CONTENTS

1. INTRODUCTION

1.1 Project Overview

In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Real-time communications (RTC) is any mode of telecommunications in which all users can exchange information instantly.

Communication plays a significant role in making the world better place. It creates a bonding and relations among the people.

1.2 Purpose

The project aims 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 using the convolutional neural network.

An app is built which enables the deaf and dumb people to convey their information using signs which is converted to human understandable language and output is given as speech.

2. LITERATURE SURVEY

2.1 Existing problem

Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language.

Only specially abled people are taught sign language and the common person is unaware its working causing a communication gap. Under emergency situations, it is even more difficult for specially abled people to get help. Non-Emergency normal environments can also be hard for them to navigate needing special assistance.

2.2 References

1. Upendran, S., and Thamizharasi, A., "American Sign Language interpreter system for deaf and dumb individuals", In the Proceedings of the International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), pp. 1477-1481, 2014
2. Lotti, F.,Tiezzi, P., Vassura, G.,Biagiotti, L., and Melchiorri, C., "UBH 3: an anthropomorphic hand with simplified endo-skeletal structure and soft continuous fingerpads", In Proceedings IEEE International Conference on Robotics and Automation, 2004 (ICRA'04), Vol.5, pp. 4736-474, IEEE, 2004.
3. Rajamohan, A., Hemavathy, R., andDhanalakshmi, M., "Deaf-Mute Communication Interpreter", International Journal of Scientific Engineering and Technology, Vol.2, No.5, pp.336-341, 2013.

4. Verma, P., Shimi S. L. and Priyadarshani, R., "Design of Communication Interpreter for Deaf and Dumb Person", Vol.4, no.1, 2013.

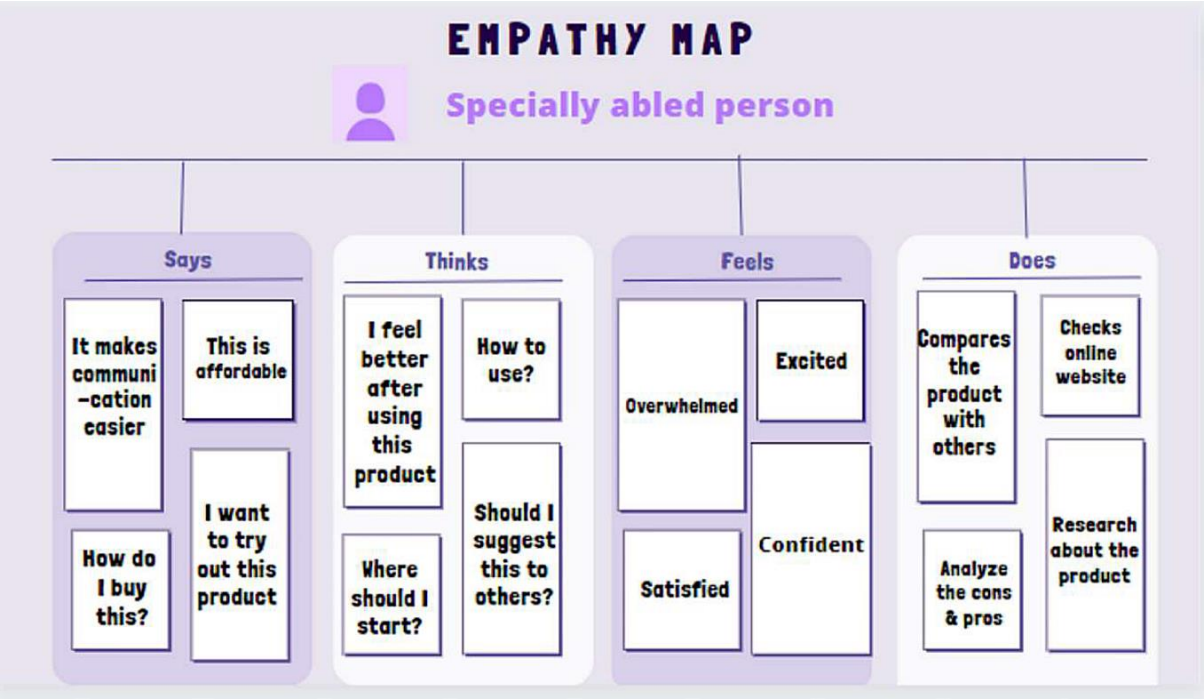
2.3 Problem Statement Definition

Only specially abled people are taught sign language and the common person is unaware its working causing a communication gap. Under emergency situations, it is even more difficult for specially abled people to get help. Non-Emergency normal environments can also be hard for them to navigate needing special assistance.

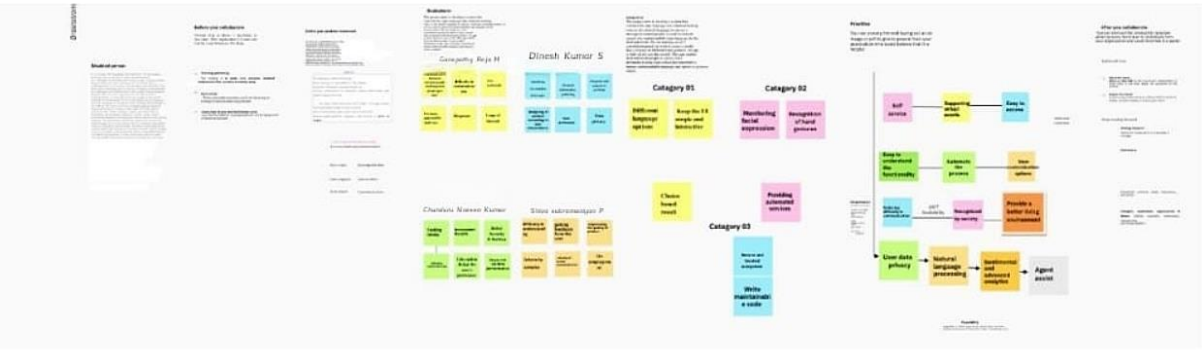
Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

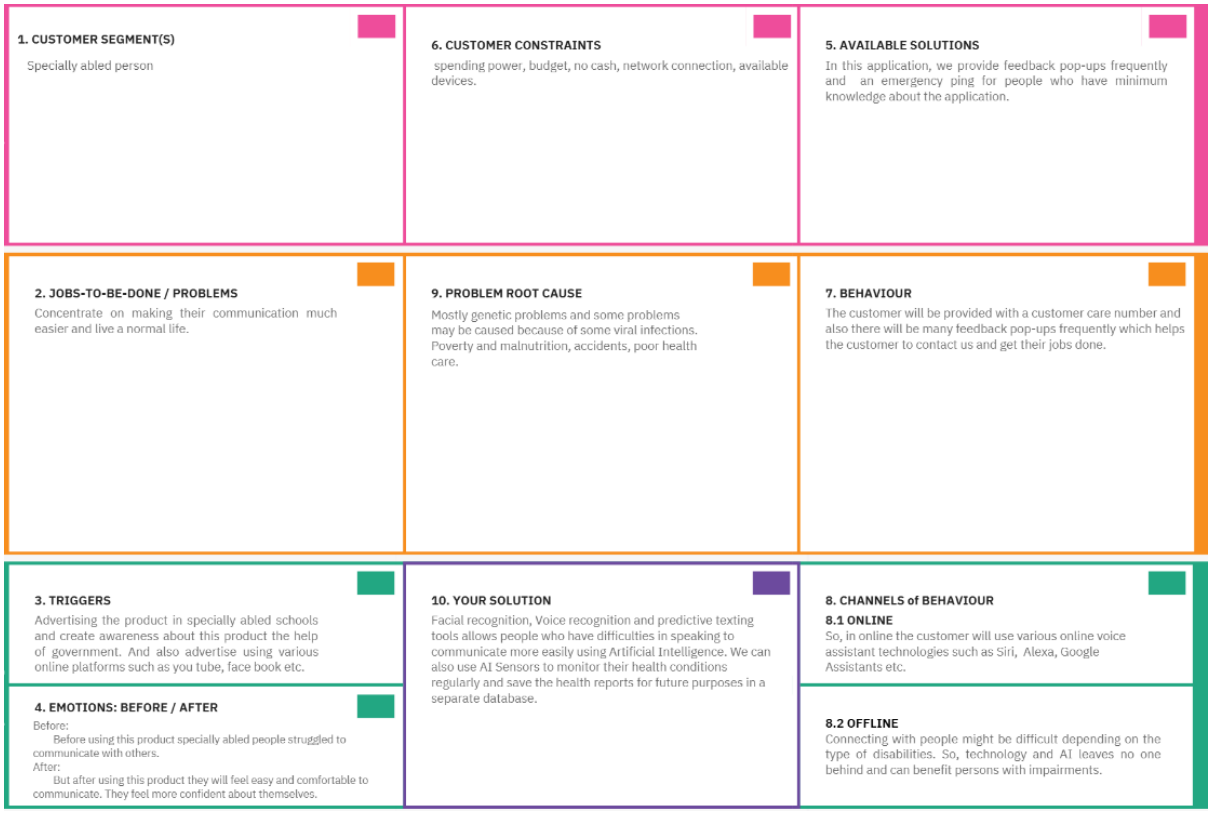


3.3 Proposed Solution

SNO	Parameter	Description
1.	Problem Statement(Problem to be sloved)	In our society, we have people with disabilities. The technology is developing day by day but no significant developments are undertaken for the betterment of these people. Communications between deaf-mute and a normal person has always been a challenging task. It is very difficult for mute people to convey their message to normal people. Since normal people are not trained on hand sign language. In emergency times conveying their message is very difficult. The human hand has remained a popular choice to convey information in situations where other forms like speech cannot be used. Voice Conversion System with Hand Gesture Recognition and translation will be very useful to have a proper conversation between a normal person and an impaired person in any language
2.	Idea / Solution description	The project aims 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	Building mobile tools with data isn't as easy as importing an XML feed of your latest headlines. But if you're going to spend thousands of dollars developing a mobile app anyway, you might as well spend a little more to build a real application that helps solve problems and makes advertisers take notice.

4.	Social Impact / Customer Satisfaction	These apps are using only for solve the problems. There are many different types of disabilities, and there are also many different ways in which people may use AI. The use of artificial intelligence is a boon for specially-abled people. Technology had opened up new opportunities and created jobs where none had existed before, such as speech to text software that helped one woman find her voice after she was paralysed in an accident. Artificial Intelligence can help those with disabilities accomplish tasks they never thought possible; here are just a few ways we've seen AI technology impact lives: Facial recognition and predictive texting tools
5.	Business Model (Revenue Model)	Building mobile tools with data isn't as easy as importing an XML feed of your latest headlines. But if you're going to spend thousands of dollars developing a mobile app anyway, you might as well spend a little more to build a real application that helps solve problems and makes advertisers take notice
6.	Scalability of the Solution	With the help of machine tasks that usually requires human intelligence, such as voice and speech synthesis, visual perception, predictive text functionality, judgement, and a variety of other tasks, AI can assist individuals with disabilities by making a significant distinction in their ability.

3.4 Problem Solution



4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

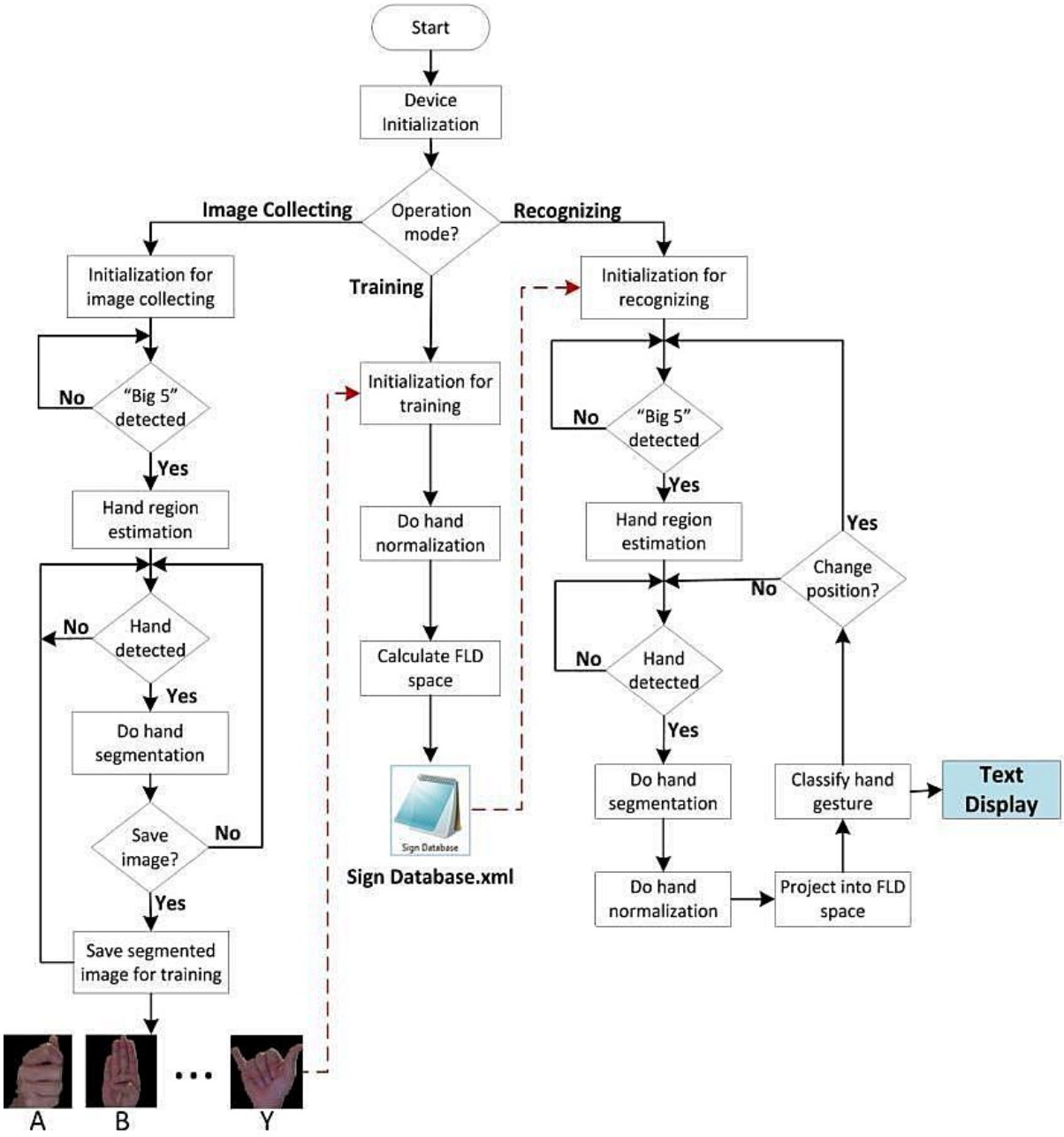
FR NO.	Functional Requirement(Epic)	Sub Requirement(Story/Sub-Task)
FR-1	User Registration	Registration through From Registration through Gmail Registration through Linked IN
FR-2	User Confirmation	Confirmation Via Email Confirmation Via OTP
FR-3	Uploading Image	Upload image through camera or Upload image through gallery
FR-4	Templates usage During any Emergencies	Select emergency templates icon to pass the message quickly
FR-5	Text to Speech	Convert Respective text to sign Language

4.2 Non Functional Requirement

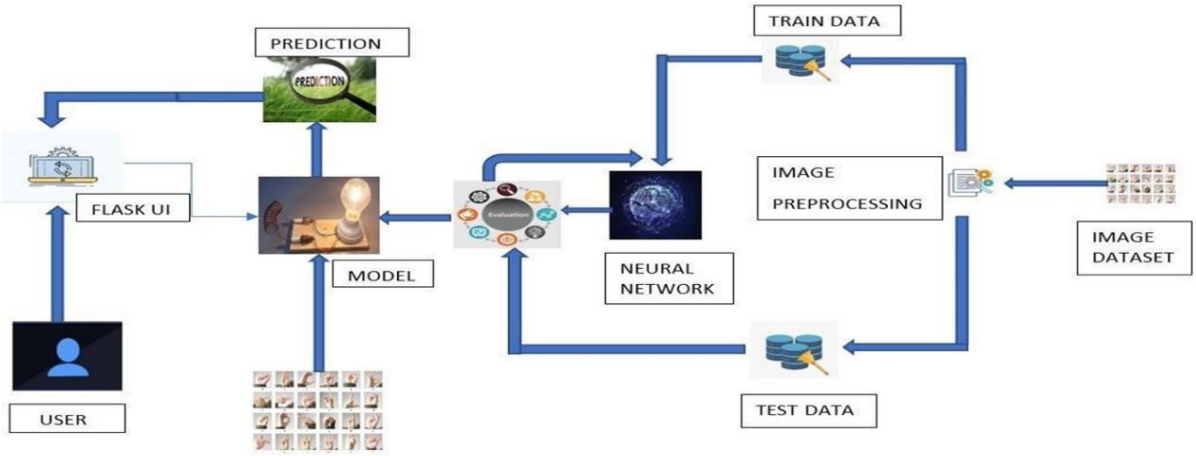
FR NO.	Non- Functional Requirement	Description
NFR-1	Usability	user can easily upload the image and this app is designed in a way where user can easily find some predefined templates or layouts.
NFR-2	Security	user should sign in into the application. so any unauthorized access will be avoided at the most.
NFR-3	Reliability	This application has robust fault tolerance and even if there is an error, it recuperates swiftly.
NFR-4	Performance	Utilizing the CNN model, the gestures made by the user is predicted by the application with a higher accuracy.
NFR-5	Availability	This is application is effortless and is accessible to all users. The predefined formats or layouts are understandable and makes it easier for accessing it.
NFR-6	Scalability	Highly scalable which uses gesture recognition and is a hands on model which is used in many other applications.

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 (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming the password	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email &click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register via some third party's link	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can type manually and also can used saved login credentials	High	Sprint-1
	Dashboard	USN-6	As a customer, I can get all services and help through the dashboard	I can access my dashboard and change profile	Medium	Sprint-2
Customer (Web user)	Registration	USN-7	As a customer, I could able to login through registered phone number by using OTP instead of Gmail	I could able to register& login via phone number to access my account	High	Sprint-2
Customer Care Executive	Service	USN-8	Can avail the service by calling customer care or reaching through E-mail.	Can avail the service by calling customer care or reaching through E- mail.	Medium	Sprint-1
Administrator	Sign up	USN-9	Customer have to sign-up to use these things and all	Have to enter valid credentials.	High	Sprint-1
	Enrollment	USN-10	The customer can avail all services once he/she enrolled.	As customer it is quite enchanting.	Medium	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Milestone	Function (Epic)	Milestone Story Number	Story / Task
Milestone 1	Data collection	M1	Collected dataset for building our project and created two folders: testing and training.
Milestone 2	Image preprocessing	M2	Importing image data generator libraries and applying image data generator functionality to train the test set.
Milestone 3	Model building	M3	Importing the model building libraries, Initializing the model, Adding Convolution layers, Adding the Pooling layers, Adding the Flatten layers, Adding Dense layers, Compiling the model Fit and Save the model.
Milestone 4	Testing the model	M4	Import the packages first. Save the model and Load the test image, preprocess it and predict it.
Milestone 5	Application layer	M5	Build the flask application and HTML pages.
Milestone 6	Train CNN model	M6	Register for IBM Cloud and train Image Classification Model.
Milestone 7	Final result	M7	Final output was verified.

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	GANAPATHY RAJA CHUNDURU NAVEEN KUMAR SHIVA SUBRAMANIYAN DINESH KUMAR
	Login	USN-2	As a user, I can log into the application by entering email & password	5	High	GANAPATHY RAJA CHUNDURU NAVEEN KUMAR SHIVA SUBRAMANIYAN DINESH KUMAR
	Data Collection	USN-3	Collecting Dataset	5	High	GANAPATHY RAJA CHUNDURU NAVEEN KUMAR SHIVA SUBRAMANIYAN DINESH KUMAR
	Image pre-processing	USN-4	Perform pre-processing techniques on the dataset	5	High	GANAPATHY RAJA CHUNDURU NAVEEN KUMAR SHIVA SUBRAMANIYAN DINESH KUMAR
Sprint-2	Model Building	USN-5	Model initialization with required layers	5	High	GANAPATHY RAJA CHUNDURU NAVEEN KUMAR SHIVA SUBRAMANIYAN DINESH KUMAR
	Training	USN-6	Training the image classification model using CNN	5	High	GANAPATHY RAJA CHUNDURU NAVEEN KUMAR SHIVA SUBRAMANIYAN DINESH KUMAR
Sprint-3	Testing	USN-7	Testing the model's performance	5	High	GANAPATHY RAJA CHUNDURU NAVEEN KUMAR SHIVA SUBRAMANIYAN DINESH KUMAR
Sprint-4	Deployment of model in web/app	USN-8	Converting text to speech API	5	High	GANAPATHY RAJA CHUNDURU NAVEEN KUMAR SHIVA SUBRAMANIYAN DINESH KUMAR

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Model Building

Importing The Required Model Building Libraries

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

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

In [ ]:
```
#training Dataset
x_train=train_datagen.flow_from_directory('/content/training_set',target_size=(64,64),batch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
x_test=test_datagen.flow_from_directory('/content/test_set',target_size=(64,64),batch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
```

```
Found 15130 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

In [ ]:
```
import tensorflow as tf
import os
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout, MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
import matplotlib.pyplot as plt
import IPython.display as display
from PIL import Image
import pathlib
```

In [ ]:
```
!unzip /content/test_set.zip
```

```
Archive:  /content/test_set.zip
   creating: test_set/A/
  inflating: test_set/A/1.png
  inflating: test_set/A/10.png
 extracting: test_set/A/100.png
  inflating: test_set/A/101.png
  inflating: test_set/A/102.png
 extracting: test_set/A/103.png
  inflating: test_set/A/104.png
  inflating: test_set/A/105.png
  inflating: test_set/A/106.png
  inflating: test_set/A/107.png
  inflating: test_set/A/108.png
  inflating: test_set/A/109.png
  inflating: test_set/A/11.png
  inflating: test_set/A/110.png
  inflating: test_set/A/111.png
```

Applying ImageDataGenerator Functionality to Trainset and Testset

In [ ]:
```
x_train=train_datagen.flow_from_directory('/content/training_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="grayscale")
```

```
Found 15130 images belonging to 9 classes.
```

In [ ]:
```
x_test=test_datagen.flow_from_directory('/content/test_set',target_size=(64,64),batch_size=300,class_mode='categorical',color_mode="grayscale")
```

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

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

```
Len x-train :  51
Len x-test :  8
```

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

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

In [ ]:
```
datagen=ImageDataGenerator(rescale=1./255.,validation_split=0.25)
```

Model Creation

In [ ]:
```
# Importing Libraries

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

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

In [ ]:
```
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale = 1./255, shear_range=0.2, zoom_range=0.2,horizontal_flip=True,vertical_flip=False)
test_datagen = ImageDataGenerator(rescale=1./255)
```

In [ ]:
```
x_train = train_datagen.flow_from_directory("/content/training_set", target_size=(64,64),batch_size=100,
                                            class_mode='categorical', color_mode ="grayscale")
```

```
Found 15130 images belonging to 9 classes.
```

In [ ]:
```
x_test = test_datagen.flow_from_directory("/content/test_set", target_size=(64,64),batch_size=100,
                                          class_mode='categorical', color_mode ="grayscale")
```

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

In [ ]:
```
len(x_train)
```

Out[ ]:
```
152
```

In [ ]:
```
len(x_test)
```

Out[ ]:
```
23
```

In [ ]:
```
x_train.class_indices
```

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

MODEL BUILDING

In [ ]:
```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from tensorflow.keras.layers import Conv2D, MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
```

In [ ]:
```
#Creating the model
model=Sequential()
#Adding the layers
model.add(Convolution2D(32,(3,3), input_shape=(64,64,1), activation = 'relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())

#adding hidden layers
model.add(Dense(400, activation='relu'))
model.add(Dense(200, activation='relu'))
model.add(Dense(100, activation='relu'))

#Adding the output layer
model.add(Dense(9, activation='softmax'))
```

In [ ]:
```
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

In [ ]:
```
model.fit_generator(x_train, steps_per_epoch=30, epochs=10, validation_data=x_test,validation_steps=50)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future versi
on. Please use `Model.fit`, which supports generators.
  """Entry point for launching an IPython kernel.
Epoch 1/10
30/30 [==============================] - ETA: 0s - loss: 1.0847 - accuracy: 0.6197
WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch
* epochs` batches (in this case, 50 batches). You may need to use the repeat() function when building your dataset.
30/30 [==============================] - 19s 570ms/step - loss: 1.0847 - accuracy: 0.6197 - val_loss: 0.4876 - val_accuracy: 0.8582
Epoch 2/10
30/30 [==============================] - 13s 435ms/step - loss: 0.2593 - accuracy: 0.9218
Epoch 3/10
30/30 [==============================] - 14s 468ms/step - loss: 0.1441 - accuracy: 0.9583
Epoch 4/10
30/30 [==============================] - 13s 441ms/step - loss: 0.1015 - accuracy: 0.9673
Epoch 5/10
30/30 [==============================] - 13s 442ms/step - loss: 0.0687 - accuracy: 0.9780
Epoch 6/10
30/30 [==============================] - 15s 494ms/step - loss: 0.0438 - accuracy: 0.9863
Epoch 7/10
30/30 [==============================] - 13s 446ms/step - loss: 0.0549 - accuracy: 0.9820
Epoch 8/10
30/30 [==============================] - 13s 442ms/step - loss: 0.0412 - accuracy: 0.9887
Epoch 9/10
30/30 [==============================] - 13s 446ms/step - loss: 0.0276 - accuracy: 0.9920
Epoch 10/10
30/30 [==============================] - 15s 487ms/step - loss: 0.0165 - accuracy: 0.9959
```

Out[ ]:

Add The Convolution Layer

In [ ]:
```
model.save('Real_time.h5')
```

In [ ]:
```
a=len(x_train)
b=len(x_test)
```

In [ ]:
```
print(a)
```

```
152
```

In [ ]:
```
print(b)
```

```
23
```

In [ ]:
```
#create model
model=Sequential()
```

In [ ]:
```
model.add(Convolution2D(32,(3,3),input_shape=(64,64,1),activation='relu'))
```

Add the Pooling Layer

In [ ]:
```
model.add(MaxPooling2D(pool_size=(2,2)))
```

Add the Flatter Layer

In [ ]:
```
model.add(Flatten())
```

The_Dense_Layers

In [ ]:
```
#1st hidden layer
model.add(Dense(units=512,activation='relu'))
#2nd hidden layer
model.add(Dense(units=261,activation='relu'))
```

In [ ]:
```
#output layer
model.add(Dense(units=9,activation='softmax'))
```

Compile the Model

In [ ]:
```
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])
```

Fit and Save the Model

In [ ]:
```
model.save('aslpng2.h5')
```

Loading the Dataset& Image Data Generation

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

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

In [ ]:
```
#training Dataset
x_train=train_datagen.flow_from_directory('/content/training_set',target_size=(64,64),batch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
x_test=test_datagen.flow_from_directory('/content/test_set',target_size=(64,64),batch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
```

```
Found 15130 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

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

```
len x-train : 76
len x-test : 12
```

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

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

Model Creation

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

In [ ]:
```
#create model
model=Sequential()
```

In [ ]:
```
#Adding Layers
model.add(Convolution2D(32,(3,3),input_shape=(64,64,1),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())

#Adding  hidden layer
model.add(Dense(units=512,activation='relu'))
model.add(Dense(units=261,activation='relu'))

#output layer
model.add(Dense(units=9,activation='softmax'))
```

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

In [ ]:
```
#Fitting the model
model.fit_generator(x_train, steps_per_epoch=30, epochs=10, validation_data=x_test,validation_steps=50)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future versi
on. Please use `Model.fit`, which supports generators.

Epoch 1/10
30/30 [==============================] - ETA: 0s - loss: 0.6843 - accuracy: 0.7643
WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch
* epochs` batches (in this case, 50 batches). You may need to use the repeat() function when building your dataset.
30/30 [==============================] - 39s 1s/step - loss: 0.6843 - accuracy: 0.7643 - val_loss: 0.3408 - val_accuracy: 0.9071
Epoch 2/10
30/30 [==============================] - 35s 1s/step - loss: 0.1345 - accuracy: 0.9608
Epoch 3/10
30/30 [==============================] - 33s 1s/step - loss: 0.0763 - accuracy: 0.9798
Epoch 4/10
30/30 [==============================] - 32s 1s/step - loss: 0.0469 - accuracy: 0.9862
Epoch 5/10
30/30 [==============================] - 32s 1s/step - loss: 0.0304 - accuracy: 0.9922
Epoch 6/10
30/30 [==============================] - 33s 1s/step - loss: 0.0225 - accuracy: 0.9935
Epoch 7/10
30/30 [==============================] - 35s 1s/step - loss: 0.0117 - accuracy: 0.9968
Epoch 8/10
30/30 [==============================] - 32s 1s/step - loss: 0.0108 - accuracy: 0.9967
Epoch 9/10
30/30 [==============================] - 32s 1s/step - loss: 0.0168 - accuracy: 0.9947
Epoch 10/10
30/30 [==============================] - 33s 1s/step - loss: 0.0178 - accuracy: 0.9952
```

8. TESTING

8.1 TEST CASES

Loading the Dataset& Image Data Generation

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

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

In [ ]:
```
#training Dataset
x_train=train_datagen.flow_from_directory('/content/training_set',target_size=(64,64),batch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
x_test=test_datagen.flow_from_directory('/content/test_set',target_size=(64,64),batch_size=200,
                                          class_mode='categorical',color_mode="grayscale")
```

```
Found 15130 images belonging to 9 classes.
Found 2250 images belonging to 9 classes.
```

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

```
len x-train : 76
len x-test : 12
```

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

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

Model Creation

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

In [ ]:
```
#create model
model=Sequential()
```

In [ ]:
```
#Adding Layers
model.add(Convolution2D(32,(3,3),input_shape=(64,64,1),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())

#Adding  hidden layer
model.add(Dense(units=512,activation='relu'))
model.add(Dense(units=261,activation='relu'))

#output layer
model.add(Dense(units=9,activation='softmax'))
```

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

In [ ]:
```
#Fitting the model
model.fit_generator(x_train, steps_per_epoch=30, epochs=10, validation_data=x_test,validation_steps=50)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future versio
n. Please use `Model.fit`, which supports generators.
Epoch 1/10
30/30 [==============================] - ETA: 0s - loss: 0.6843 - accuracy: 0.7643
WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch *
epochs` batches (in this case, 50 batches). You may need to use the repeat() function when building your dataset.
30/30 [==============================] - 39s 1s/step - loss: 0.6843 - accuracy: 0.7643 - val_loss: 0.3408 - val_accuracy: 0.9071
Epoch 2/10
30/30 [==============================] - 35s 1s/step - loss: 0.1345 - accuracy: 0.9608
Epoch 3/10
30/30 [==============================] - 33s 1s/step - loss: 0.0763 - accuracy: 0.9798
Epoch 4/10
30/30 [==============================] - 32s 1s/step - loss: 0.0469 - accuracy: 0.9862
Epoch 5/10
30/30 [==============================] - 32s 1s/step - loss: 0.0304 - accuracy: 0.9922
Epoch 6/10
```

8.2 User Acceptance Testing

1. Purpose of Document

The purpose of this document is briefly explain the test coverage and open issues of the [Product Name] project at the time of the release to User Acceptance Testing (UAT) .

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity lever and how they were resolved**.**

Resolution	**Severity 1**	**Severity 2**	**Severity 3**	**Severity 4**	**Subtotal**
By Design	11	2	3	2	18
Duplicate	1	3	4	0	8
External	3	5	0	0	8
Fixed	12	2	5	22	41
Not Reproduced	0	1	0	0	1
Skipped	0	0	1	2	3
Won't Fix	0	4	1	1	7
Totals	27	17	14	27	86

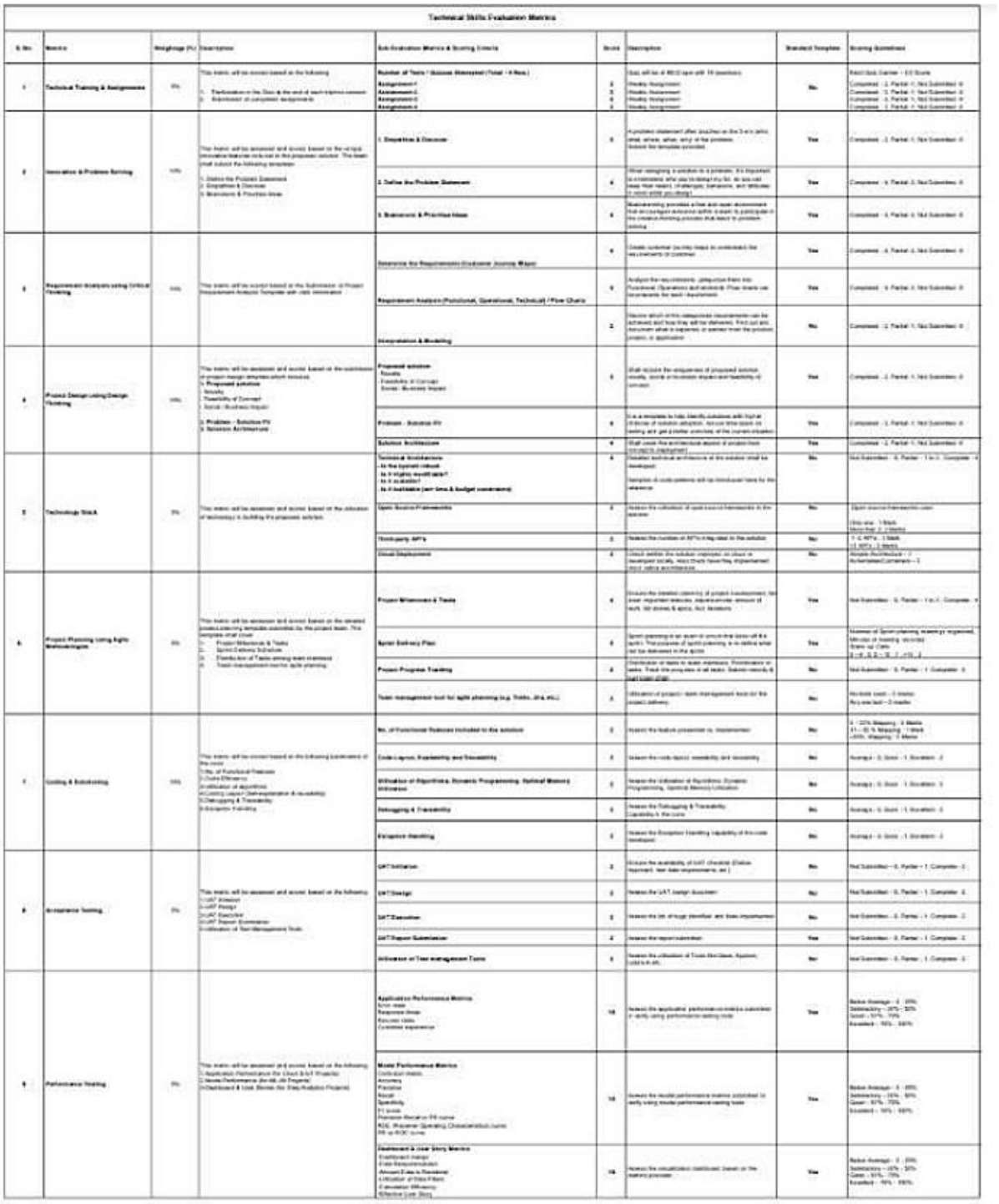
3. Test Case Analysis

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

Section	**Total Cases**	**Not Tested**	**Fail**	**Pass**
Print	8	0	0	8
Client Application	49	0	0	49
Security	4	0	0	4
Outsource Shipping	4	0	0	4
Exception Reporting	11	0	0	11
Final Report Output	2	0	0	2
Version Control	1	0	0	1

9. RESULTS

9.1 Performance Metrics



Technical Skills Evaluation Metrics

S.No	Metrics	Weightage (%)	Description	Sub-Evaluation Metrics & Scoring Criteria	Score	Description	Standard Template	Scoring Guidelines
1	**Technical Training & Assignments**	5%	This metric will be scored based on the following: 1. Participation in the Quiz at the end of each training session 2. Submission of completed assignments	**Number of Tests / Quizzes Attempted (Total – 4 Nos.)** **Assignment-1** **Assignment-2** **Assignment-3** **Assignment-4**	5 5 5 5 5	Quiz will be of MCQ type with 10 questions Weekly Assignment Weekly Assignment Weekly Assignment Weekly Assignment	No	Pass/Quiz Cutoff – 3.5 Score Completed – 2, Partial -1, Not Submitted -0 Completed – 2, Partial -1, Not Submitted -0 Completed – 2, Partial -1, Not Submitted -0 Completed – 2, Partial -1, Not Submitted -0
2	**Innovation & Problem Solving**	10%	This metric will be assessed and scored based on the unique innovative features included in the proposed solution. The team shall submit the following templates: 1. Define the Problem Statement 2. Empathize & Discover 3. Brainstorm & Prioritize Ideas	**1. Empathize & Discover**	5	A problem statement after analyses on the 5 w's who, what, where, when, why of the problem. Submit the template provided	Yes	Completed – 2, Partial -1, Not Submitted -0
				2. Define the Problem Statement	4	While designing a solution to a problem, it's important to understand who you're trying to help, do you care, needs, their wants, challenges, behaviors, and attitudes in some while you design	Yes	Completed – 4, Partial -2, Not Submitted -0
				3. Brainstorm & Prioritize Ideas	6	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	Yes	Completed – 6, Partial -3, Not Submitted -0
3	**Requirement Analysis using Critical Thinking**	10%	This metric will be scored based on the Submission of Project Requirement Analysis Template with clear information	Determine the Requirements (Customer Journey Maps)	4	Create customer journey maps to understand the requirements of customers	Yes	Completed – 4, Partial -2, Not Submitted -0
				Requirement Analysis (Functional, Operational, Technical) / Flow Charts	4	Analyze the requirements, categorize them into Functional (Operational) and non-functional. Flow charts can be templates for each requirement	Yes	Completed – 4, Partial -2, Not Submitted -0
				Interpretation & Modelling	2	Discuss which of the categorized requirements can be achieved and how they will be delivered. Find out any concurrent efforts in progress, or planned from the previous projects, or applications	No	Completed – 2, Partial -1, Not Submitted -0
4	**Project Design using Design Thinking**	10%	This metric will be assessed and scored based on the submission of project design templates which includes: **1. Proposed solution** - Novelty - Feasibility of Concept - Social / Business Impact **2. Problem - Solution Fit** **3. Solution Architecture**	**Proposed solution** Novelty - Feasibility of Concept Social / Business Impact	3	Shall include the uniqueness of proposed solution, novelty, social or business impact and feasibility of concept	Yes	Completed – 2, Partial -1, Not Submitted -0
				Problem - Solution Fit	4	It is a template to help identify solutions with high chances of solution adoption. Not over time spent on testing and get a better overview of the current situation	Yes	Completed – 3, Partial -1, Not Submitted -0
				Solution Architecture	4	Shall cover the architectural aspect of project from concept to deployment	Yes	Completed – 2, Partial -1, Not Submitted -0
5	**Technology Stack**	5%	This metric will be assessed and scored based on the selection of technology & building the proposed solution	**Technical Architecture** **- Is the system robust** **- Is it highly modifiable?** **- Is it scalable?** **- Is it available (part time & budget constraints)**	4	Detailed technical architecture of the solution shall be developed. Selection of code patterns will be introduced here by the vendor(s)	No	Not Submitted – 0, Partial – 1 to 3, Complete – 4
				Open Source Frameworks	2	Assess the utilization of open source frameworks in the solution	No	Open source frameworks used: Only one - 1 Mark More than 2 - 2 Marks
				Third-party API's	2	Assess the number of API's integrated in the solution	No	1 -2 API's - 1 Mark >3 API's - 2 Marks
				Cloud Deployment	2	Check within the solution (vendor) on cloud is developed locally, which clearly describes deployment and it's native architecture	No	Multiple Architecture – 2 Architecture Comments – 0
6	**Project Planning using Agile Methodologies**	5%	This metric will be assessed and scored based on the detailed project planning template submitted by the project team. The template shall cover: 1. Project Milestones & Tasks 2. Sprint Delivery Schedule 3. Distribution of Tasks among team members 4. Tools / management tool for agile planning	**Project Milestones & Tasks**	4	Ensure the detailed planning of project development, list down important features, expected number / amount of work, list stories & epics, key decisions	Yes	Not Submitted – 0, Partial – 1 to 3, Complete – 4
				Sprint Delivery Plan	3	Sprint planning is an event to ensure the team will be in sync. The purpose of sprint planning is to define what can be delivered in the sprint	Yes	Number of Sprint planning meetings organized, Minutes of meeting recorded Wrap up Calls 3 - 2 - 1 - 0
				Project Progress Tracking	3	Dashboard or tools to track milestones, Prioritization of tasks. Track the progress in all tasks. Submit weekly & over tasks done	No	Not Submitted – 0, Partial – 1, Complete – 3
				Team management tool for agile planning (e.g. Trello, Jira, etc.)	3	Utilization of project / team management tool for the project delivery	No	No tools used – 0 marks Any one tool – 2 marks
7	**Coding & Refactoring**	10%	This metric will be scored based on the following parameters of the code: 1.No. of Functional Features 2.Code Efficiency 3.Utilization of algorithms 4.Coding Layer / Data exploration & re-usability 5.Debugging & Traceability 6.Exception Handling	**No. of Functional Features included in the solution**	2	Assess the feature proposed vs. implemented	No	0 – 25% Wrapping - 0 Marks 31 – 50 % Wrapping - 1 Mark >50% Wrapping - 2 Marks
				Code Layout, Explainability and Reusability	2	Assess the code layout, readability and reusability	No	Average – 0, Good – 1, Excellent - 2
				Utilization of Algorithms, Dynamic Programming, Optimal Memory Utilization	2	Assess the utilization of Algorithms, Dynamic Programming, Optimal Memory Utilization	No	Average – 0, Good – 1, Excellent - 2
				Debugging & Traceability	2	Assess the Debugging & Traceability Capability in the code	No	Average – 0, Good – 1, Excellent - 2
				Exception Handling	2	Assess the Exception Handling capability of the code developed	No	Average – 0, Good – 1, Excellent - 2
8	**Acceptance Testing**	5%	This metric will be assessed and scored based on the following: 1.UAT Artefact 2.UAT Design 3.UAT Execution 4.UAT Report Submission 5.Utilization of Test Management Tools	**UAT Definition**	2	Ensure the availability of UAT checklist (Define Approach, test data requirements, etc.)	No	Not Submitted – 0, Partial – 1, Complete - 2
				UAT Design	2	Assess the UAT design document	No	Not Submitted – 0, Partial – 1, Complete - 2
				UAT Execution	2	Assess the list of bugs identified and fixes implemented	No	Not Submitted – 0, Partial – 1, Complete - 2
				UAT Report Submission	2	Assess the report submission	Yes	Not Submitted – 0, Partial – 1, Complete - 2
				Utilization of Test management Tools	2	Assess the utilization of Tools like Qtest, Appium, Load N etc.	No	Not Submitted – 0, Partial – 1, Complete - 2
9	**Performance Testing**	5%	This metric will be assessed and scored based on the following: 1.Application Performance like Load & IoT Projects 2.Model Performance like ML/DL Projects 3.Dashboard & Load Stories like Data Analytics Projects	**Application Performance Metrics** Error rates Response times Execution rates Customer experience	10	Assess the application performance metrics submitted & verify using performance testing tools	Yes	Below Average – 0 - 25% Satisfactory – 26% - 50% Good – 51% - 75% Excellent – 76% - 100%
				Model Performance Metrics Confusion matrix Accuracy Precision Recall Specificity F1 score Precision-Recall or PR curve ROC (Receiver Operating Characteristic) curve PR vs ROC curves	10	Assess the model performance metrics submitted & verify using model performance testing tools	Yes	Below Average – 0 - 25% Satisfactory – 26% - 50% Good – 51% - 75% Excellent – 76% - 100%
				Dashboard & Load Story Metrics Dashboard design Data Responsiveness Actual Data vs Simulated Utilization of Data Filters Calculation Efficiency Effective Load Story	10	Assess the visualization / dashboard based on the metrics provided	Yes	Below Average – 0 - 25% Satisfactory – 26% - 50% Good – 51% - 75% Excellent – 76% - 100%

10. ADVANTAGES & DISADVANTAGES

Advantages :

- It is a cost-effective way of getting several people from different locations to attend meetings and conferences.
- It enables employees from across the world to communicate with each other 24×7 and share ideas or solve problems quickly.

Disadvantages :

- Also accuracy depends upon distance between camera and object.
- It takes a lot of time to listen, speak, read, or write to someone.

11. CONCLUSION

The proposed communication system between Deaf and Dumb people and ordinary people are aiming for it when bridging the communication gap between two societies. It provides complete two - sided communication in an efficient manner between the disabled and the normal person.

For communication between deaf person and a second person, a mediator is required to translate sign language of deaf person. But a mediator is required to know the sign language used by deaf person. But this is not always possible since there are multiple sign languages for multiple languages.

So to understand all sign languages, Hand gestures of deaf peoples by normal peoples this system is proposed.

12. FUTURE SCOPE

The speech-to-text and text-to-speech technologies helped those people who had difficulties in communicating or expressing their feelings to the normal people.

This reduces the communication gap between the normal people and the specially abled people.

Using image pre-processing and Artificial Intelligence it is easy to understand the context of objects and clearly explains it to the people who use it for communication.

13. APPENDIX

Source Code :

```python
import cv2

video = cv2.VideoCapture(0)

while True:
    ret, frame = video.read()
    cv2.imshow("Frame", frame)
    k = cv2.waitKey(1)
    if k == ord('q'):
        break

video.release()
cv2.destroyAllWindows()
```

```python
import cv2
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

class Video(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.roi_start = (50, 150)
        self.roi_end = (250, 350)
        self.model = load_model('asl_model.h5') # Execute Local Trained Model
        # self.model = load_model('IBM_Communication_Model.h5') # Execute IBM Trained Model
        self.index=['A','B','C','D','E','F','G','H','I']
        self.y = None
    def __del__(self):
        self.video.release()
    def get_frame(self):
        ret,frame = self.video.read()
        frame = cv2.resize(frame, (640, 480))
        copy = frame.copy()
        copy = copy[150:150+200,50:50+200]
        # Prediction Start
        cv2.imwrite('image.jpg',copy)
        copy_img = image.load_img('image.jpg', target_size=(64,64))
        x = image.img_to_array(copy_img)
        x = np.expand_dims(x, axis=0)
        pred = np.argmax(self.model.predict(x), axis=1)
        self.y = pred[0]
        cv2.putText(frame,'The Predicted Alphabet is: '+str(self.index[self.y]),(100,50),cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,0),3)
        ret,jpg = cv2.imencode('.jpg', frame)
        return jpg.tobytes()
```

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1">
<style>
body {font-family: Arial, Helvetica, sans-serif;}

/* Full-width input fields */
input[type=text], input[type=password] {
  width: 100%;
  padding: 12px 20px;
  margin: 8px 0;
  display: inline-block;
  border: 1px solid #ccc;
  box-sizing: border-box;
}

/* Set a style for all buttons */
button {
  background-color: #273298;
  color: white;
  padding: 14px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  width: 100%;
}

button:hover {
  opacity: 0.8;
}
```

```
/* Extra styles for the cancel button */
.cancelbtn {
  width: auto;
  padding: 10px 18px;
  background-color: #f44336;
}

/* Center the image and position the close button */
.imgcontainer {
  text-align: center;
  margin: 24px 0 12px 0;
  position: relative;
}

img.avatar {
  width: 40%;
  border-radius: 50%;
}

.container {
  padding: 16px;
}

span.psw {
  float: right;
  padding-top: 16px;
}

/* The Modal (background) */
.modal {
  display: none; /* Hidden by default */
  position: fixed; /* Stay in place */
  z-index: 1; /* Sit on top */
  left: 0;
  top: 0;
  width: 100%; /* Full width */
  height: 100%; /* Full height */
```

```
    overflow: auto; /* Enable scroll if needed */
    background-color: rgb(0,0,0); /* Fallback color */
    background-color: rgba(0,0,0,0.4); /* Black w/ opacity */
    padding-top: 60px;
  }

  /* Modal Content/Box */
  .modal-content {
    background-color: #fefefe;
    margin: 5% auto 15% auto; /* 5% from the top, 15% from the bottom and centered */
    border: 1px solid #888;
    width: 80%; /* Could be more or less, depending on screen size */
  }

  /* The Close Button (x) */
  .close {
    position: absolute;
    right: 25px;
    top: 0;
    color: #000;
    font-size: 35px;
    font-weight: bold;
  }

  .close:hover,
  .close:focus {
    color: red;
    cursor: pointer;
  }

  /* Add Zoom Animation */
  .animate {
    -webkit-animation: animatezoom 0.6s;
    animation: animatezoom 0.6s
  }
```

```
@-webkit-keyframes animatezoom {
  from {-webkit-transform: scale(0)}
  to {-webkit-transform: scale(1)}
}

@keyframes animatezoom {
  from {transform: scale(0)}
  to {transform: scale(1)}
}

/* Change styles for span and cancel button on extra small screens */
@media screen and (max-width: 300px) {
  span.psw {
    display: block;
    float: none;
  }
  .cancelbtn {
    width: 100%;
  }
}
</style>
</head>
<body>

<h2 style="text-align: center; color: rgba(1, 2, 2, 0.874);">REAL TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED</h2>

<button onclick="document.getElementById('id01').style.display='block'">Login</button>

<div id="id01" class="modal">

  <form class="modal-content animate" action="/action_page.php" method="post">
    <div class="imgcontainer">
      <span onclick="document.getElementById('id01').style.display='none'" class="close" title="Close Modal">&times;</span>
      <img src="https://static.vecteezy.com/system/resources/thumbnails/007/407/996/small/user-icon-person-icon-client-symbol-login-head-sign-icon-design-vector.jpg">
    </div>
```

```
    <div class="container">
      <label for="uname"><b>Username</b></label>
      <input type="text" placeholder="Enter Username" name="uname" required>

      <label for="psw"><b>Password</b></label>
      <input type="password" placeholder="Enter Password" name="psw" required>

      <button type="submit">Login</button>
      <label>
        <input type="checkbox" checked="checked" name="remember"> Remember me
      </label>
    </div>

    <div class="container" style="background-color:#f1f1f1">
      <button type="button" onclick="document.getElementById('id01').style.display='none'" class="cancelbtn">Cancel</button>
      <span class="psw">Forgot <a href="#">password?</a></span>
    </div>
  </form>
</div>
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
          content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css">
    <link rel="stylesheet" href="style.css">
    <title>Document</title>
</head>
<body>
<div class="display-cover">
    <video autoplay></video>
    <canvas class="d-none"></canvas>

    <div class="video-options">
        <select name="" id="" class="custom-select">
```

```
            <option value="">Select camera</option>
        </select>
    </div>

    <img class="screenshot-image d-none" alt="">

    <div class="controls">
        <button class="btn btn-danger play" title="Play"><i data-feather="play-circle"></i></button>
        <button class="btn btn-info pause d-none" title="Pause"><i data-feather="pause"></i></button>
        <button class="btn btn-outline-success screenshot d-none" title="ScreenShot"><i data-feather="image"></i></button>
    </div>
</div>

<script src="https://unpkg.com/feather-icons"></script>
<script src="script.js"></script>
</body>
<html><head>
</head><body>
    <video src="" ></video>
    <br />
<button id='flipCamera'>Flip</button>
</body>
<script>
  var front = false;
var video = document.querySelector('video');
  document.getElementById('flipCamera').onclick = function() { front = !front; };
  var constraints = { video: { facingMode: (front? "user" : "environment"), width: 640, height: 480  } };
  navigator.mediaDevices.getUserMedia(constraints)
  .then(function(mediaStream) {
    video.srcObject = mediaStream;
    video.onloadedmetadata = function(e) {
    video.play();
};
})
.catch(function(err) { console.log(err.name + ": " + err.message); })
</script></html>
</html>
```

```
<style>
.screenshot-image {
    width: 150px;
    height: 90px;
    border-radius: 4px;
    border: 2px solid whitesmoke;
    box-shadow: 0 1px 2px 0 rgba(0, 0, 0, 0.1);
    position: absolute;
    bottom: 5px;
    left: 10px;
    background: white;
}

.display-cover {
    display: flex;
    justify-content: center;
    align-items: center;
    width: 70%;
    margin: 5% auto;
    position: relative;
}

video {
    width: 100%;
    background: rgba(0, 0, 0, 0.2);
}

.video-options {
    position: absolute;
    left: 20px;
    top: 30px;
}

.controls {
    position: absolute;
    right: 20px;
    top: 20px;
```

```
        display: flex;
    }

    .controls > button {
        width: 45px;
        height: 45px;
        text-align: center;
        border-radius: 100%;
        margin: 0 6px;
        background: transparent;
    }

    .controls > button:hover svg {
        color: white !important;
    }

    @media (min-width: 300px) and (max-width: 400px) {
        .controls {
            flex-direction: column;
        }

        .controls button {
            margin: 5px 0 !important;
        }
    }

    .controls > button > svg {
        height: 20px;
        width: 18px;
        text-align: center;
        margin: 0 auto;
        padding: 0;
    }

    .controls button:nth-child(1) {
        border: 2px solid #1a12b3;
    }
```

```
.controls button:nth-child(1) svg {
    color: #2b128e;
}

.controls button:nth-child(2) {
    border: 2px solid #008496;
}

.controls button:nth-child(2) svg {
    color: #008496;
}

.controls button:nth-child(3) {
    border: 2px solid #0048b5;
}

.controls button:nth-child(3) svg {
    color: #0f0a5b;
}

.controls > button {
    width: 45px;
    height: 45px;
    text-align: center;
    border-radius: 100%;
    margin: 0 6px;
    background: transparent;
}

.controls > button:hover svg {
    color: rgb(75, 173, 230);
}
</style>

<script>
// Get the modal
var modal = document.getElementById('id01');
```

```
// When the user clicks anywhere outside of the modal, close it
window.onclick = function(event) {
    if (event.target == modal) {
        modal.style.display = "none";
    }
}
feather.replace();

const controls = document.querySelector('.controls');
const cameraOptions = document.querySelector('.video-options>select');
const video = document.querySelector('video');
const canvas = document.querySelector('canvas');
const screenshotImage = document.querySelector('img');
const buttons = [...controls.querySelectorAll('button')];
let streamStarted = false;

const [play, pause, screenshot] = buttons;

const constraints = {
  video: {
    width: {
      min: 1280,
      ideal: 1920,
      max: 2560,
    },
    height: {
      min: 720,
      ideal: 1080,
      max: 1440
    },
  }
};
</script>
<script>
const getCameraSelection = async () => {
  const devices = await navigator.mediaDevices.enumerateDevices();
```

```
    const videoDevices = devices.filter(device => device.kind === 'videoinput');
    const options = videoDevices.map(videoDevice => {
      return `<option value="${videoDevice.deviceId}">${videoDevice.label}</option>`;
    });
    cameraOptions.innerHTML = options.join('');
  };

  </script>
  <script>

  play.onclick = () => {
    if (streamStarted) {
      video.play();
      play.classList.add('d-none');
      pause.classList.remove('d-none');
      return;
    }
    if ('mediaDevices' in navigator && navigator.mediaDevices.getUserMedia) {
      const updatedConstraints = {
        ...constraints,
        deviceId: {
          exact: cameraOptions.value
        }
      };
      startStream(updatedConstraints);
    }
  };

  const startStream = async (constraints) => {
    const stream = await navigator.mediaDevices.getUserMedia(constraints);
    handleStream(stream);
  };

  const handleStream = (stream) => {
    video.srcObject = stream;
    play.classList.add('d-none');
    pause.classList.remove('d-none');
```

```
    screenshot.classList.remove('d-none');
    streamStarted = true;
  };

  getCameraSelection();
  ...
  cameraOptions.onchange = () => {
    const updatedConstraints = {
      ...constraints,
      deviceId: {
        exact: cameraOptions.value
      }
    };
    startStream(updatedConstraints);
  };

  const pauseStream = () => {
    video.pause();
    play.classList.remove('d-none');
    pause.classList.add('d-none');
  };

  const doScreenshot = () => {
    canvas.width = video.videoWidth;
    canvas.height = video.videoHeight;
    canvas.getContext('2d').drawImage(video, 0, 0);
    screenshotImage.src = canvas.toDataURL('image/webp');
    screenshotImage.classList.remove('d-none');
  };

  pause.onclick = pauseStream;
  screenshot.onclick = doScreenshot;
  </script>

  </body>
  </html>
```

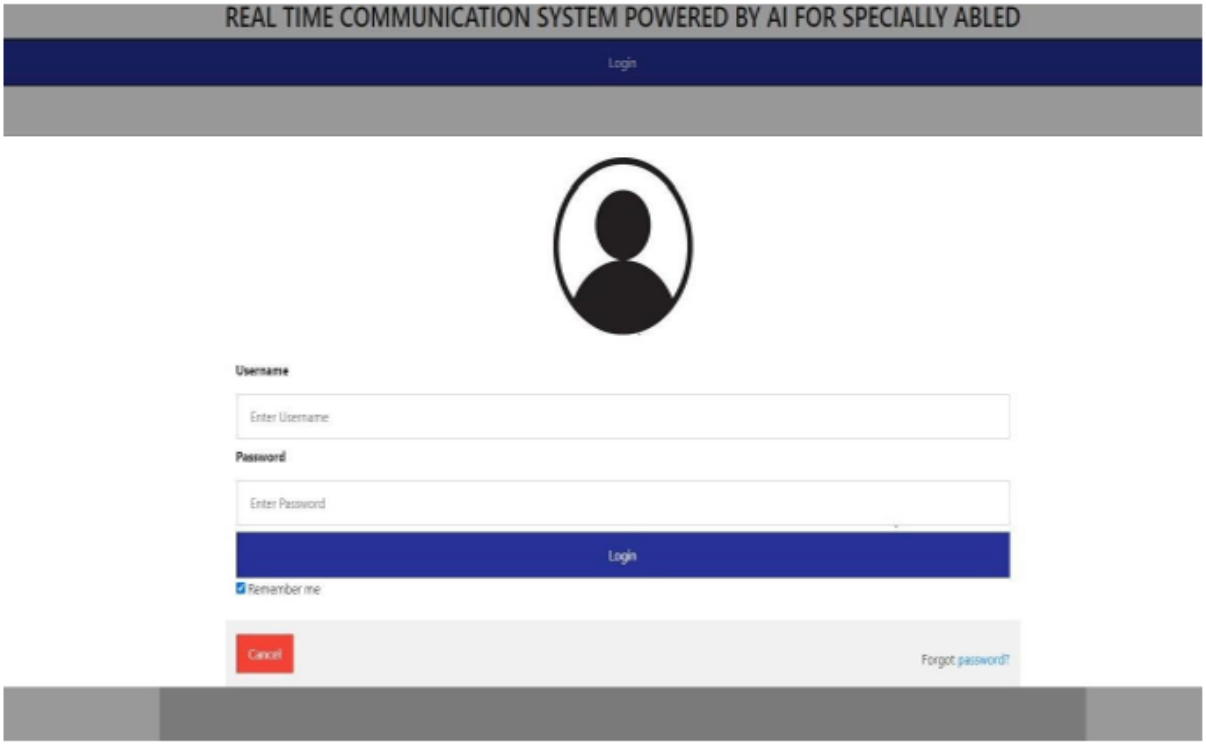
```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, shrink-to-fit=no">
    <title>SmartBridge_WebApp_VideoTemplate</title>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css">
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.12.0/css/all.css">
    <link rel="stylesheet" href="assets/css/Banner-Heading-Image.css">
    <link rel="stylesheet" href="assets/css/Navbar-Centered-Brand.css">
    <link rel="stylesheet" href="assets/css/styles.css">
</head>

<body style="background: rgb(39,43,48);">
    <nav class="navbar navbar-light navbar-expand-md py-3" style="background: #212529;">
        <div class="container">
            <div></div><a class="navbar-brand d-flex align-items-center" href="#"><span
                    class="bs-icon-sm bs-icon-rounded bs-icon-primary d-flex justify-content-center align-items-center me-2 bs-icon"><i
                        class="fas fa-flask"></i></span><span style="color: rgb(255,255,255);">Real-Time Communication
                    System Powered By AI For Specially Abled</span></a>
            <div></div>
        </div>
    </nav>
    <section>
        <div class="d-flex flex-column justify-content-center align-items-center">
            <div class="d-flex flex-column justify-content-center align-items-center" id="div-video-feed"
                style="width: 640px;height: 480px;margin: 10px;min-height: 480px;min-width: 640px;border-radius: 10px;border: 4px dashed rgb(255,255,255) ;">
                <img src="{{ url_for('video_feed') }}" style="width: 100%;height: 100%;color: rgb(255,255,255);text-align: center;font-size: 20px;"
                    alt="Camera Access Not Provided!">
            </div>
        </div>
        <div class="d-flex flex-column justify-content-center align-items-center" style="margin-bottom: 10px;"><button
                class="btn btn-info" type="button" data-bs-target="#modal-1" data-bs-toggle="modal">Quick Reference
                -<strong> ASL Alphabets</strong></button></div>
    </section>
    <section>
```

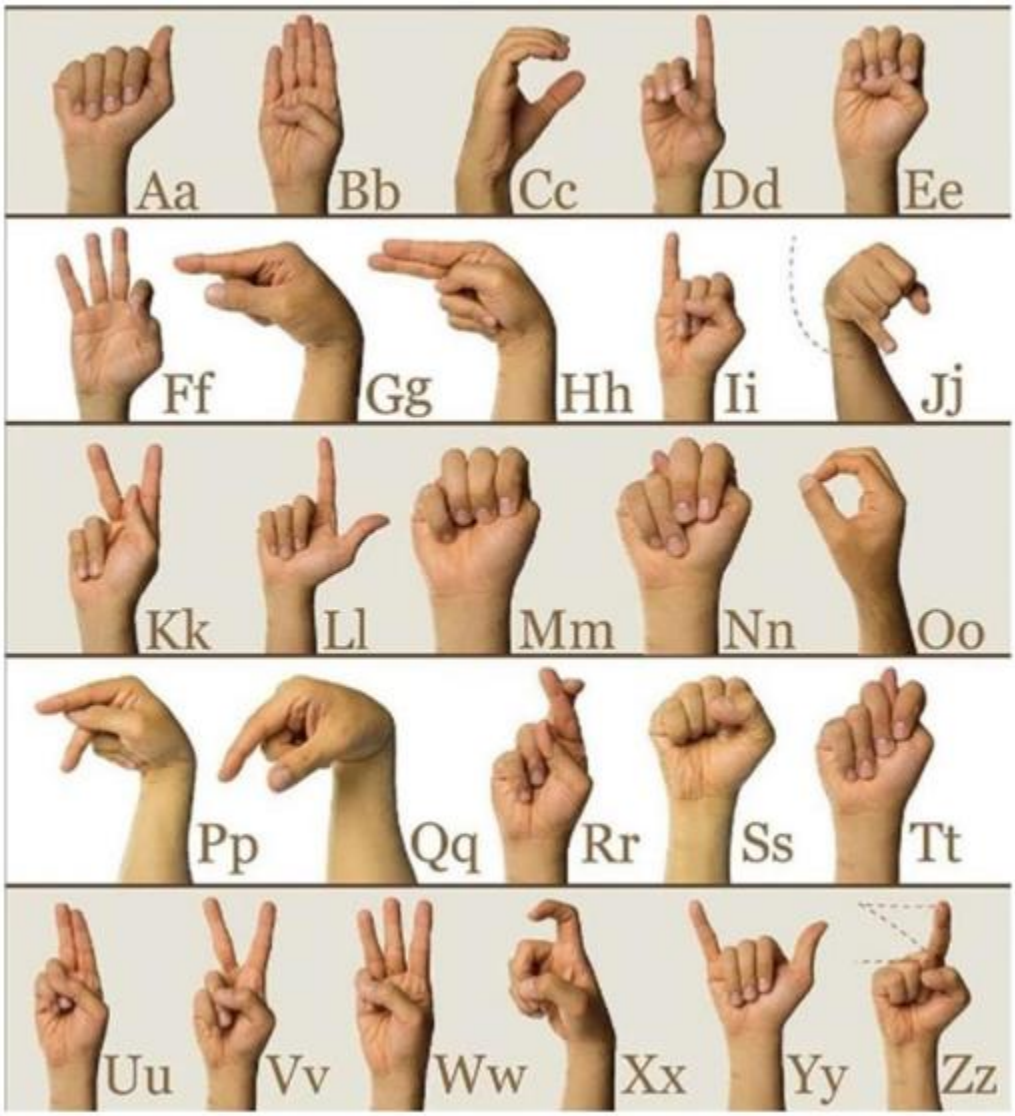
```html
        <div class="container">
            <div class="accordion text-white" role="tablist" id="accordion-1">
                <div class="accordion-item" style="background: rgb(33,37,41);">
                    <h2 class="accordion-header" role="tab"><button class="accordion-button" data-bs-toggle="collapse"
                            data-bs-target="#accordion-1 .item-1" aria-expanded="true"
                            aria-controls="accordion-1 .item-1"
                            style="background: rgb(39,43,48);color: rgb(255,255,255);">About The Project</button></h2>
                    <div class="accordion-collapse collapse show item-1" role="tabpanel" data-bs-parent="#accordion-1">
                        <div class="accordion-body">
                            <p class="mb-0">Artificial Intelligence has made it possible to handle our daily activities
                                in new and simpler ways. With the ability to automate tasks that normally require human
                                intelligence, such as speech and voice recognition, visual perception, predictive text
                                functionality, decision-making, and a variety of other tasks, AI can assist people with
                                disabilities by significantly improving their ability to get around and participate in
                                daily activities.<br><br>Currently, Sign Recognition is available <strong>only for
                                    alphabets A-I</strong> and not for J-Z, since J-Z alphabets also require Gesture
                                Recognition for them to be able to be predicted correctly to a certain degree of
                                accuracy.</p>
                        </div>
                    </div>
                </div>
                <div class="accordion-item" style="background: rgb(33,37,41);">
                    <h2 class="accordion-header" role="tab"><button class="accordion-button collapsed"
                            data-bs-toggle="collapse" data-bs-target="#accordion-1 .item-2" aria-expanded="false"
                            aria-controls="accordion-1 .item-2"
                            style="background: rgb(39,43,48);color: rgb(231,241,255);">Developed By</button></h2>
                    <div class="accordion-collapse collapse item-2" role="tabpanel" data-bs-parent="#accordion-1">
                        <div class="accordion-body">
                            <p class="mb-0">Students at VIT-Bhopal University during SmartBridge AI Externship
                                Program.<br><br>1. <strong>Nirlov Deb</strong> 19BCG10067<br>2.
                                <strong>Kushagra</strong> 19BCG10025<br>3. <strong>Kartik Dhasmana</strong> 19BCG10002
                            </p>
                        </div>
                    </div>
                </div>
            </div>
        </div>
```

```
    </section>
    <div class="modal fade" role="dialog" tabindex="-1" id="modal-1">
        <div class="modal-dialog" role="document">
            <div class="modal-content">
                <div class="modal-header">
                    <h4 class="modal-title">American Sign Language - Alphabets</h4><button type="button"
                        class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
                </div>
                <div class="modal-body"><img src="{{ url_for('static', filename='img/ASL_Alphabets.png') }}" width="100%"></div>
                <div class="modal-footer"><button class="btn btn-secondary" type="button"
                        data-bs-dismiss="modal">Close</button></div>
            </div>
        </div>
    </div>
    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
</body>

</html>
```

```
.bs-icon {
  --bs-icon-size: .75rem;
  display: flex;
  flex-shrink: 0;
  justify-content: center;
  align-items: center;
  font-size: var(--bs-icon-size);
  width: calc(var(--bs-icon-size) * 2);
  height: calc(var(--bs-icon-size) * 2);
  color: var(--bs-primary);
}

.bs-icon-xs {
  --bs-icon-size: 1rem;
  width: calc(var(--bs-icon-size) * 1.5);
  height: calc(var(--bs-icon-size) * 1.5);
}

.bs-icon-sm {
  --bs-icon-size: 1rem;
}

.bs-icon-md {
  --bs-icon-size: 1.5rem;
}

.bs-icon-lg {
  --bs-icon-size: 2rem;
}

.bs-icon-xl {
  --bs-icon-size: 2.5rem;
}

.bs-icon.bs-icon-primary {
  color: var(--bs-white);
  background: var(--bs-primary);
```

```
}

.bs-icon.bs-icon-primary-light {
  color: var(--bs-primary);
  background: rgba(var(--bs-primary-rgb), .2);
}

.bs-icon.bs-icon-semi-white {
  color: var(--bs-primary);
  background: rgba(255, 255, 255, .5);
}

.bs-icon.bs-icon-rounded {
  border-radius: .5rem;
}

.bs-icon.bs-icon-circle {
  border-radius: 50%;
}
```





GITHUB LINK :

https://github.com/IBM-EPBL/IBM-Project-47296-1660798042