Real-Time Communication System Powered By AI For Specially Abled

IBM Project Report
Artificial Intelligence

Submitted By

Team ID: PNT2022TMID35302

Team Members:

Prathiba D - 2019103557

Ramya P - 2019103569

Rohith Kumar M - 2019103572

Abhimanyu SG - 2019103578

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING, GUINDY

ANNA UNIVERSITY: CHENNAI 600 025

MAY 2022

CONTENTS

CHAPTER 1

Introduction

1.1 Project Overview

Communication is essential for a person to express thoughts and emotions. . In the modern world, despite the technical advancements, it is still a struggle for Specially-Abled people to communicate and express their thoughts with the rest of the world. Sign language is developed for specially impaired people so that they can communicate with ease. It is the most structured language where every gesture has a specific meaning associated to it.

But the normal people are not trained on hand sign language and conveying the message becomes difficult at times of emergency and causes a huge gap between the normal people and the inarticulate community.

Hence it is crucial to develop a solution that acts as a bridge between the hearing impaired and the hearing community.

1.2 Purpose

The project intends to create such an application which helps the specially challenged people to communicate among them and the common people. Communication between a person with hearing/speech impairment and a normal person has always been a challenging task. This application tries to reduce the barrier of communication by developing an assistive application for specially challenged people.

CHAPTER 2

Literature Survey

2.1 Existing Problem

[1] Title: Portable Communication Aid for Specially Challenged : Conversion of Hand Gestures into Voice and Vice Versa

Author: T Meera Devi, K M Shravan Raju

Methodology: The work is to develop a portable device for the disabled people who are not able to communicate with the normal persons properly. There are various steps involved in recognizing the feature distinguishing hand gesticulation. The collected gesticulation is trained using Neural Network. The hand movement pattern is separated from a continuous recording of gestures. Low-Level understanding for the feature pattern comprises the gestural segment.

Advantage: This will be useful for the normal people to communicate with differently abled people and vice versa.

Limitation: Separation of the hand movements from continuous hand gestures may result in accuracyissues.

[2]Title: Real-Time Two-Way Communication Approach for Hearing Impaired and Dumb Person Based on Image Processing.

Author: Shweta. S. Shinde, Rajesh M. Autee, Vitthal K. Bhosale

Methodology: Proposed system is based on vision-based hand recognition approach. The hand gestures are identified under varying illumination conditions The proposed

method performs background segmentation of the hand from the acquired data and then is assigned a particular gesture for different alphabets. It involves feature extraction methods to calculate peak calculation and angle calculation of hand gestures. Finally, the gestures are recognized by converting these gestures into speech and vice versa. For extracting the features of speech signal Mel-frequency cepstrum coefficients and dynamic time warping are used. The proposed system is based on MATLAB.

Advantage: Two-way communication is possible enabling effective communication between normal people and physically impaired

Limitations: Detected only limited hand gestures (From alphabets A to I) Memory consumption is high as image processing is done using the built-in model of MATLAB.

[3] Title: Hand Gesture Detection based Real-time American Sign Language Letters Recognition using Support Vector Machine

Authors: Xinyun Jiang, Wasim Ahmad

Methodology: Features extraction by Principal Component Analysis(PCA) SVM is used for mapping hand gestures.

Advantage: Principal Component Analysis used to select 8 features, reduces computational complexity and processing time.

Limitation: Reorientation stage- rotation angle of alphabets difficult to determine. Only static images are used.

[4] Title: Sign Language Recognition Using Deep Learning on Custom Processed Static Gesture Images.

Authors: Aditya Das, Shantanu Gawde, Khyati Suratwala, Dhananjay Kalbande

Methodology: CNN to recognize sign language gestures, Transfer learning using Inception v3.

Advantage: Average around 90% is obtained.

Limitation: Dynamic hand gestures are not used. Only static finger spellings are used.

[5] Title: Machine Learning Model for Sign Language Interpretation using Webcam Images.

Author: Kanchan Dabre, Surekha Dholay

Advantage: Prediction using Haar Cascade Classifier integrated with SVM, Classification based on supervised feed forward backpropagation algorithm. Convergence rate is faster. Average recognition rate: 91.11 %

Limitation: Haar Cascade Classifier compromises on precision.

[6] Title: MUDRAKSHARA - A Voice for Deaf/Dumb People

Author: Dr.Yeresime Suresh, J Vaishnavi, M Vindhya, Mohammed Sadiq Afreed Meeran, SuprithaVemala

Methodology: A system that recognizes hand gestures and performs the task same as translators is developed **-** MUDRAKSHARA. It identifies hand gestures in the images obtained from video that is captured by a web ‘cam recorder and gives the meaning of

signs made by hearing/speech disabled people thus making communication complete.

Advantages: Provides the opportunity for common people to understand sign language thus bridging the communication gap between the deaf/dumb and the common people. High accuracy because of the highly trained CNN model.

Limitation: The system does not respond to dynamic hand gestures. Compared to other latest algorithms, CNN is a bit slow.

2.2 References

[1] https://ieeexplore.ieee.org/document/8997140

[2] https://ieeexplore.ieee.org/document/7919572

[3] https://ieeexplore.ieee.org/document/8890379

[4] https://ieeexplore.ieee.org/document/8537248

[5] https://ieeexplore.ieee.org/document/6839279

[6] https://ieeexplore.ieee.org/document/9225656

2.3 Problem Statement Definition

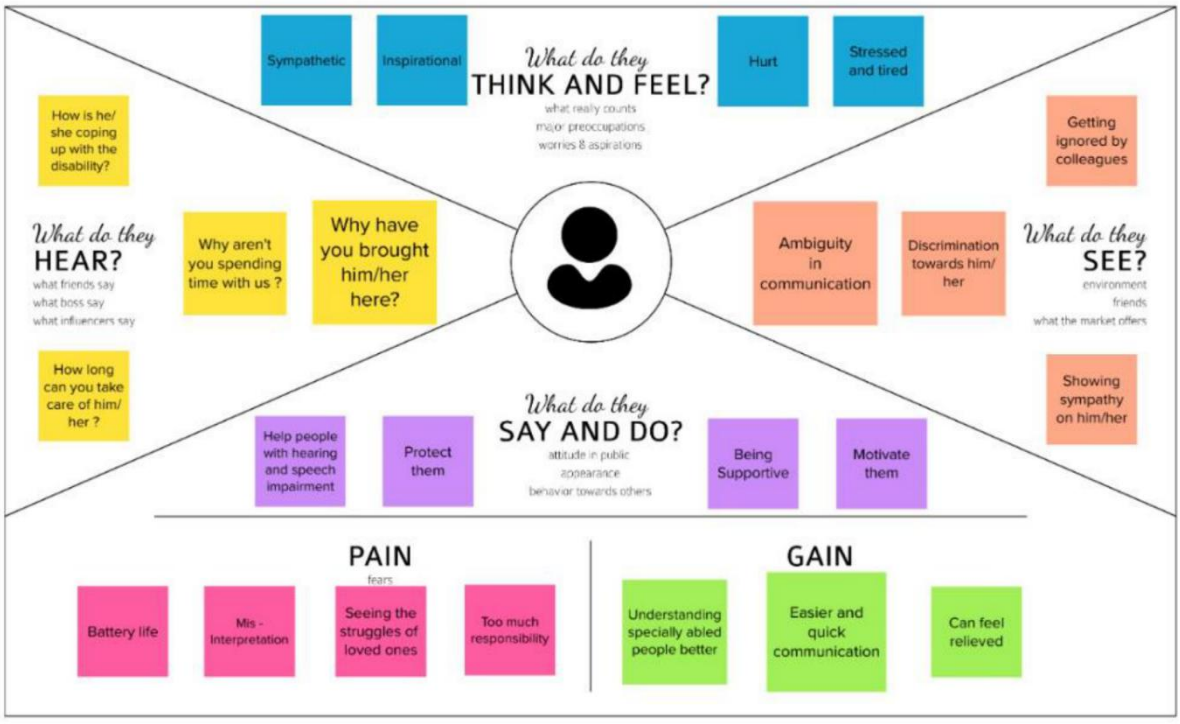
The project aims to develop a system that converts the sign language into a human hearing voice in the desired language as well as to convert speech into understandable sign language for the deaf and dumb. A convolution neural network is used to create a model that will be trained on different hand gestures. A web application to use the model will be built. This application will enable the deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.

CHAPTER 3

Ideation and Proposed Solution

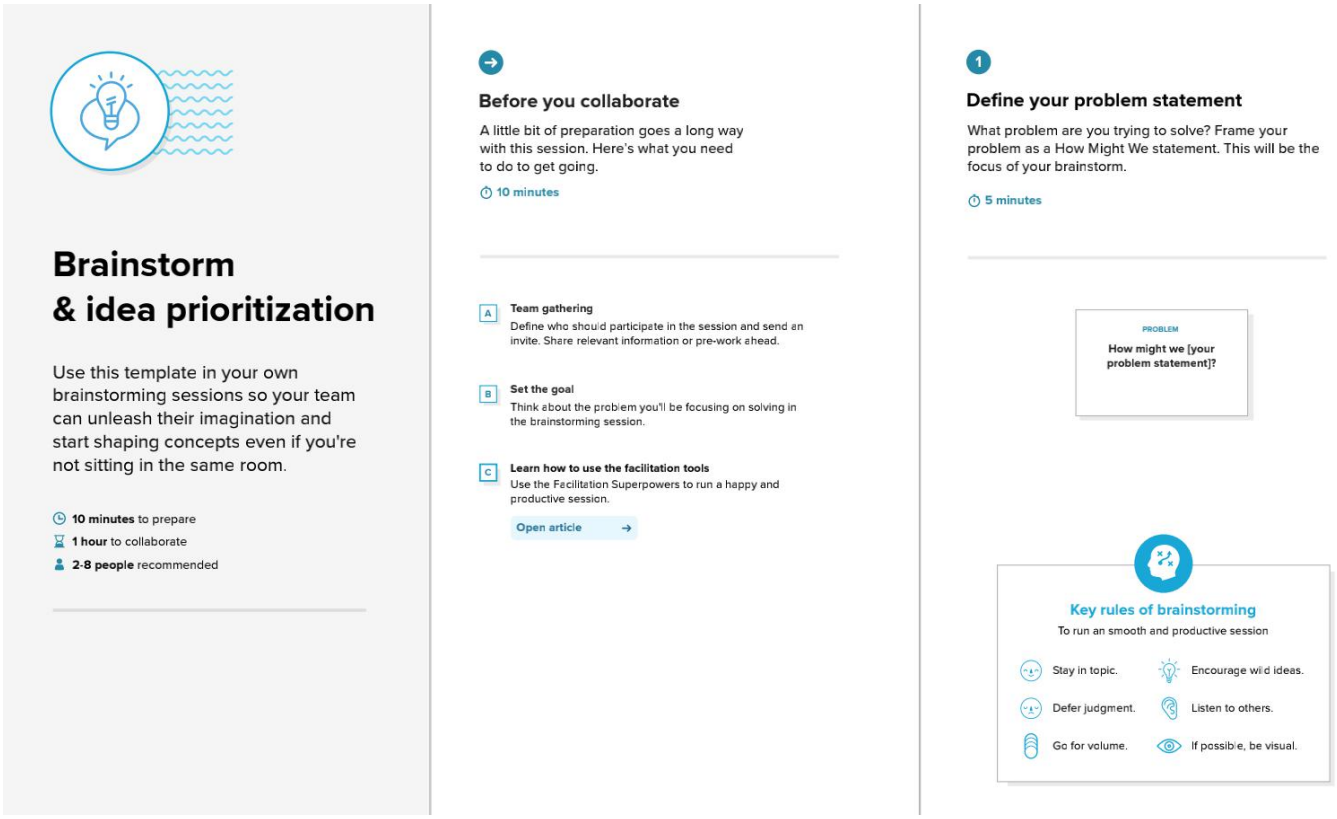
3.1 Empathy Map Canvas

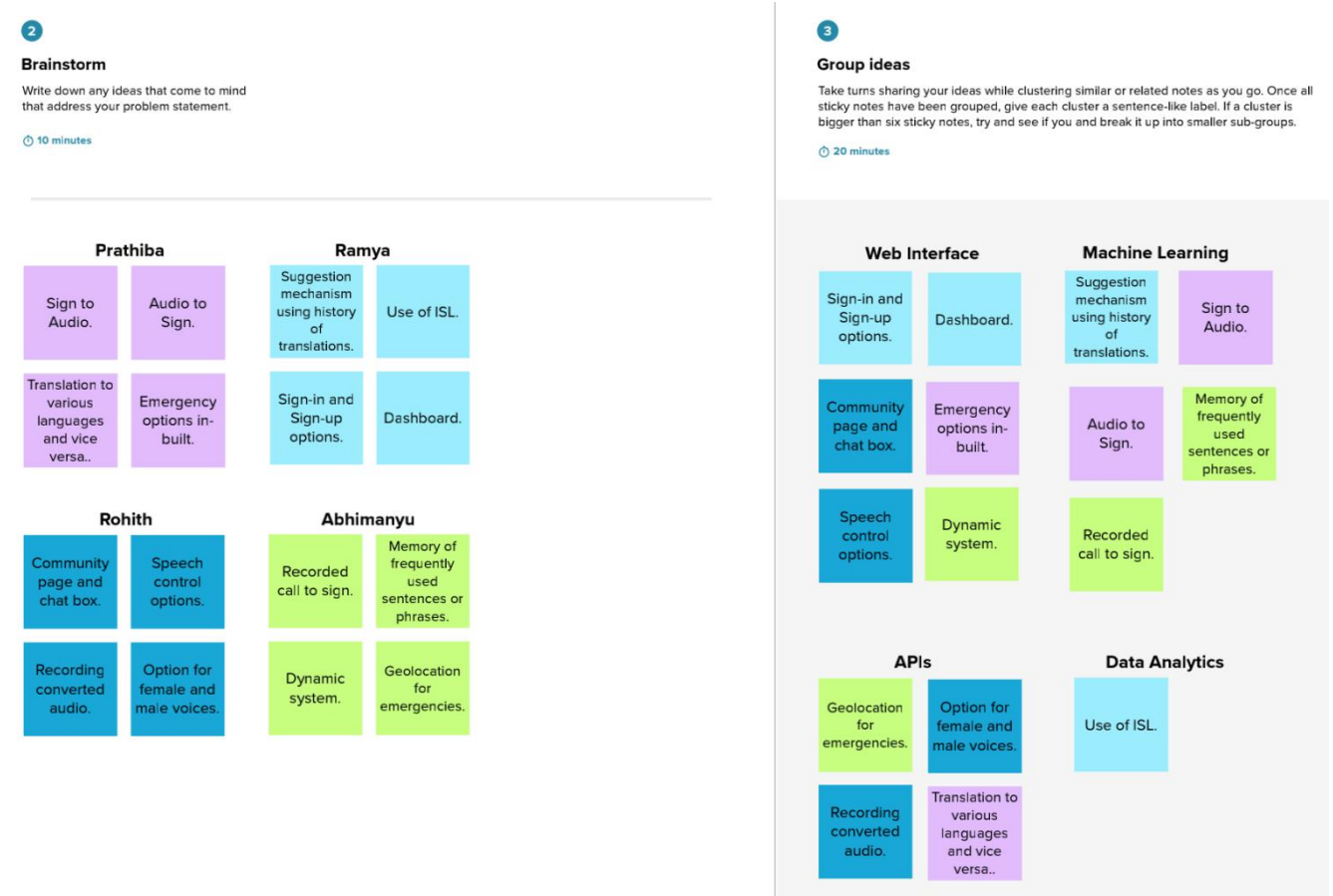
User1 (Person with hearing/speech impairment)

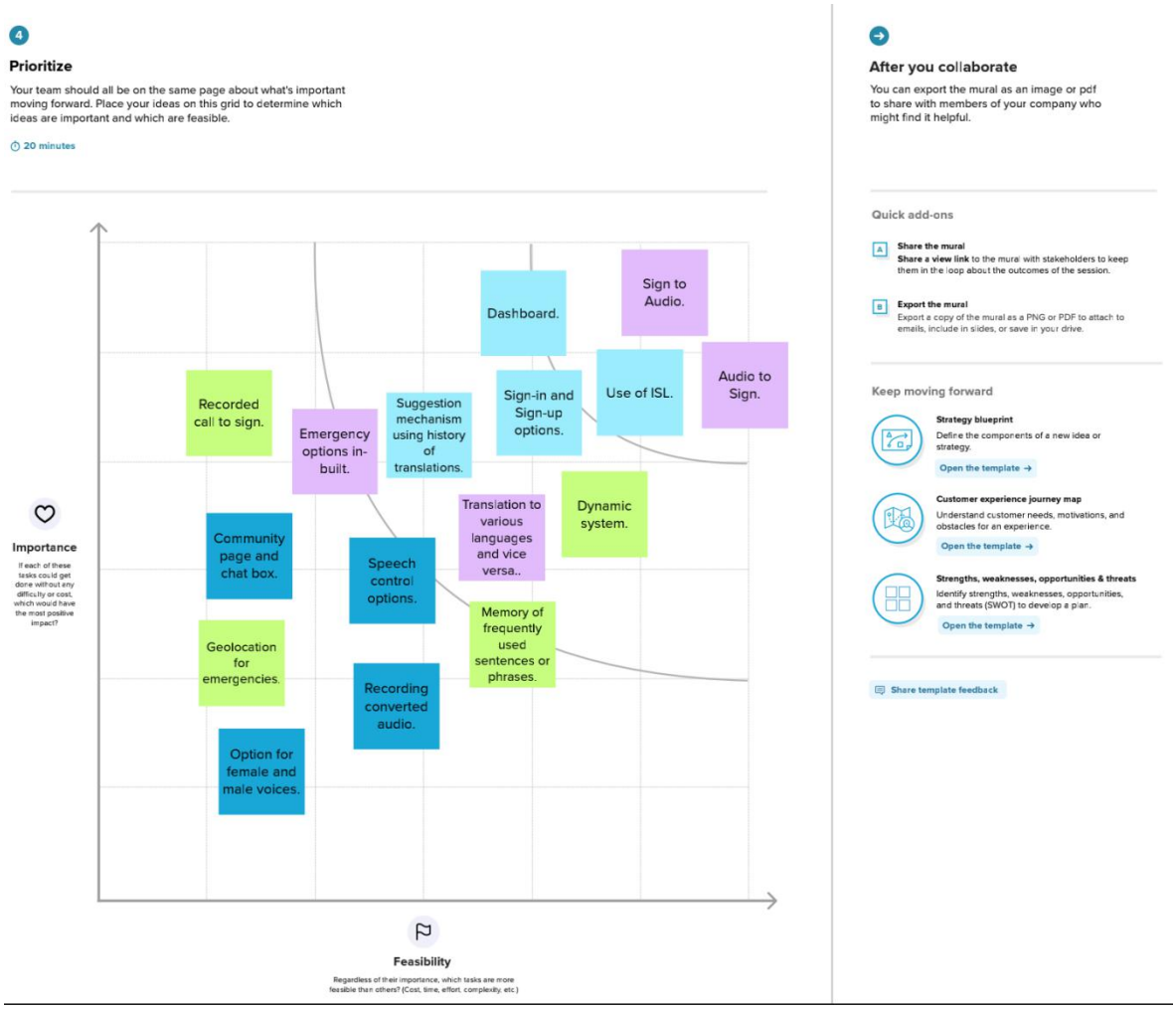


User2 Person without any impairments (parent/friend/relative/colleague)

3.2 Ideation & Brainstorming







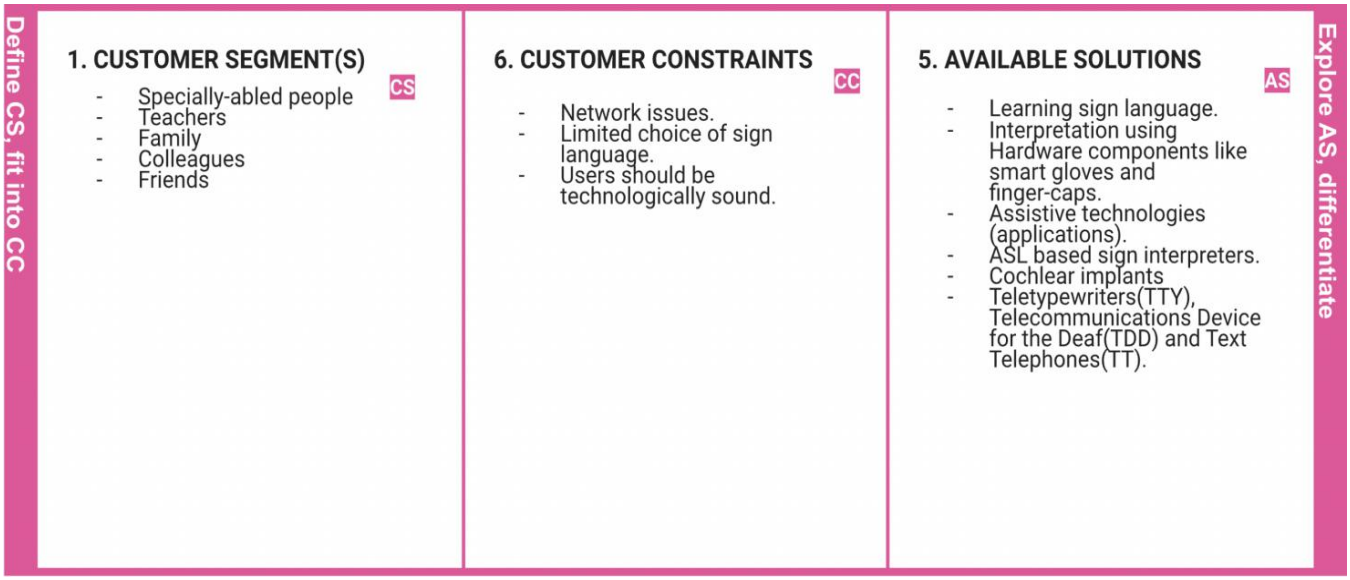
3.3 Proposed Solution

S No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The project aims to develop a system that converts the sign language into a human hearing voice in the desired language as well as to convert speech into understandable sign language for the deaf and dumb. A convolution neural network is used to create a model that will be trained on different hand gestures. A web application to use the model will be built.

2.	Idea / Solution description	The idea is to develop a system that converts the sign language into a human hearing voice in the desired language to convey a message to normal people, as well as convert speech into understandable sign language for the deaf and dumb. We are making use of a convolution neural network to create a model that is trained on different hand gestures. An app is built which uses this model. This app enables deaf and dumb people to convey their information using signs which get converted to human-understandable language and speech is given as output.
3.	Novelty/ Uniqueness	➢ Two-way communication. ➢ An application with embedded features: ➢ Translation into various languages. ➢ Suggestion mechanism based on the history of translations. ➢ In-built emergency options.
4.	Social Impact / Customer Satisfaction	➢ Understanding specially-abled people in a better way. ➢ Equal opportunities ➢ Break the social barrier
5.	Business Model	Not many models of this kind are available in the society, so educating people about the uses and technology of this

	(Revenue Model)	model will surely create a huge market for this product. This can be done by means of mass communication and advertisements. Instead of promoting it in business point of view, promoting it in a service point of view will expand its reach.
6.	Scalability of the Solution	If a basic model of this solution is created, then expanding it, needs some minimal man power with sufficient knowledge regarding this. So, with the increase in its demand, we can scale its production comfortably.

3.4 Problem Solution fit



Focus on J&P, tap into BE,

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- There is a need to develop a system to convert sign language to speech and vice versa.
- There should be an application to convey the information.

9. PROBLEM ROOT CAUSE RC

- Everyone does not know the sign language.
- Inability to communicate normally and effectively.
- Improper interpretation.

7. BEHAVIOUR BE

- Text usage to convey information.
- Use of understandable signs.
- Lip reading.

Focus on J&P, tap into BE,

3. TRIGGERS TR

- Seeing people being bullied and isolated.
- Inability to convey their thoughts during emergencies.
- Frustration upon missing opportunities.
- Wish to lead a normal life.

4. EMOTIONS: BEFORE / AFTER EM

Before

- Socially secluded
- Dependent
- Hurt

After

- Feel equal
- Confident
- Relieved

10. YOUR SOLUTION SL

The project deals with building an application which helps the specially challenged people to communicate between them and the common people. This application tries to reduce the barrier of communication by developing an assistive application for specially challenged people.

8. CHANNELS of BEHAVIOR CH

8.1 ONLINE

Video calls for distant communication involving either sign language or lip reading.

8.2 OFFLINE

Dependent on a person for communication assistance.

Identify strong TR & EM

CHAPTER 4

Requirement Analysis

4.1 Functional requirement

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 Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Live capturing	Camera enabled devices can be used
FR-4	Translation	Conversion of sign language to text.
FR-5	Speech	Text to speech conversion.

4.2 Non-Functional requirements

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

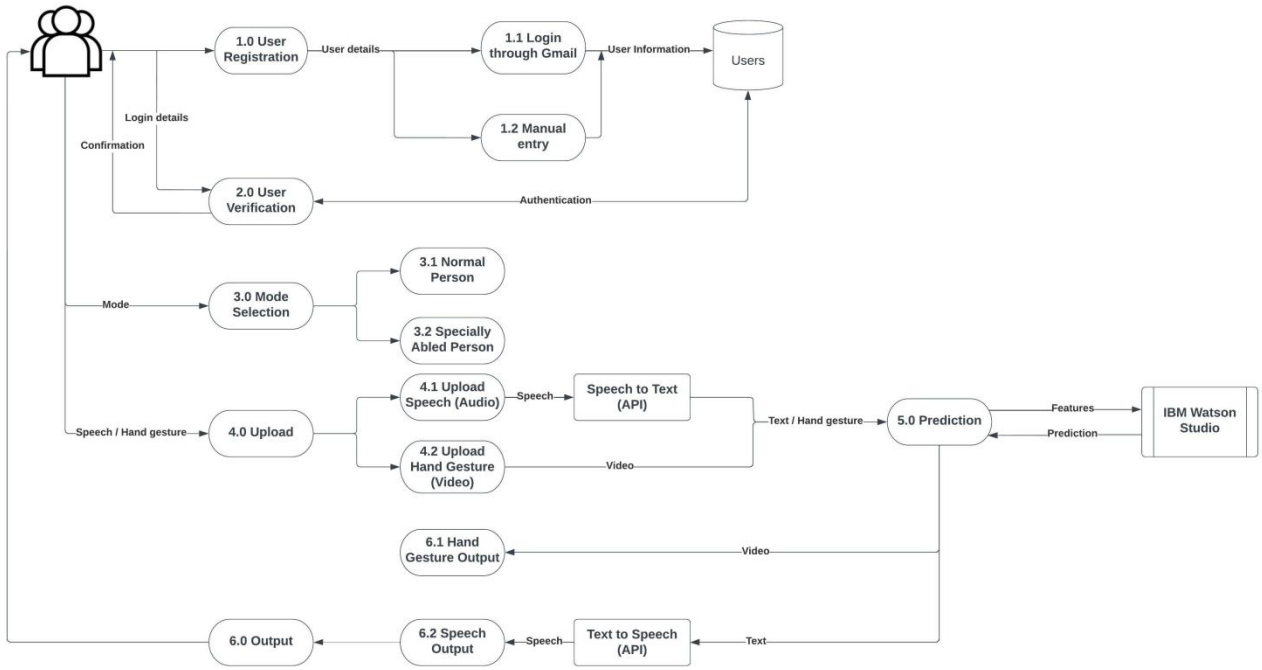
FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Simple GUI for easier interactions.
NFR-2	Security	Authentication, authorization and encryption of the application using IBM Watson cloud security.
NFR-3	Reliability	System should work dynamically without any failure for most of the time.

NFR-4	Performance	By using IBM Cloud APM, data center, cloud infrastructure, and workloads are managed with cognitive intelligence. Outages and slowdowns can be reduced and prevented around the clock in a hybrid application world as Cloud APM assists in moving from identifying performance issues to isolating where the problem is occurring and diagnosing issues before the application is impacted.
NFR-5	Availability	IBM Cloud uses global load balancing to ensure that a redundant, highly available platform is available to host the workloads and applications.
NFR-6	Scalability	IBM Cloud Bare metal servers help in achieving scalability whenever needed especially when number of users increase.

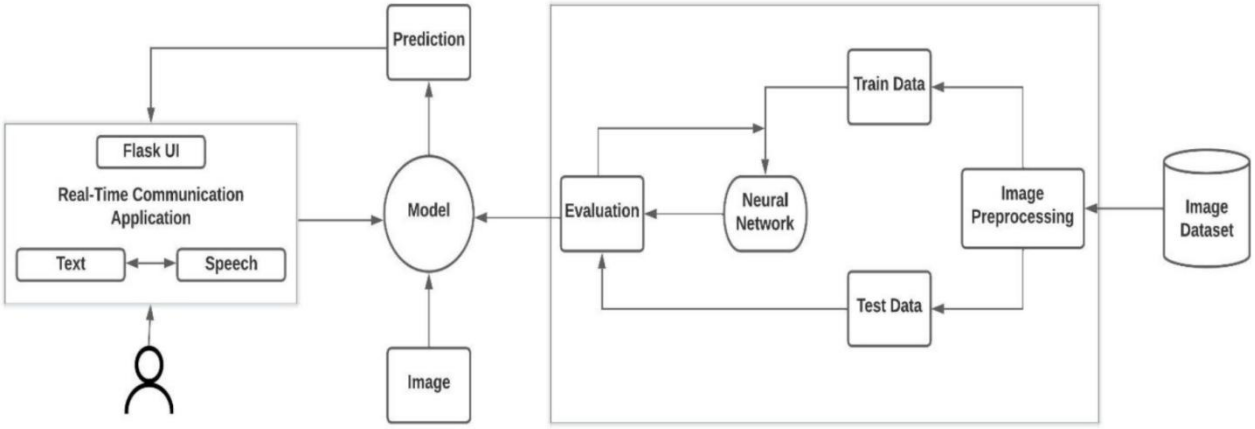
CHAPTER 5

Project Design

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Specially Abled Person)	Registration	USN-1	As a user, I can register to create an account for the application by entering my username, email and password.	I can access my account and select the mode of usage	High	Sprint-1
	Validation	USN-2	As a user, I will be authenticated to the website after checking my account credentials.		Medium	Sprint-2
	Login	USN-3	As a user, I will be prompted to select the mode of communication and I will select the specially abled mode (Gesture to Speech)	Credentials has to be matched	High	Sprint-3
	Mode Selection	USN-4	As a user of this mode I will capture my hand gesture as video	Either of the modes has to be chosen for further processing	High	Sprint-1
	Video Capturing	USN-5	As a user of this mode, Iwill be able to receive	Minimum video quality criteria	Low	Sprint-1

			and interpret the translated gestures from the other end.	has to be met		
	Gesture interpretation	USN-6	As a user, I can register for the application by entering my email, password, and confirming my password.	Must be a valid gesture	High	Sprint-1
Customer (Normal Person)	Registration	USN-7	As a user, I can register to create an account for the application by entering my username, email and password.	I can access my account and select the mode of usage	High	Sprint-1
	Validation	USN-8	As a user, I will be authenticated to the website after checking my account credentials.		Medium	Sprint-2
	Confirmation	USN-9	As a user, I can log into the application by entering email & password	I can receive confirmation email & click confirm	Medium	Sprint-2
	Login	USN-10	As a user, I will be prompted to select the mode of	Credentials has to be matched	High	Sprint-3

			communicati on and I will select the specially abled mode (Gesture to Speech)			
	Mode Selection	USN-11	As a user of this mode I will record the speech in order to convert it into gesture	Either of the modes has to be chosen for further processing	High	Sprint-1
	Speech Recording	USN-12	As a user of this mode, I will be able to receive and interpret the translated speech from the other end.	Minimum audio quality criteria has to be met	Low	Sprint-1
	Speech recognition	USN-13	As an admin, I will be responsible for controlling the user activities and further upgradations of the application	The words must be a recognizabl e	Mediu m	Sprint-3
Administr ator	Application monitoring and controlling	USN-14	As an admin, I will be responsible for controlling the user activities and further upgradations of the app.	Admin level privilege	Mediu m	Sprint-3

CHAPTER 6

Project Planning and Scheduling

6.1 Sprint Planning & Estimation

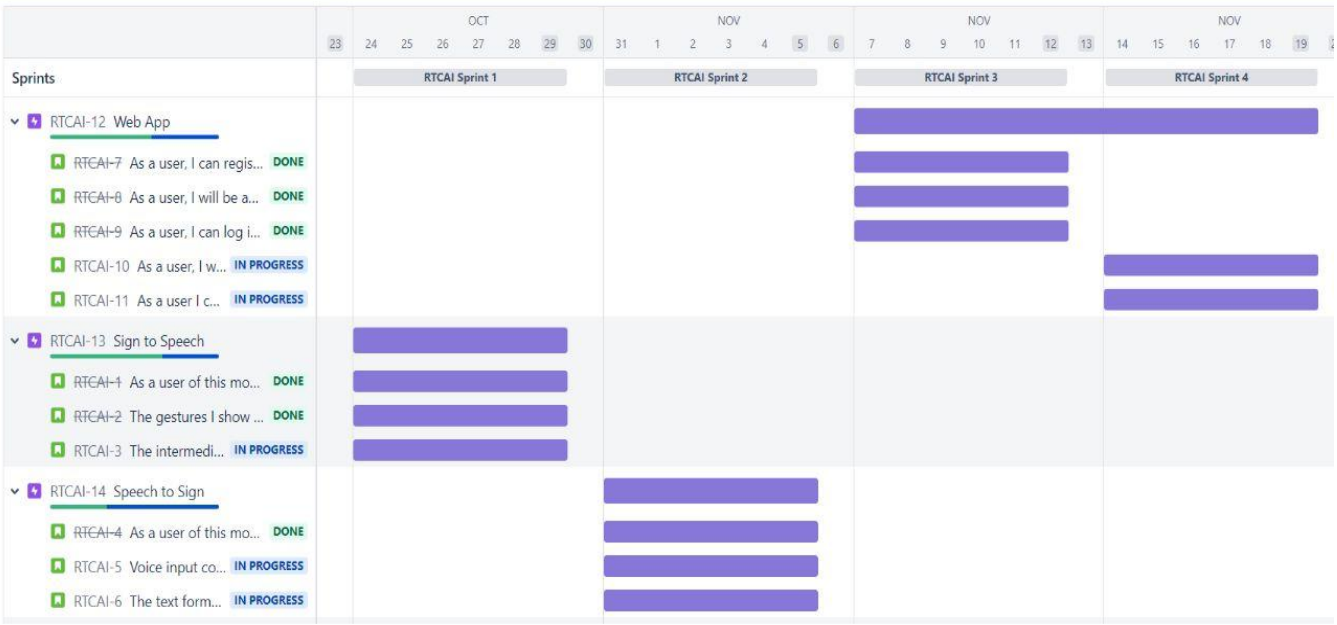
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
3	Registration	USN-1	As a user, I can register to create an account for the application by entering my username, email and password.	5	High	• Prathiba D • Rohith Kumar M
3	Validation	USN-2	As a user, I will be authenticated to the website after checking my account credentials.	5	Medium	• Ramya P • Rohith Kumar M
3	Login	USN-3	As a user, I can log into the account by entering the appropraite credentials	5	Medium	• Prathiba D • Sendamangalam Gandhi Abhimanyu
4	Mode Selection	USN-4	As a user, I will be prompted to select the mode of communication.(Sign to Speech / Speech to Sign)	10	High	• Prathiba D • Rohith Kumar M
1	Video Capturing (Sign to Speech)	USN-5	As a user of this mode my hand gesture will be captured once I click the 'Click to Start Capture' button.	4	Medium	• Rohith Kumar M • Sendamangalam Gandhi Abhimanyu

1	Gesture to text conversion (Sign to Speech)	USN-6	The gestures I show will be converted into a text format internally and will be displayed on the screen.	12	High	• Prathiba D • Ramya P
1	Text to speech conversion (Sign to Speech)	USN-7	The intermediate text format will be converted to speech using API and will be produced as output.	4	Medium	• Rohith Kumar M • Sendamangalam Gandhi Abhimanyu
2	Speech Recording (Speech to Sign)	USN-8	As a user of this mode my voice/speech input will be recorded once I click the 'Click to Record' button.	4	Medium	• Ramya P • Sendamangalam Gandhi Abhimanyu
2	Speech to text conversion (Speech to Sign)	USN-9	Voice input converted to text format internally and will be displayed in a text box	8	Medium	• Ramya P • Rohith Kumar M
2	Text to gesture conversion (Speech to Sign)	USN-10	The text format is converted into sign language gestures(displayed as images) using an alphabet to image mapper function.	8	High	• Prathiba D • Sendamangalam Gandhi Abhimanyu
4	Logout	USN-11	As a user I can logout from my account at any point of time.	10	Low	• Ramya P • Sendamangalam Gandhi Abhimanyu

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story points completed (as on planned date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA



CHAPTER 7

Coding and Execution

7.1 Feature 1 - Sign to Speech

Model Building

Sign to Text

Loading the Dataset & Image Data Generation

In [1]:
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [2]:
```
# Training Dataset
train_datagen = ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
# Testing Dataset
test_datagen = ImageDataGenerator(rescale=1/255)
```

In [4]:
```
# Training Dataset
x_train=train_datagen.flow_from_directory('training_set',target_size=(64,64), class_mode='categorical',batch_size=900)
# Testing Dataset
x_test=test_datagen.flow_from_directory('test_set',target_size=(64,64), class_mode='categorical',batch_size=900)
```

```
Found 15750 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

In [5]:
```
print("Len x-train : ", len(x_train))
print("Len x-test : ", len(x_test))
```

```
Len x-train :  18
Len x-test :  3
```

In [6]:
```
# The Class Indices in Training Dataset
x_train.class_indices
```

Out[6]:
```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```

Model Creation

In [6]:
```
# Importing Libraries
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

In [7]:
```
# Creating Model
model=Sequential()
```

In [8]:
```
# Adding Convolution Layers
model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))
```

In [9]:
```
# Adding Pooling Layers
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [10]:
```
# Adding Flatten Layers
model.add(Flatten())
```

In [11]:
```
# Adding Dense Layers
model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(9,activation='softmax'))
```

In [12]:
```
# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

In [13]:
```
# Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
```

```
C:\Users\91936\AppData\Local\Temp/ipykernel_20056/1042518445.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
Epoch 1/10
18/18 [==============================] - 40s 2s/step - loss: 1.0239 - accuracy: 0.6818 - val_loss: 0.3153 - val_accuracy: 0.9173
Epoch 2/10
18/18 [==============================] - 39s 2s/step - loss: 0.1846 - accuracy: 0.9447 - val_loss: 0.2546 - val_accuracy: 0.9547
Epoch 3/10
18/18 [==============================] - 40s 2s/step - loss: 0.0730 - accuracy: 0.9806 - val_loss: 0.2105 - val_accuracy: 0.9733
Epoch 4/10
18/18 [==============================] - 41s 2s/step - loss: 0.0369 - accuracy: 0.9910 - val_loss: 0.2486 - val_accuracy: 0.9636
Epoch 5/10
18/18 [==============================] - 41s 2s/step - loss: 0.0221 - accuracy: 0.9952 - val_loss: 0.2183 - val_accuracy: 0.9689
Epoch 6/10
18/18 [==============================] - 42s 2s/step - loss: 0.0143 - accuracy: 0.9970 - val_loss: 0.2178 - val_accuracy: 0.9756
Epoch 7/10
18/18 [==============================] - 42s 2s/step - loss: 0.0092 - accuracy: 0.9983 - val_loss: 0.2357 - val_accuracy: 0.9764
Epoch 8/10
18/18 [==============================] - 44s 2s/step - loss: 0.0093 - accuracy: 0.9978 - val_loss: 0.2137 - val_accuracy: 0.9769
Epoch 9/10
18/18 [==============================] - 47s 3s/step - loss: 0.0069 - accuracy: 0.9986 - val_loss: 0.2290 - val_accuracy: 0.9773
Epoch 10/10
18/18 [==============================] - 43s 2s/step - loss: 0.0059 - accuracy: 0.9990 - val_loss: 0.2644 - val_accuracy: 0.9769
```

Out[13]:

Saving the Model

In [ ]:
```
#Saving the model
model.save('aslpng1.h5')
```

Testing the model

In [1]:
```
#Importing the necessary packages
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

In [9]:
```
#Loading the saved model
model=load_model('aslpng1.h5')
```

In [10]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 62, 62, 32)        896

 max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
 )

 flatten (Flatten)           (None, 30752)             0

 dense (Dense)               (None, 300)               9225900

 dense_1 (Dense)             (None, 150)               45150

 dense_2 (Dense)             (None, 9)                 1359

=================================================================
Total params: 9,273,305
Trainable params: 9,273,305
Non-trainable params: 0
_________________________________________________________________
```

Test Cases

Alphabet 'G'

In [12]:
```
#Loading the test image
test_image=image.load_img('test_set/G/162.png',
                  target_size=(64,64))
test_image
```

Out[12]:

Image Preprocessing

In [13]:
```
#image to array conversion
tmp=image.img_to_array(test_image)
```

In [14]:
```
#Image Dimension expansion
print('Image Dimension before expansion',tmp.ndim)
tmp=np.expand_dims(tmp,axis=0)
print('Image Dimension after expansion',tmp.ndim)
```

```
Image Dimension before expansion 3
Image Dimension after expansion 4
```

Prediction

In [15]:
```
prediction=np.argmax(model.predict(tmp),axis=1)
index=['A','B','C','D','E','F','G','H','I']
print(index[prediction[0]])
```

```
1/1 [==============================] - 0s 134ms/step
G
```

Alphabet 'H'

In [26]:
```
#Loading the test image
test_image=image.load_img('test_set/H/44.png',
                  target_size=(64,64))
test_image
```

Out[26]:

Image Preprocessing

In [27]:
```
#image to array conversion
tmp=image.img_to_array(test_image)
```

In [28]:
```
#Image Dimension expansion
print('Image Dimension before expansion',tmp.ndim)
tmp=np.expand_dims(tmp,axis=0)
print('Image Dimension after expansion',tmp.ndim)
```

```
Image Dimension before expansion 3
Image Dimension after expansion 4
```

Prediction

In [29]:
```
prediction=np.argmax(model.predict(tmp),axis=1)
index=['A','B','C','D','E','F','G','H','I']
print(index[prediction[0]])
```

```
1/1 [==============================] - 0s 25ms/step
H
```

Text to Speech

```
import pyttsx3

def TexttoSpeech(command):
    engine = pyttsx3.init()
    engine.say(command)
    engine.runAndWait()

text=input("Enter the text ")
TexttoSpeech(text)
```

Real-time Gesture Recognition

Test Cases

In [6]:

```
import numpy as np
import keras
import os
import cv2
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
from skimage.transform import resize

index = ['A','B','C','D','E','F','G','H','I']
model=load_model('model.h5')
video = cv2.VideoCapture(0)
while True:
    success, frame = video.read()
    cv2.imwrite('frame.jpg', frame)
    img=resize(frame,(64,64,1))
    img=np.expand_dims(img,axis=0)
    if(np.max(img)>1):
        img = img/255.0


    predict_x=model.predict(img)
    print(predict_x)
    predict=np.argmax(predict_x,axis=1)
    y=predict[0]

    copy = frame.copy()
    cv2.rectangle(copy, (320, 100), (620, 400), (255, 0, 0), 5)
    cv2.putText(frame, "The Predicted Alphabet : " + str(index[y]), (100, 100), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 4)
    cv2.imshow('frame', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
video.release()
cv2.destroyAllWindows()
```

signtospeech.html

```
{% extends "layout.html" %}

{% block nav %}
<li><a href="/services" class="active">Home</a></li>
<li><a href="/homepage">Sign Out</a></li>
{% endblock %}

{% block body %}
<section class="tm-welcome-section">
      <div class="container tm-position-relative">

         <div class="row tm-welcome-content">
            <br><br>
            <h2 class="white-text tm-handwriting-font tm-welcome-header">
               <img src="{{ url_for('static', filename='img/header-line.png') }}" alt="Line"
                  class="tm-header-line"> Sign On!  <img src="{{ url_for('static',
filename='img/header-line.png') }}" alt="Line"
                  class="tm-header-line"></h2>
            <h2 class="gold-text tm-welcome-header-2">Sign To Speech Translation</h2>
            <p class="gray-text tm-welcome-description"><span
                     class="gold-text" style="font-size:22px; ">Real Time Communication Powered
By AI For Specially Abled</span>
                        <span style="font-size:20px; color:#222222">    The project intends to
create an application which helps the specially challenged people to communicate among them and
the common people. Communication between a person with hearing/speech impairment and a
normal person has always been a challenging task. This application tries to reduce the barrier of
communication by developing an assistive application for specially challenged people.</p>
            </span>
            <br><br><br>
         </div>
         <img src="{{ url_for('static', filename='/Others/SignSpeech.jpg') }}" alt="Table Set"
class="tm-table-set img-responsive" width="100%">
      </div>
   </section>
   <div class="tm-main-section light-gray-bg">
      <div class="container" id="main">
         <section class="tm-section tm-section-margin-bottom-0 row">
            <div class="col-lg-12 tm-section-header-container">
               <h2 class="tm-section-header gold-text tm-handwriting-font">Our Services</h2>
               <div class="tm-hr-container">
                  <hr class="tm-hr">
               </div>
```

```
            </div>
            <div class="col-lg-12 tm-popular-items-container">
               <div class="tm-popular-item">
                  <img src="{{ url_for('static', filename='Others/camera.jpeg') }}" alt="Popular"
class="tm-popular-item-img" width="286px" height="166px">
                  <div class="tm-popular-item-description">
                     <h3 class="tm-handwriting-font tm-popular-item-title"><span
                           class="tm-handwriting-font bigger-first-letter">C</span>lick To
Capture</h3>
                     <hr class="tm-popular-item-hr">
                     <p>Click here to turn on your camera and start showing gestures! </p>
                     <div class="order-now-container">
                        <a href="/predict" class="order-now-link tm-handwriting-font">Proceed
Now</a>
                     </div>
                  </div>
               </div>
               <div class="tm-popular-item">
                  <img src="{{ url_for('static', filename='/Others/home.png') }}" alt="Popular"
class="tm-popular-item-img" width="286px" height="166px">
                  <div class="tm-popular-item-description">
                     <h3 class="tm-handwriting-font tm-popular-item-title"><span
                           class="tm-handwriting-font bigger-first-letter">B</span>ack To Home</h3>
                     <hr class="tm-popular-item-hr">
                     <p>Click here to go back and toggle between modes!</p>
                     <div class="order-now-container">
                        <a href="/services" class="order-now-link tm-handwriting-font">Proceed
Now</a>
                     </div>
                  </div>
               </div>

            </div>
         </section>
      </div>
   </div>
{% endblock %}
```

7.2 Feature 2 - Speech to Sign

Model Building

Speech to Text

In [17]:
```
import speech_recognition as sr
import pyttsx3
import cv2
import time

# Initializing the recognizer
r = sr.Recognizer()

# Function to convert text to speech
def SpeakText(command):
# Initialize the engine
    engine = pyttsx3.init()
    engine.say(command)
    engine.runAndWait()

# Loop infinitely for user to speak


timeout = time.time() + 35   # 35 seconds from now


while(1):
    try:
        with sr.Microphone() as source2:
            r.adjust_for_ambient_noise(source2, duration=0.2)
            audio2 = r.listen(source2)
            MyText = r.recognize_google(audio2)
            MyText = MyText.lower()
            print("Recorded Speech ","-",MyText)
            SpeakText(MyText)
            test = 0
            if test == 5 or time.time() > timeout:
                break
            test = test - 1
    except sr.RequestError as e:
        print("Could not request results; {0}".format(e))
    except sr.UnknownValueError:
        print("Unknown error occurred")
```

```
Recorded Speech  - real time communication
Recorded Speech  - working on ibm project
Recorded Speech  - my voice is being recorded
```

Text to Sign

In [2]:
```
from PIL import Image
import os
import matplotlib.pyplot as plt
while(1):
    try:
            letter=input('Enter alphabet: ')
            char=letter
            if letter.islower():
                char=letter.upper()

            file_path="Gesture_Images/"+char+".jpg"

            if os.path.exists(file_path):
                print("File Path: ", file_path)
                im = Image.open(file_path)
                plt.imshow(im)
                plt.show()

    except KeyboardInterrupt:
        break
```

```
Enter alphabet: H
File Path:  Gesture_Images/H.jpg
```


```
Enter alphabet: g
File Path:  Gesture_Images/G.jpg
```

speechtosign.html

```
{% extends "layout.html" %}

{% block nav %}
<li><a href="/services" class="active">Home</a></li>
<li><a href="/homepage">Sign Out</a></li>
{% endblock %}

{% block body %}

<section class="tm-welcome-section">
      <script type="text/javascript" src="{{ url_for('static', filename='script.js') }}"></script>
      <div class="container tm-position-relative">

         <div class="row tm-welcome-content">
            <br><br>
            <h2 class="white-text tm-handwriting-font tm-welcome-header">
                  <img src="{{ url_for('static', filename='/img/header-line.png') }}" alt="Line"
                     class="tm-header-line"> Sign On!  <img src="{{ url_for('static',
filename='/img/header-line.png') }}" alt="Line"
                     class="tm-header-line"></h2>
               <h2 class="gold-text tm-welcome-header-2">Speech To Sign Translation</h2>
               <p class="gray-text tm-welcome-description"><span
                           class="gold-text" style="font-size:22px; ">Real Time Communication
Powered By AI For Specially Abled</span>
                              <span style="font-size:20px; color:#222222">  The project intends to
create an application which helps the specially challenged people to communicate among them and
the common people. Communication between a person with hearing/speech impairment and a
normal person has always been a challenging task. This application tries to reduce the barrier of
communication by developing an assistive application for specially challenged people.</p>
               </span>
               <br><br><br>
            </div>
            <img src="{{ url_for('static', filename='/Others/SpeechSign.jpg') }}" alt="Table Set"
class="tm-table-set img-responsive" width="100%">
         </div>

   </section>


   <div class="tm-main-section light-gray-bg">
      <div class="container" id="main">
         <section class="tm-section tm-section-margin-bottom-0 row">
```

```
            <div class="col-lg-12 tm-section-header-container">
               <h2 class="tm-section-header gold-text tm-handwriting-font">Our Services</h2>
               <div class="tm-hr-container">
                  <hr class="tm-hr">
               </div>
            </div>

               <div class="col-lg-12 tm-popular-items-container">
               <div class="tm-popular-item">
                  <center>
                  <textarea style="margin-top:50px" id="convert_text" rows="14"
cols="28"></textarea>
              </center>
                  <div class="tm-popular-item-description">
                     <div>
                        <button id="click_to_record" onclick="addImg()" class="btn" style="color:
#c79c60; font-weight: bold;background-color: black; width: 230px; height:50px;
                        border-radius:12px; font-size: 21px;">Capture</button><br/></br>

                        <script>

                           function addImg(){
                              var dimg = document.getElementById('convert_text').value;
                              var letter=dimg.toUpperCase();
                              p="../static/Gesture_Images/"+letter+".jpg";
                              document.getElementById('demo').src=p;

                           }
                        </script>
                     </div>
                  </div>
                  </div>


            <script type="text/javascript" src="{{ url_for('static', filename='/js/script.js') }}"></script>

                  <img id ="demo" src="{{ url_for('static', filename='/Others/asl.jpeg') }}"
height="500px" width="300px" style="width:50%;"/><br><br>

            </div>
         </section>
      </div>
   </div>
{% endblock %}
```

Flask Application

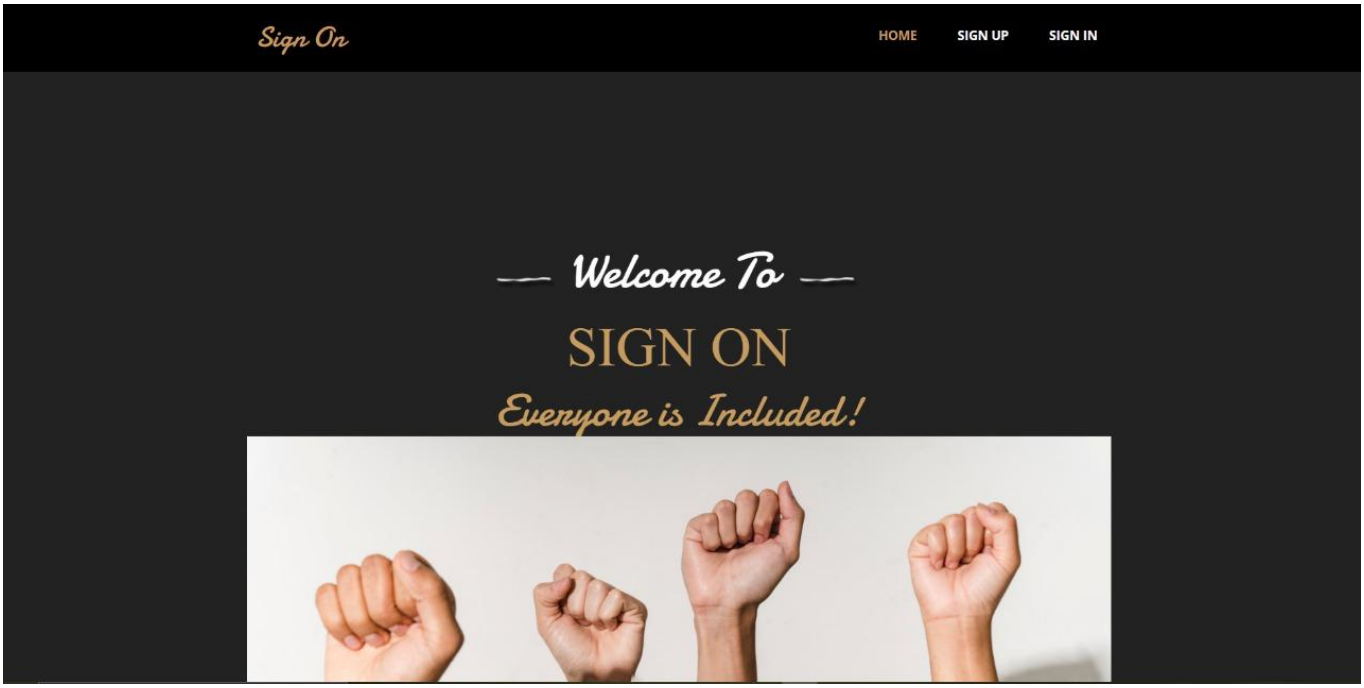
app.py

```
from flask import Flask, render_template, request, redirect, url_for, session,Response
from flask_mysqldb import MySQL
import MySQLdb.cursors
import re
import cv2
from keras.models import load_model
import numpy as np
from gtts import gTTS
import os
from keras.preprocessing import image
from skimage.transform import resize
from playsound import playsound
import pyttsx3
import speech_recognition as sr
from PIL import Image
import matplotlib.pyplot as plt


app = Flask(__name__)

index = ['A','B','C','D','E','F','G','H','I']
model=load_model('model.h5')
video = cv2.VideoCapture(0)
r = sr.Recognizer()
app.secret_key = 'your secret key'

app.config['MYSQL_HOST'] = 'localhost'
app.config['MYSQL_USER'] = 'root'
app.config['MYSQL_PASSWORD'] = 'Prathidp28#'
app.config['MYSQL_DB'] = 'applogin'

mysql = MySQL(app)


def SpeakText(command):
    engine = pyttsx3.init()
    engine.say(command)
    engine.runAndWait()

def generate_frames():
```

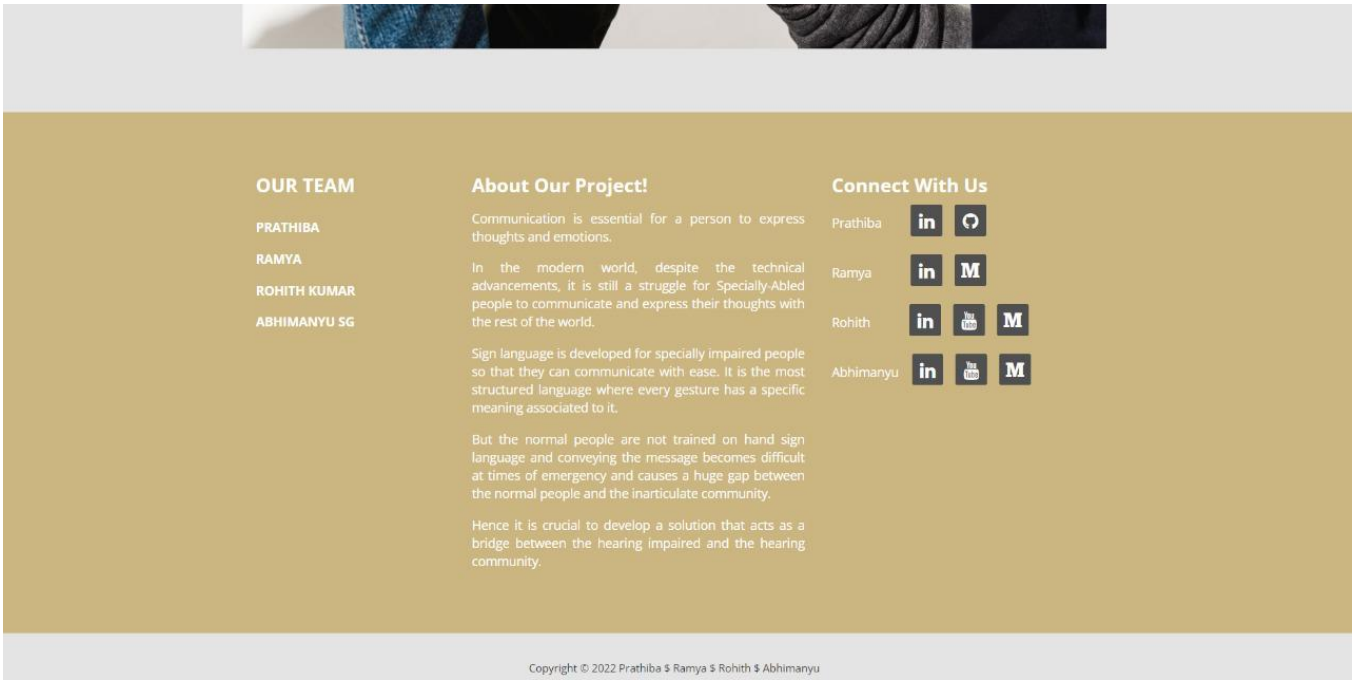
```
    while (video.isOpened()):
        success, frame = video.read()
        cv2.imwrite('frame.jpg', frame)
        img=resize(frame,(64,64,1))
        img=np.expand_dims(img,axis=0)
        if(np.max(img)>1):
            img = img/255.0

        predict_x=model.predict(img)
        print(predict_x)
        predict=np.argmax(predict_x,axis=1)
        y=predict[0]
        SpeakText(index[y])
        copy = frame.copy()
        cv2.rectangle(copy, (320, 100), (620, 400), (255, 0, 0), 5)
        cv2.putText(frame, "The Predicted Alphabet : " + str(index[y]), (100, 100),
cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 0), 4)
        cv2.imshow('frame', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break
        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')


@app.route('/')
@app.route('/homepage', methods =['GET', 'POST'])
def homepage():
    return render_template('homepage.html')


@app.route('/services', methods =['GET', 'POST'])
def services():
    return render_template('services.html')

@app.route('/login', methods =['GET', 'POST'])
def login():
    msg = ''
    if request.method == 'POST' and 'username' in request.form and 'password' in request.form:
        username = request.form['username']
        password = request.form['password']
        cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
        cursor.execute('SELECT * FROM accounts WHERE username = % s AND password = %
s', (username, password, ))
```

```
            account = cursor.fetchone()
            if account:
                session['loggedin'] = True
                session['id'] = account['id']
                session['username'] = account['username']
                msg = 'Logged in successfully !'
                return render_template('services.html', msg = msg)
            else:
                msg = 'Incorrect username / password !'
        return render_template('login.html', msg = msg)

@app.route('/signtospeech', methods =['GET', 'POST'])
def signtospeech():
        return render_template('signtospeech.html')

@app.route('/speechtosign', methods =['GET', 'POST'])
def speechtosign():
        return render_template('speechtosign.html')


@app.route('/register', methods =['GET', 'POST'])
def register():
        msg = ''
        if request.method == 'POST' and 'username' in request.form and 'password' in request.form and
'email' in request.form :
            username = request.form['username']
            password = request.form['password']
            email = request.form['email']
            cursor = mysql.connection.cursor(MySQLdb.cursors.DictCursor)
            cursor.execute('SELECT * FROM accounts WHERE username = % s', (username, ))
            account = cursor.fetchone()
            if account:
                msg = 'Account already exists !'
            elif not re.match(r'[^@]+@[^@]+\.[^@]+', email):
                msg = 'Invalid email address !'
            elif not re.match(r'[A-Za-z0-9]+', username):
                msg = 'Username must contain only characters and numbers !'
            elif not username or not password or not email:
                msg = 'Please fill out the form !'
            else:
                cursor.execute('INSERT INTO accounts VALUES (NULL, % s, % s, % s)', (username,
password, email, ))
                mysql.connection.commit()
                msg = 'You have successfully registered !'
```

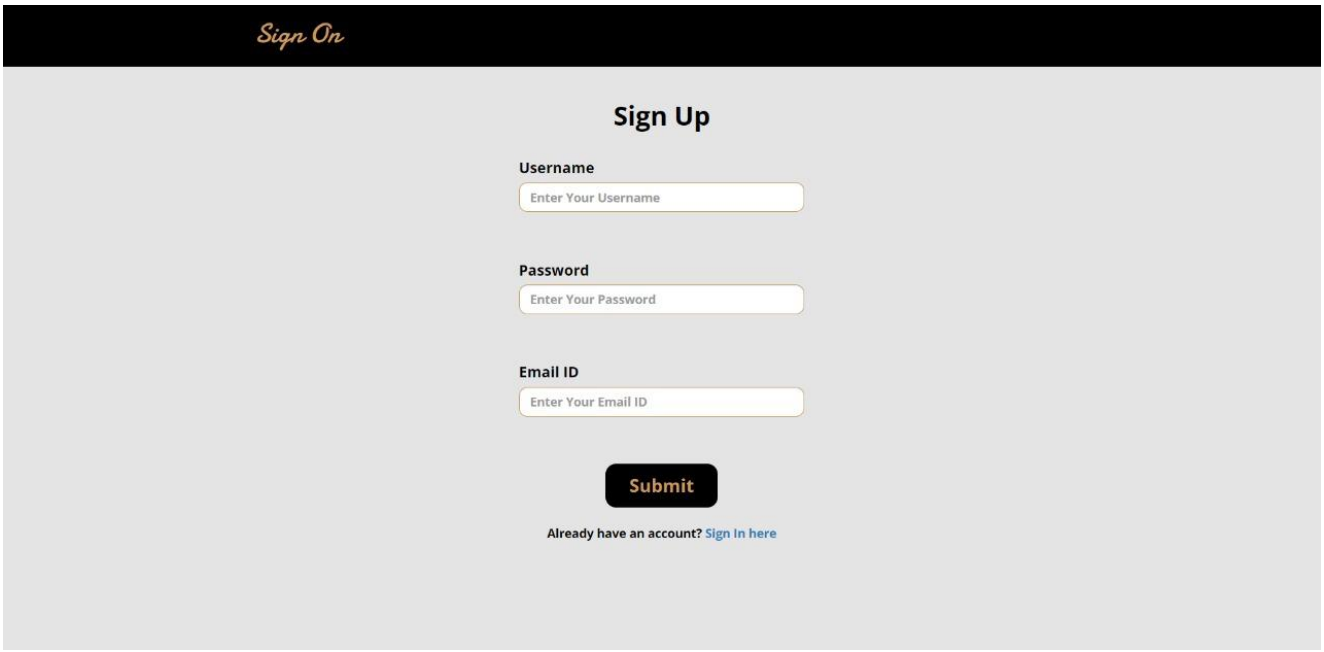
```
    elif request.method == 'POST':
        msg = 'Please fill out the form !'
    return render_template('register.html', msg = msg)

@app.route('/predict', methods=['GET', 'POST'])
def predict():
    return Response(generate_frames(), mimetype='multipart/x-mixed-replace; boundary=frame')
```



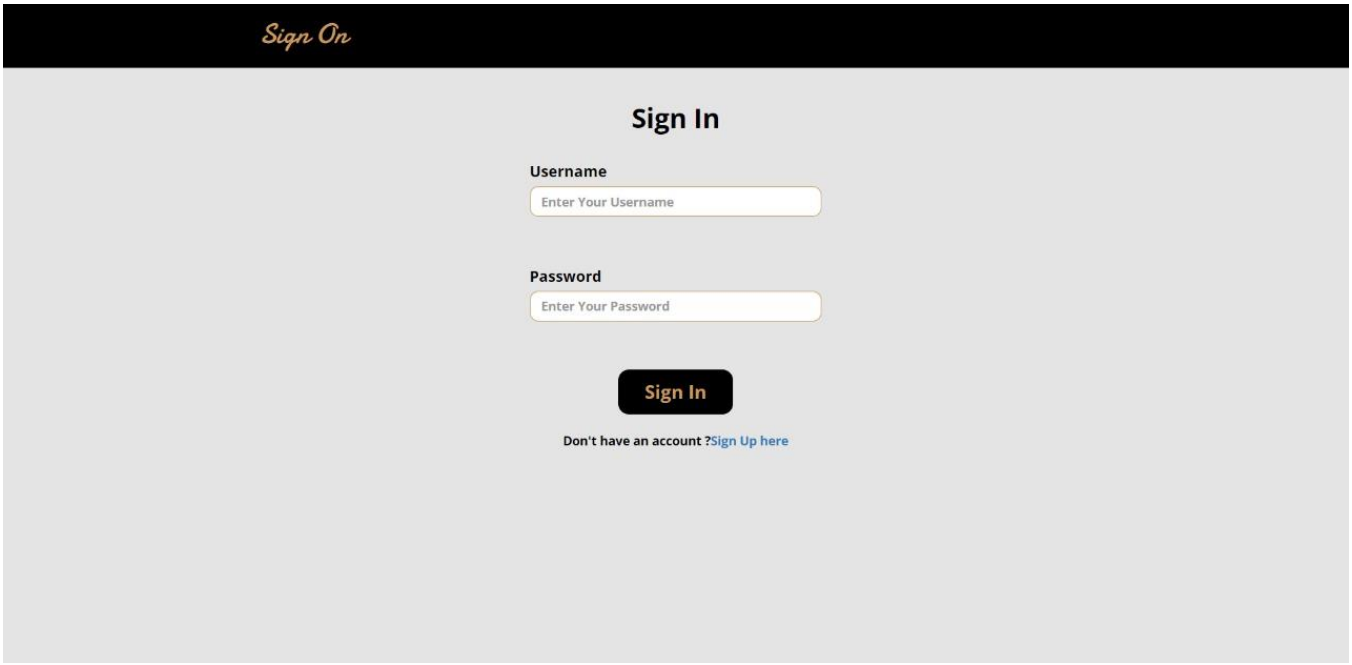
Homepage - 1



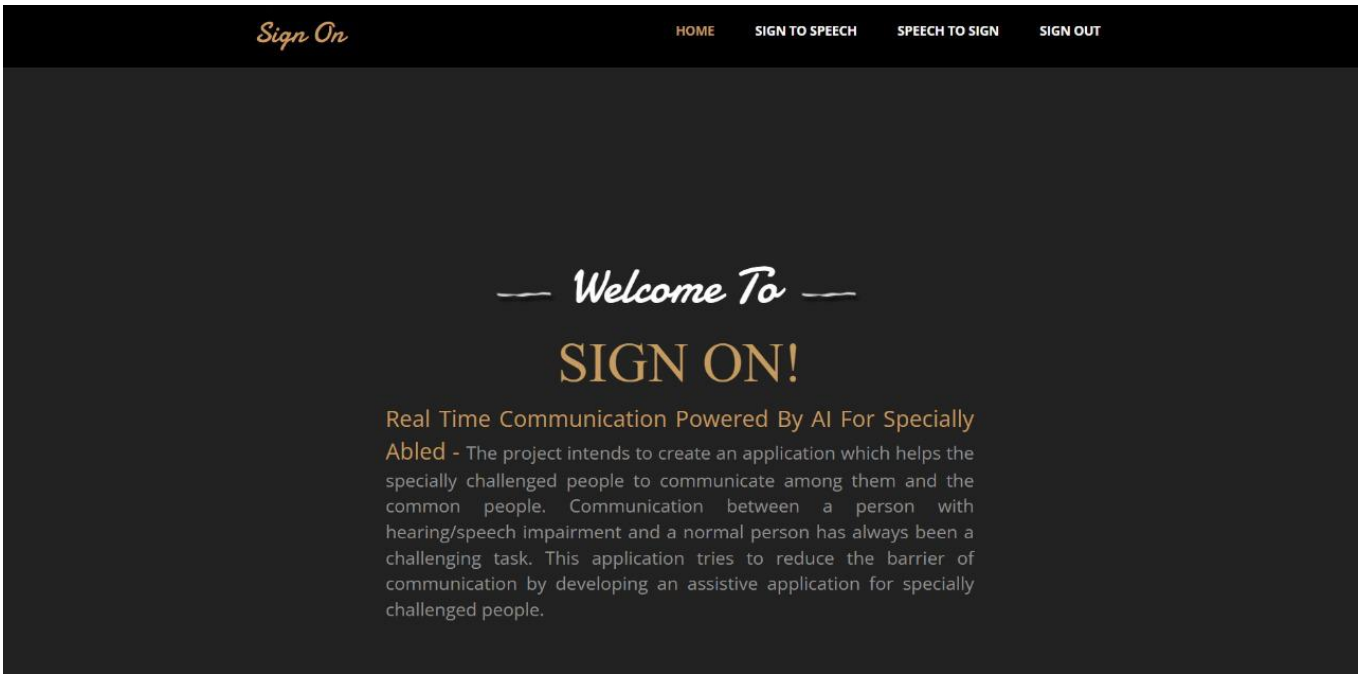
Homepage - 2



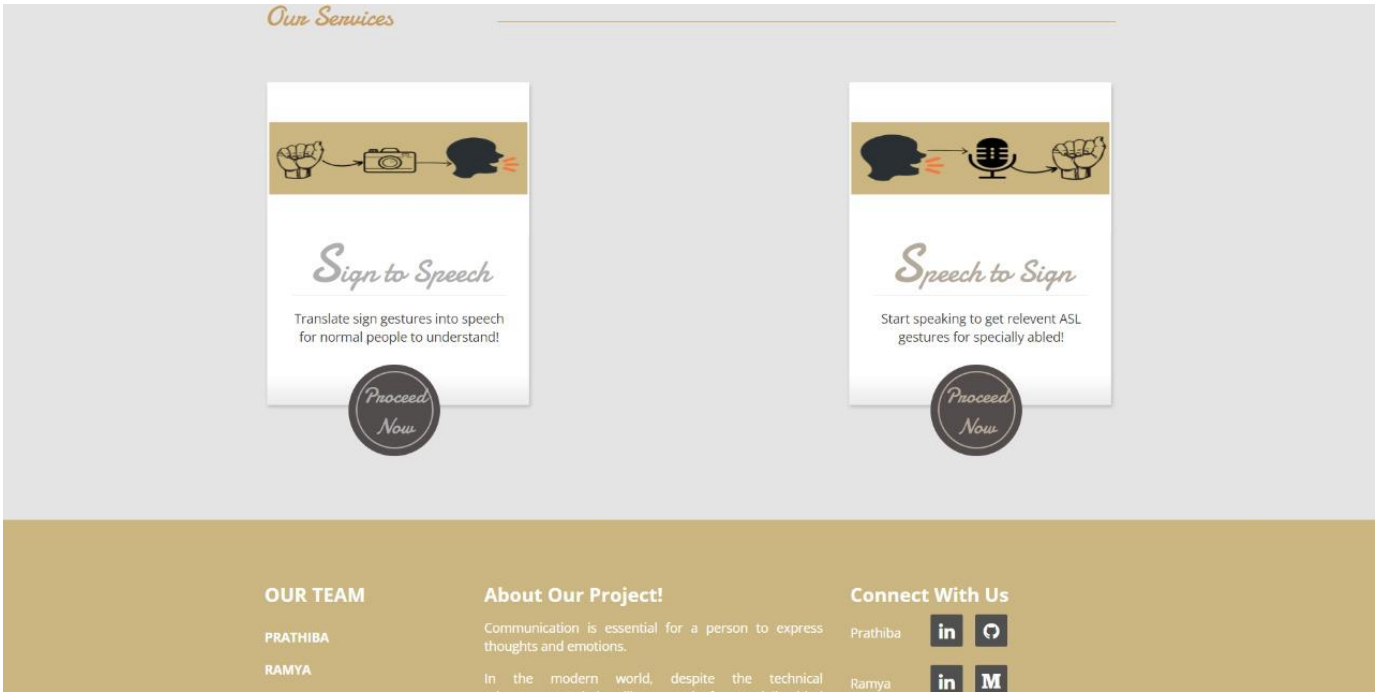
Sign Up page



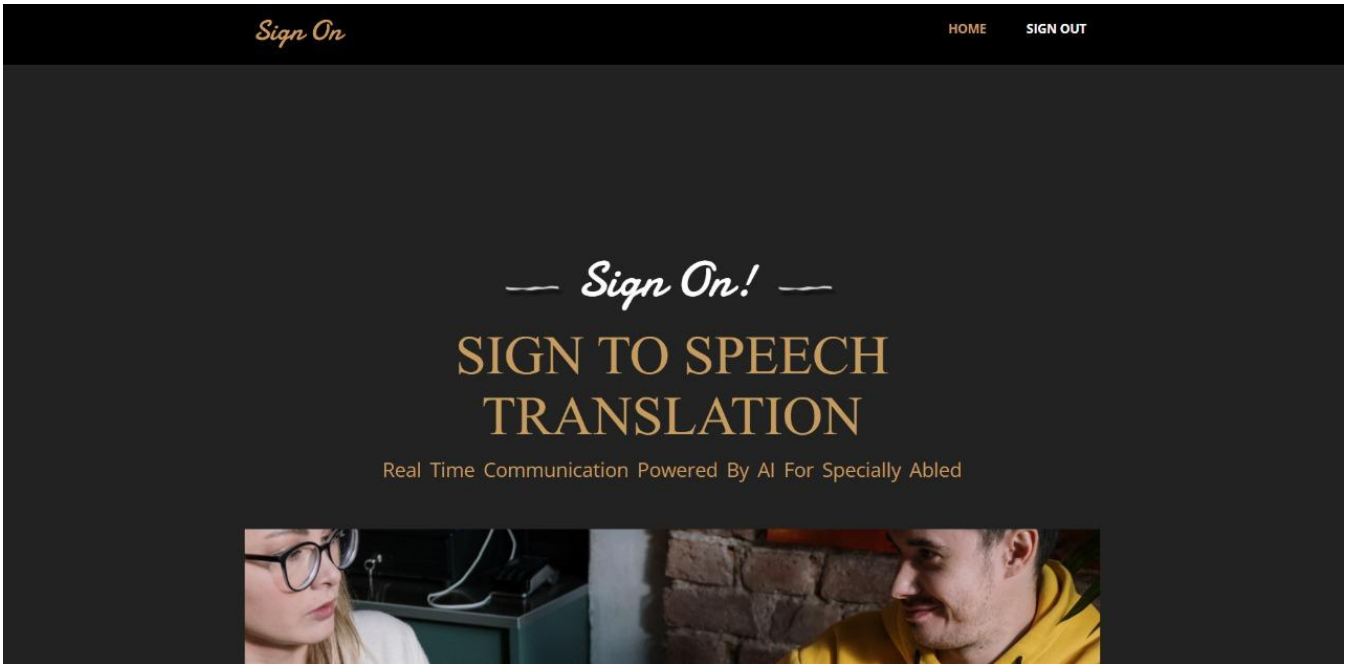
Sign In page



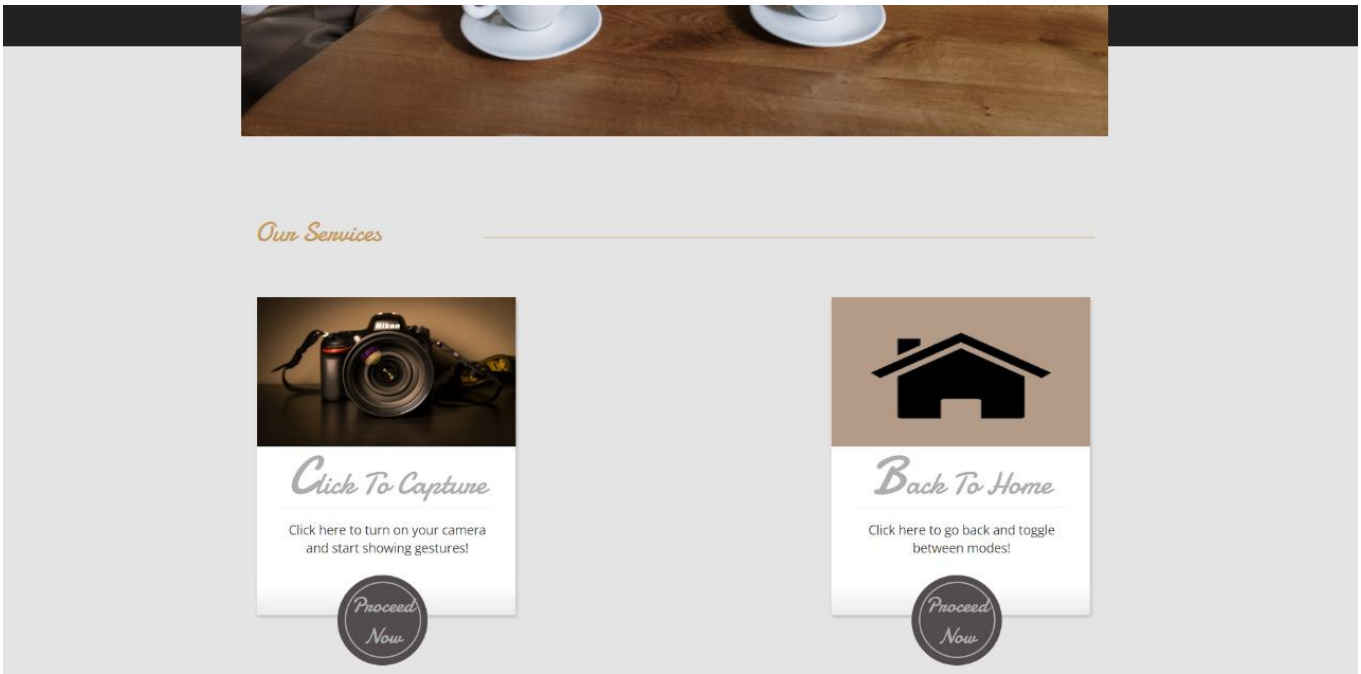
Dashboard - 1



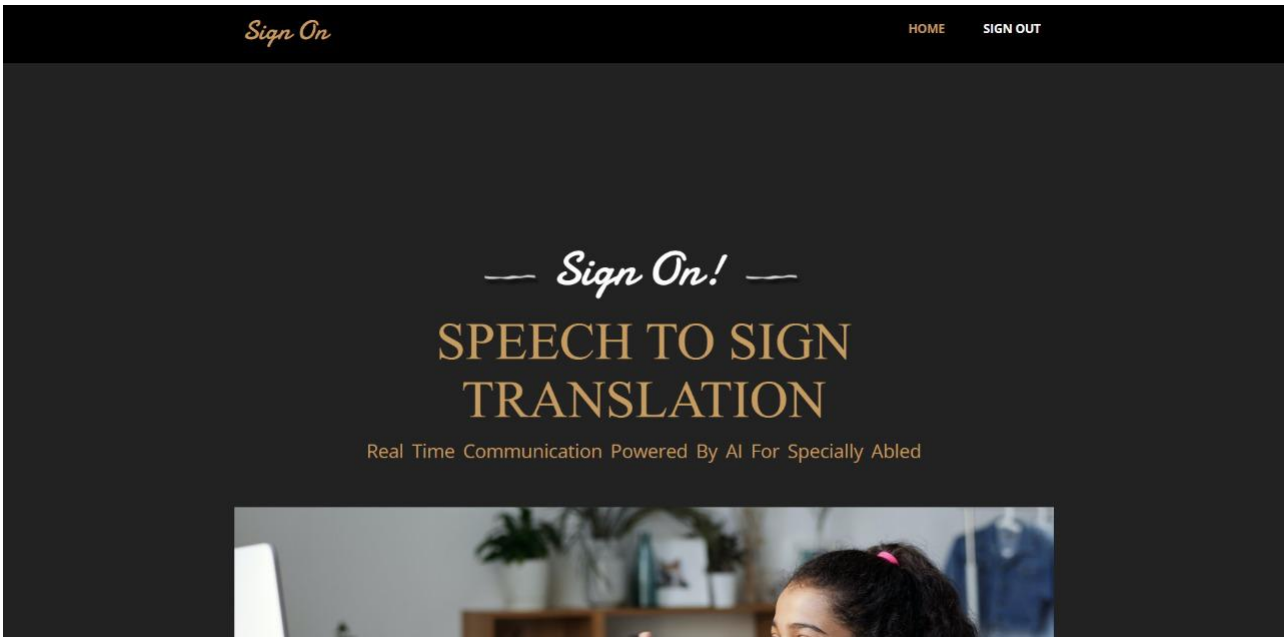
Dashboard - 2



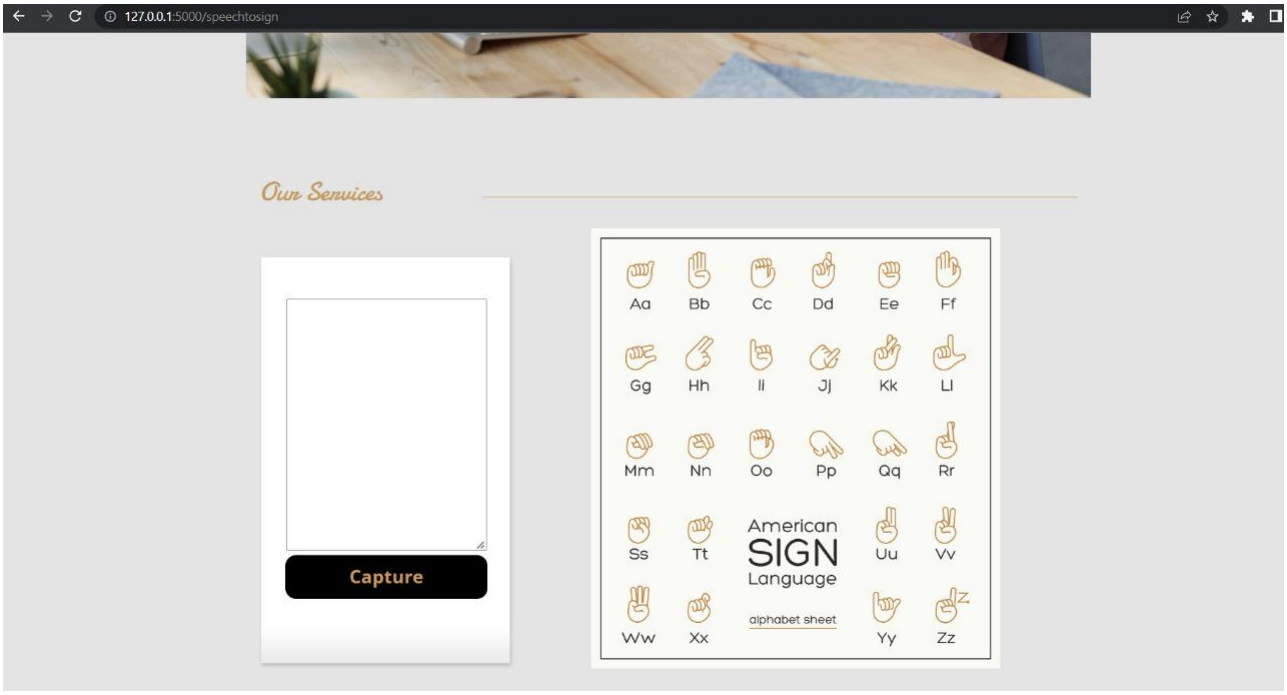
Sign to Speech page



Capture and Back to Homepage options



Speech to Sign page



ASL reference

7.3 Database Schema

```sql
CREATE DATABASE IF NOT EXISTS `applogin` DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
USE `applogin`;

CREATE TABLE IF NOT EXISTS `accounts` (
    `id` int(11) NOT NULL AUTO_INCREMENT,
    `username` varchar(50) NOT NULL,
    `password` varchar(255) NOT NULL,
    `email` varchar(100) NOT NULL,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8;

SELECT * FROM applogin.accounts;
```

CHAPTER 8

Testing

8.1 Test Cases

Feature Type	Component	Test Scenario	Steps to execute	Test Data	Expected Result	Actual Result	Status
Functional	Signup	Verify the user is able to register	(i) Enter details and click signup (ii) Enter Valid username and password	Username: Password:	If entered the user's details shoud be registered in the database	Working as expected	Pass
Functional	Login	Verify that the user is able to login	(i) Enter details and click login (ii) Enter valid username and password	Username: Password:	Entering these the user should be able to login to dashboard	Working as expected	Pass
Functional	Dashboard	Verify whether the user is able to redirect to respective pages	(i) Click on SigntoSpeech button (ii) Click on SpeechtoSign button	Input to the buttons inorder to move to respective pages	By clicking on the two buttons he should be able to move to respective pages	Working as expected	Pass
Functional	SigntoSpeech	Verify the sign language is converted to speech	(i) Give a sign language and get the result (ii) Click on back button	Photos of sign language alphabets	Sign language shown converted to voice.	Working as expected	Pass
Functional	SpeechtoSign	Verify the speech is converted to sign	(i) Give a speech and get the result as signs (ii) Click on the back button	Voice of a person dictating alphabets.	Voice converted to Sign.	Working as expected	Pass

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of digital naturalists project at the time of the release to User Acceptance Testing.

2. Defect Analysis

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

Resolution	Low severity	Medium Severity	High Severity	Subtotal of Bugs
By UI	1	2	2	2
By Functionality	0	2	2	2
Duplicate	0	4	7	2
External	0	0	0	0
Fixed	1	4	4	9
Not Reproduced	0	0	0	0
Skipped	0	0	0	0
Won't fix	0	0	0	0
Totals	2	15	16	29

3. Defect Analysis

Section	Total Cases	Not Tested	Fail	Pass
User Interface	2	0	0	2
Flask application	4	0	0	4
Exception Reporting	1	0	0	1
Final Report Output	2	0	0	2

CHAPTER 9

Results

9.1 Performance Metrics

Parameter	Values	Screenshot
Model Summary	Total params: 9,273,305 Trainable params: 9,273,305 Non-trainable params: 0	`model.summary()` Model: "sequential" Layer (type) — Output Shape — Param # conv2d (Conv2D) — (None, 62, 62, 32) — 896 max_pooling2d (MaxPooling2D) — (None, 31, 31, 32) — 0 flatten (Flatten) — (None, 30752) — 0 dense (Dense) — (None, 300) — 9225900 dense_1 (Dense) — (None, 150) — 45150 dense_2 (Dense) — (None, 9) — 1359 Total params: 9,273,305 Trainable params: 9,273,305 Non-trainable params: 0

Accuracy	Training Accuracy - 0.9995 Validation Accuracy - 0.977	Epoch 9/10 18/18 [==============================] - 41s 2s/step - loss: 0.0045 - accuracy: 0.9994 - val_loss: 0.2632 - val_accuracy: 0.9773 Epoch 10/10 18/18 [==============================] - 38s 2s/step - loss: 0.0035 - accuracy: 0.9995 - val_loss: 0.2710 - val_accuracy: 0.9773
Confidence Score (Only Yolo Projects)	Class Detected - Confidence Score -	Not Applicable

CHAPTER 10

ADVANTAGES

- Two-way translation (sign to speech and vice versa) is possible.
- The application can identify and translate the live and moving images.
- The proposed system ensures the easy translation of sign language to English. Even people who are unaware of the sign language can use the application easily.
- No high-end devices are required to use the application and is compatible with all operating systems and browses.
- The proposed system is user friendly and makes the life of the person with disability easy.

DISADVANTAGES

- Since it is a web page-based system, it does require internet connectivity which can be inconvenient at times.
- It would have been convenient if it is application based.
- There is a slight chance for a mistranslation. It is difficult to build a perfect system.
- Sign language customization feature is not available.
- The person should have a fundamental knowledge on the operation of computer systems and devices

CONCLUSION

The proposed system is easy to implement as there is no complex feature calculation. This system provides us with high gesture recognition rate with accuracy 90% within minimum time. The system aims to lower the communication gap between deaf people and normal world, since it facilitates dual communications. The projected methodology interprets hand gestures into speech and vice versa. The system overcomes disadvantages of previous existing system and improves their manner. With this project the deaf-mute people can use the hand gestures to perform sign language and it will be converted into speech with accuracy 93%; and the speech of normal person is converted into hand gesture, so the communication between them can take place easily.

FUTURE SCOPE

- The application can be extended to receive user feedback. Improvements in the existing application features or the addition of new features can be carried based on the feedback and ratings.
- The implementation of our model for other sign languages such as Indian Sign Language (ISL) and Russian Sign Language (RSL), as it is currently limited to American Sign Language (ASL).
- Further training the neural network to efficiently recognize symbols involving two hands.
- This application can be enhanced for dynamic gesture recognition wherein LSTM

networks can be employed.

- Users can be given the privilege to specify additional gestures for recognition.

APPENDIX

Source Code

https://github.com/IBM-EPBL/IBM-Project-18089-1659679109/tree/main/Final%20Deliverables/Final%20Code

GitHub

https://github.com/IBM-EPBL/IBM-Project-18089-1659679109

Project Demo Link

https://github.com/IBM-EPBL/IBM-Project-18089-1659679109/blob/main/Final%20Deliverables/Demonstration%20Video%20Link/demo_link.txt