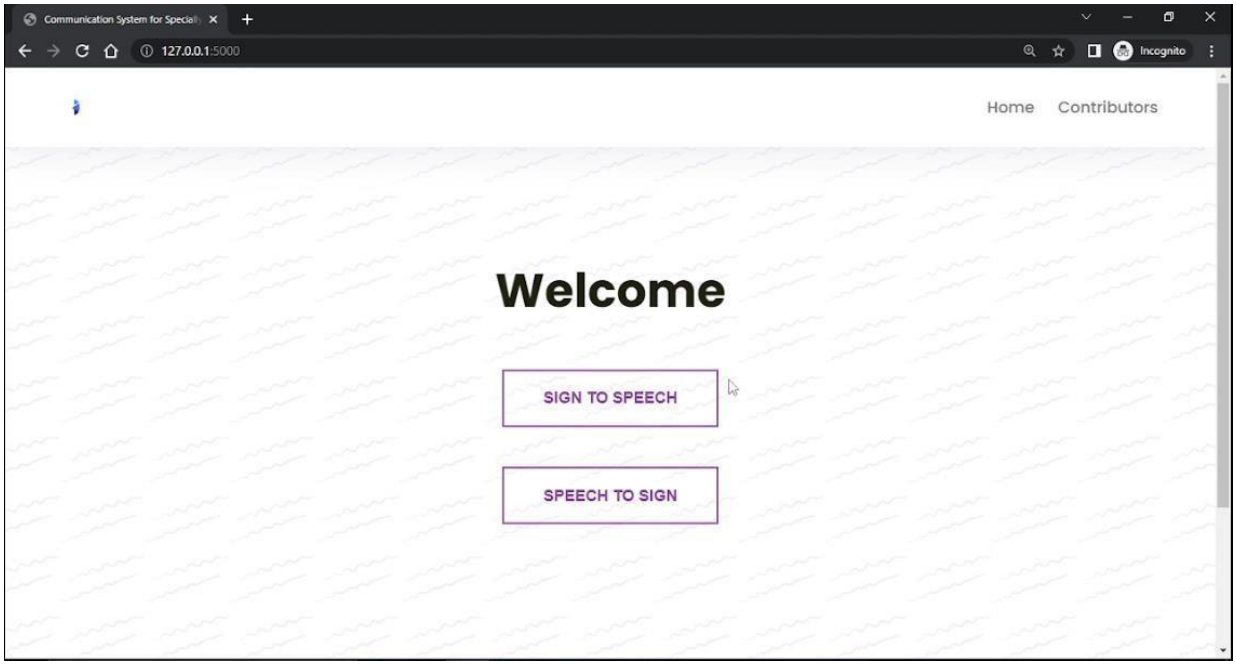
- **A communication app can be built with the same set of features. The user can choose the appropriate mode (speech to sign or sign to speech) and accordingly the real time detection would take place on both the end users' application.**
- **The accuracy of the model shall be increased.**
- **Customization of languages shall be added.**
- **Users shall be allowed to write notes while on call.**
- **Customization of signs can also be added as a feature.**

13. Appendix

Source Code

Index.html

```
<div class="section full-height" >
      <div class="absolute-center">
        <div class="section">
          <div class="container">
            <div class="row">
              <div class="col-12">


<h1><span>W</span><span>e</span><span>l</span><span>c</span><span>o</span><span>
m</span><span>e</span></h1>
            <br>

            <form action="/sign_to_audio/"><button class="btn fifth">Sign To
Speech</button></form>
            <form action="/audio_to_sign/"><button class="btn fifth">Speech to
Sign</button></form>

             </div>
            </div>
          </div>
        </div>

      </div>
```



app.py

```
from flask import Flask,
render_template,redirect,url_for,request,Response
from moviepy.editor import VideoFileClip import cv2
import speech_recognition as
sr from PIL import Image
import numpy as np from
skimage.transform import
resize from gtts import gTTS
import playsound
from keras.utils import image_utils as
image from keras.models import
load_model app=Flask(_name_)

vals=['A','B','C','D','E','F','G','H',

'I']
model=load_model('IBM.h5')

app.secret_key = "secret key"
arr = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l',
        'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z','.']

@app.route('/')
def index():
   return render_template("index.html")


@app.route('/audio_to_sign/')
def audio_to_sign():
   return render_template('SpeechtoSign.html')

@app.route('/audio',
methods=['POST']) def audio():
r = sr.Recognizer() frameSize = (281,
363)
out =
cv2.VideoWriter('./static/uploads/output_video.mp4',cv2.VideoWriter_fourcc(*'DIV
X'),
1,frameSize)
with open('Speech/audio.mp3', 'wb') as f:
      f.write(request.data)
   with      sr.AudioFile('Speech/audio.mp3')      as      source:
      audio_data = r.record(source)
    text = r.recognize_google(audio_data, language='en-IN', show_all=True)
```

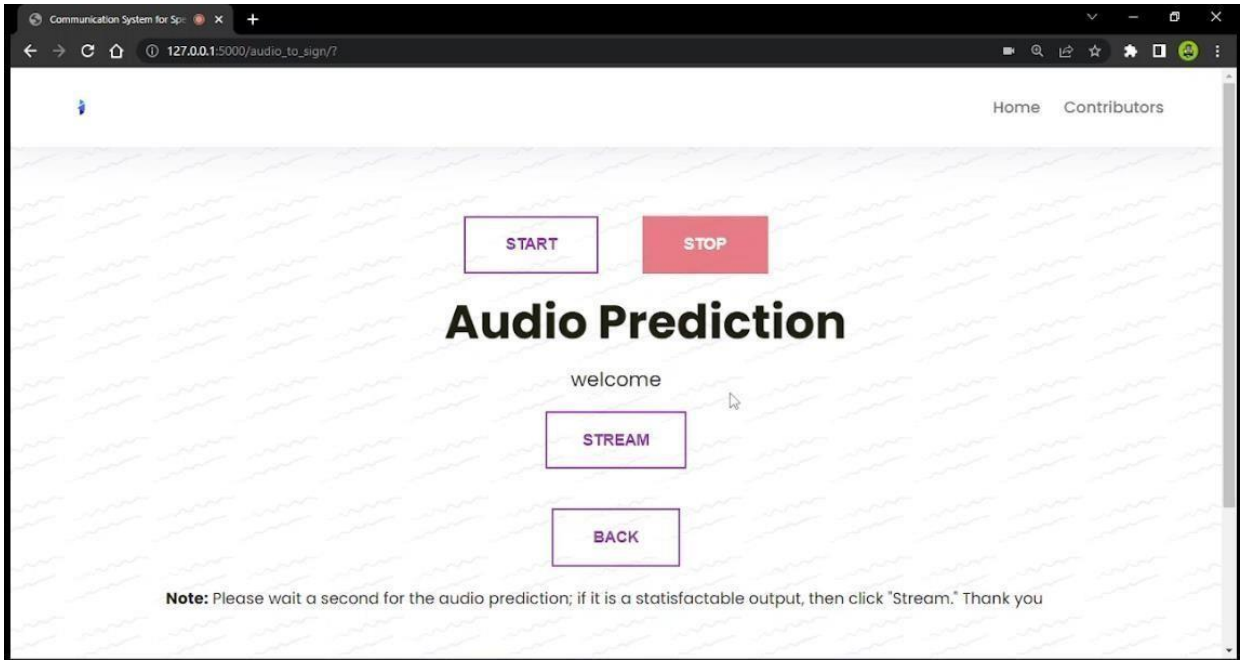
```
        print(text
        )       try:       for       num,       texts       in
           enumerate(text['alternative']):
              transcript = texts['transcript'].lower()
              print(transcript) break
        except:
           transcript = " Sorry!!!! Voice not Detected "

     for            i            in
        range(len(transcript)):
        if transcript[i] in arr:
           ImageAddress = 'L/'+transcript[i]+'.png'
           ImageItself = Image.open(ImageAddress)
           ImageNumpyFormat =
           np.asarray(ImageItself)                img                =
           cv2.imread(ImageAddress) out.write(img)
     out.release()
     videoFileClip=VideoFileClip("./static/uploads/output_video.mp4")
     videoFileClip.write_gif("./static/uploads/output_video.gif")
     videoFileClip.write_gif("./static/uploads/output_video1.gif") return
     str(transcript)

@app.route('/scrn',
methods=['POST'])                    def
upload_video():       r=sr.Recognizer()
file=sr.AudioFile("Speech/audio.mp3"
) with file as source:
        audio_data = r.record(source)
     text = r.recognize_google(audio_data, language='en-IN',
     show_all=True) text=text['alternative'] text=text[0]
     text=text['transcript']
     return render_template('stream.html', filename='output_video.gif',text=text)

@app.route('/display/<filename>')    def    display_video(filename):    return
redirect(url_for('static', filename='uploads/' + filename), code=301)

@app.route('/sign_to_audio/')
def sign_to_audio():
return render_template('SigntoSpeech.html')
def gen():
     string  =
     "          "
     count  =
     0   video
     =
```

```
cv2.VideoCapture(0) while (video.isOpened()): ret,
frame = video.read()
gray=cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY
)
  gray = cv2.threshold(gray, 0, 255,cv2.THRESH_BINARY_INV |
  cv2.THRESH_OTSU)[1] color_dict=(0,255,0) cv2.rectangle(frame,(24,24),(250 ,
  250),color_dict,2) copy=gray.copy()
  copy =
  copy[24:250,24:250] count
  = count + 1
  cv2.imwrite('static/image.jpg',copy)
  img =
  cv2.imread('static/image.jpg')
  img=resize(img,(64,64,1))
  img=image.img_to_array(img)
  img=np.expand_dims(img,axis=0)
  if(np.max(img)>1):
    img=img/255.0
  prediction=model.predict(img)
  prediction=np.argmax(prediction,
  axis=1) pred=vals[prediction[0]]
  print(pred) if(count == 200): count = 99
  prev= vals[prediction[0]] if(len(prev) ==
  0):
      string = string + "_"
      myobj = gTTS(text=string, lang="en", slow=False)
      myobj.save("./Speech/sign.mp3")
    else: string = string +
      prev
      myobj = gTTS(text=string, lang="en", slow=False)
      myobj.save("./Speech/sign.mp3")

  cv2.putText(frame, pred, (24,
14),cv2.FONT_HERSHEY_SIMPLEX,0.8,(255,255,255),2)
  cv2.putText(frame, string, (275,
50),cv2.FONT_HERSHEY_SIMPLEX,0.8,(200,200,200),2)
  if not
    ret:
    break
  else:
    ret,buffer=cv2.imencode('.jpg',frame)
    frame=buffer.tobytes()
  yield (b'--frame\r\n' b'Content-Type: image/jpeg\r\n\r\n' +
      frame + b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
```

```
    return Response(gen(), mimetype='multipart/x-mixed-replace;
                boundary=frame')

@app.route('/redirect
' ) def delet(): video =
cv2.VideoCapture(0)
video.release()
   return render_template("index.html")

@app.route('/play')
def play():
   playsound.playsound("./Speech/sign.mp3", True)
   return render_template("SigntoSpeech.html")

@app.route('/alp')
def alp():
   return render_template("Alphabet.html")

if_name_== "_main_": app.run(debug=True)
```

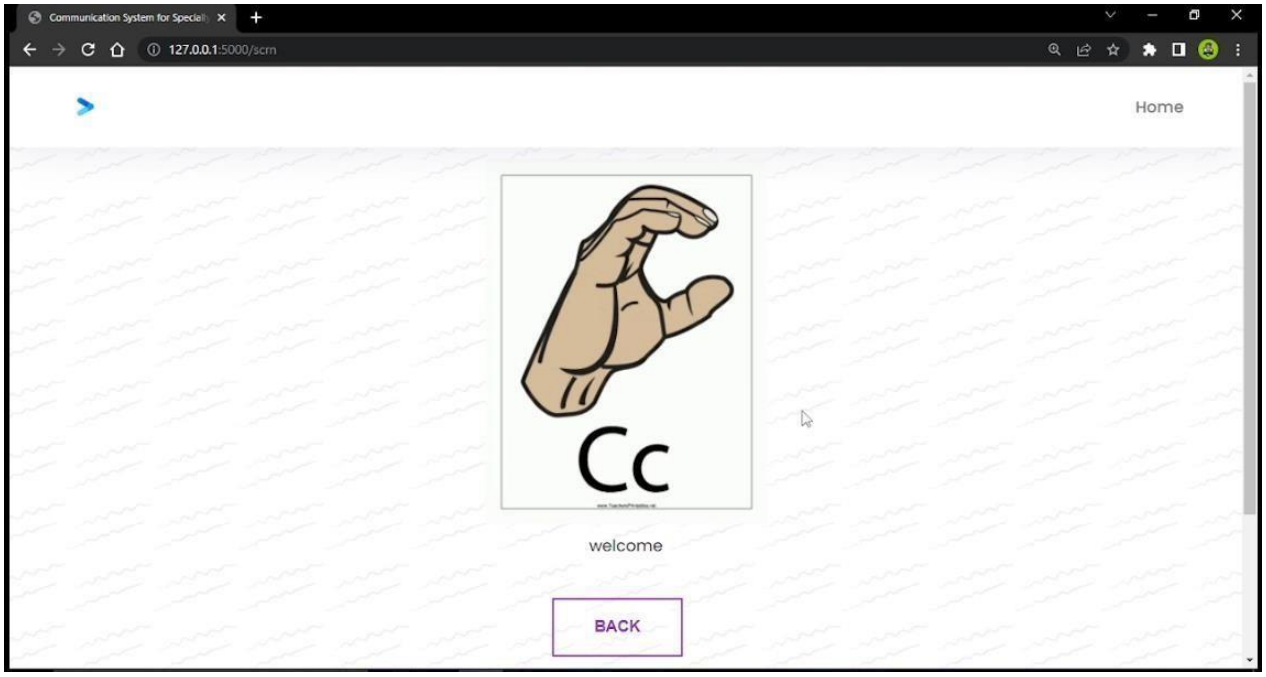
Speech to Sign.html

```
<div class="row">
  <div class="col-12">
    <button    id="start"    class="btn    btn-success"    onclick="startRecording()"
disabled>Start</button>
    <button    id="stop"    class="btn    btn-danger"    onclick="stopRecording()"
disabled>Stop</button>
  </div>
  <div class="column2">
    <h1 style="padding-left: 60px;">
      <span>Audio Prediction</span>
    </h1>
    <p class="glow" id="output" style="font-size: 20px;"></p>
    <form action="/scrn" method="post" enctype="multipart/form-data">
      <input  type="submit"  id="stream"  value="Stream"  class="btn  btn-info
stream" disabled> </form>
    <a href="/">
      <button  class="btn  btn-danger  btn-lg">Back</button>
    </a>
  </div>
  <div class="note">
    <b>Note:</b> Please wait a second for the audio prediction; if it is
a statisfactable output, then click "Stream." Thank you </div>
</div>
```



STREAM.HTML

```
<div class="section">
  <div class="container">
    <div class="row">
      <div class="col-12">
        <img src="{{ url_for('display_video', filename=filename) }}">
        <p>{{text}}</p>
      </div>
      <form action="/audio_to_sign/">
        <button class="btn fifth back">Back</button>
      </form>
    </div>
  </div>

</div>
```



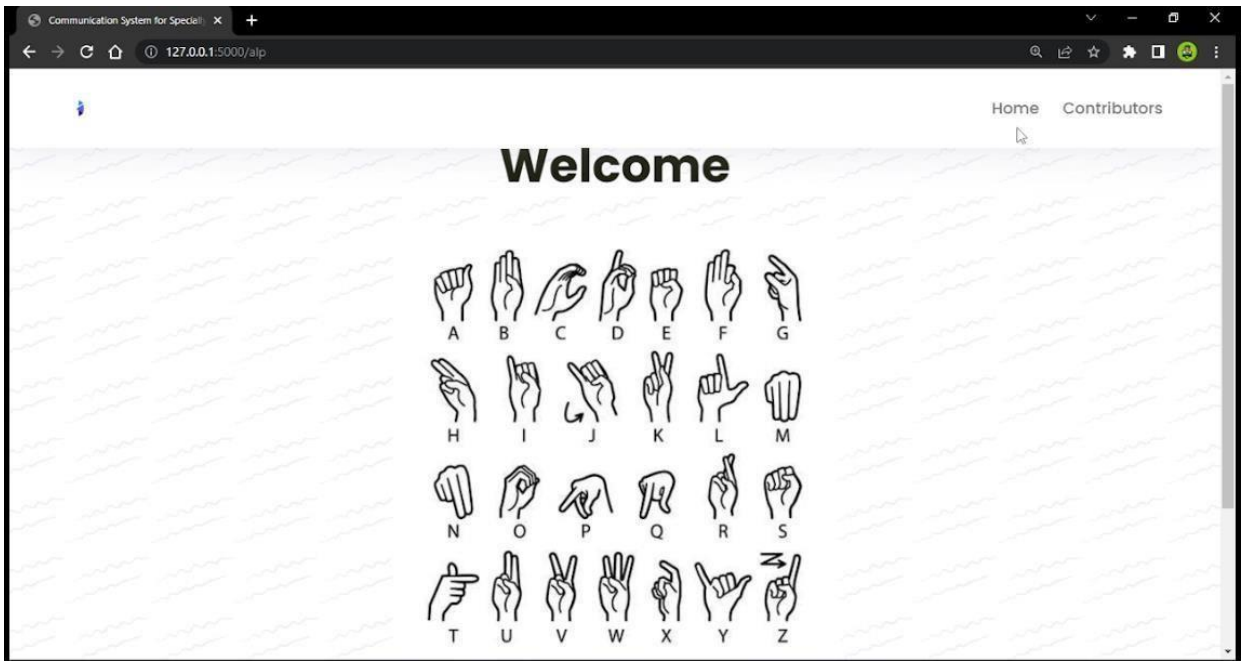
SigntoSpeech.html

```
<div class="column2">
  <h1 style="margin-top: 10%;">Live </h1>
  <br>
  <div class="stream-video">
    <img src="{{ url_for('video_feed') }}" width="576px" height="324px" />
    <br>
    <a href="/play">
   <button class="btn btn-info btn-lg" id="play">Play & Clear</button>
    </a>
    <a href="/redirect">
      <button class="btn btn-danger btn-lg">Exit</button>
    </a>
  </div>
  <div>
    <b>Note:</b> According to the given dataset, the model is designed for the alphabet A–I, so it predicts only limited alphabets And press the "Play & Clear" button after the predicted letter displays on the right side of the green rectangle. Thank you
  </div>
</div> <script>
```

```
document.getElementById("play").disabled      =
true;
setTimeout(function(){document.getElementById("play").disabled      =
false;},10000);
</script>
```

Alphabet.html

```
<div
      class="container">
       <div class="row">
         <div class="col-12">

         <h1><span>W</span><span>e</span><span>l</span><span>c</spa
n><span>o</span><span>m</span><span>e</span></h1>
                                    <br> <img
src="https://i.pinimg.com/564x/98/2c/d1/982cd1fb0bac83d6b12c9a6acd
ff0879.jpg" style="text-align: left;">
                                  </div>
                                  </div>
</div>
```



13.2 GitHub and Demo Link:

GitHub: https://github.com/IBM-EPBL/IBM-Project-37265-1660302714