V.S.B. ENGINEERING COLLEGE
(Affiliated by Anna University)
Karudayamapalyam,Karur-639111

Real-Time Communication System Powered by AI for Specially Abled Person

A person report submitted in partial fulfilment of 7th semester in degree of

BACHELOR OF ENGINEERING
IN

COMPUTER SCIENCE AND ENGINEERING

Submitted by

TEAM ID: PNT2022TMID33389

TEAM LEADER : PAVITHRA V (922519104109)
TEAM MEMBER 1 : KAVIPRIYA T (922519104073)
TEAM MEMBER 2 : KIRUTHIKA PARAMESHWARI M (922519104081)
TEAM MEMBER 3 : MADHUMIDHA M (922519104091)

CONTENTS

13. APPENDIX

14. SOURCECODE AND GITHUB LINK

1.INTRODUCTION

a. Project Overview:

- There are handicapped people in our society. Although technology is constantly evolving, little is being done to improve the lives of these people.
- Any method of telecommunication in which all users can instantly exchange information is known as real-time communications (RTC).
- The improvement of communication is crucial to a better world. It forges relationships and bonds amongst people.

b.Purpose:

Aim :

- The project intends to create a system that can translate speech into acceptable sign language for the deaf and dumb and convert sign language into a human hearing voice in the preferred language to communicate with normal people.
- The development of an app allows persons who are deaf or dumb to communicate using signs that are translated into speech and understood by others.

2.LITERATURE SURVEY

a.Existing problem

- It has always been difficult to communicate with someone who is deaf-mute.
- It is quite challenging for silent persons to communicate with non-mute people.
- Since normal people are not trained on hand sign language.
- Only those with special needs are taught sign language, and because the average person has no idea how it works, there is a communication gap.
- Under emergency situations, it is even more difficult for specially abled people to get help.
- Non-Emergency normal environments can also behard for them to navigateneeding special assistance.

b.References

- Koufos, K., EL Haloui, K., Dianati, M., Higgins, M., Elmirghani, J., Imran, M. A., & Tafazolli, R. (2021). Trends in Intelligent Communication Systems: Review of Standards,

Major Research Projects, and Identification of Research Gaps. Journal of Sensor and Actuator Networks, 10(4), 60.

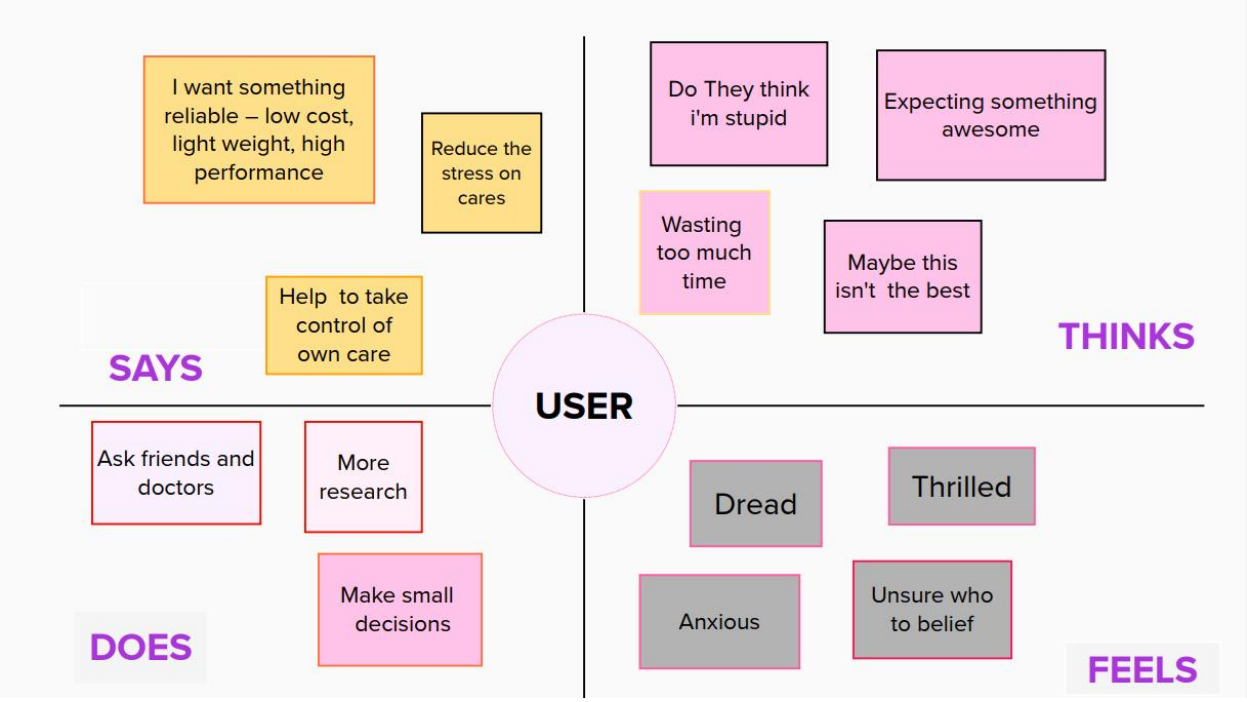
- Panda, G., Upadhyay, A. K., & Khandelwal, K. (2019). Artificial intelligence: A strategic disruption in public relations. Journal of Creative Communications, 14(3), 196-213.
- Xu, G., Mu, Y., & Liu, J. (2017). Inclusion of artificial intelligence in communication networks and services. ITU J. ICT Discov. Spec, 1, 1-6

c.Problem Statement Definition

- In the world, all the people are having the six senses are a multi-platform app for aiding the people in need that is people who are handicapped in the form of lack of speech (dumb), lack of hearing (deaf), lack of sight (blind), lack of judicial power to differentiate between objects (visual agnosia) and people suffering from autism (characterized by great difficulty in communicating and forming relationships with other people and in using language and abstract concepts).
- Our current implementation of the product is on two platforms, namely, mobile and a web app.
- Hence, it has been implemented on the mobile app utilizing the Firebase ML toolkit and different pre-trained models, which are both available offline as well as online

3.IDEATION & PROPOSED SOLUTION

a.Empathy Map Canvas



b.<u>Ideation & Brainstorming</u>

Brainstorm

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.

Pavithra

communication between normal people and inspired people gets easy

difficulty in communicat ion

User credentials

Too many apps used by single user

Desperate

Usage of internet

kavipriya

Identifying the condition of the users

Focused information gathering

Diagnosis and analysis of problem

designing of product according to user convenience

User preference

Data privacy

Madhumidha

Feeling lonely

Inconvenient thoughts

Better Security & Backup

Effective communication

Like option design the user's preference

Improved on time performance

Kiruthuka Parameshwari

difficulty in understandi ng

getting feedback from the user

Improvising the quality of product

Inferiority complex

missing of proper communication

Un employme nt

Group Idea

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.

Catagory 01

Different language options

Keep the UI simple and interactive

Catagory 02

Monitoring facial expression

Recognition of hand gestures

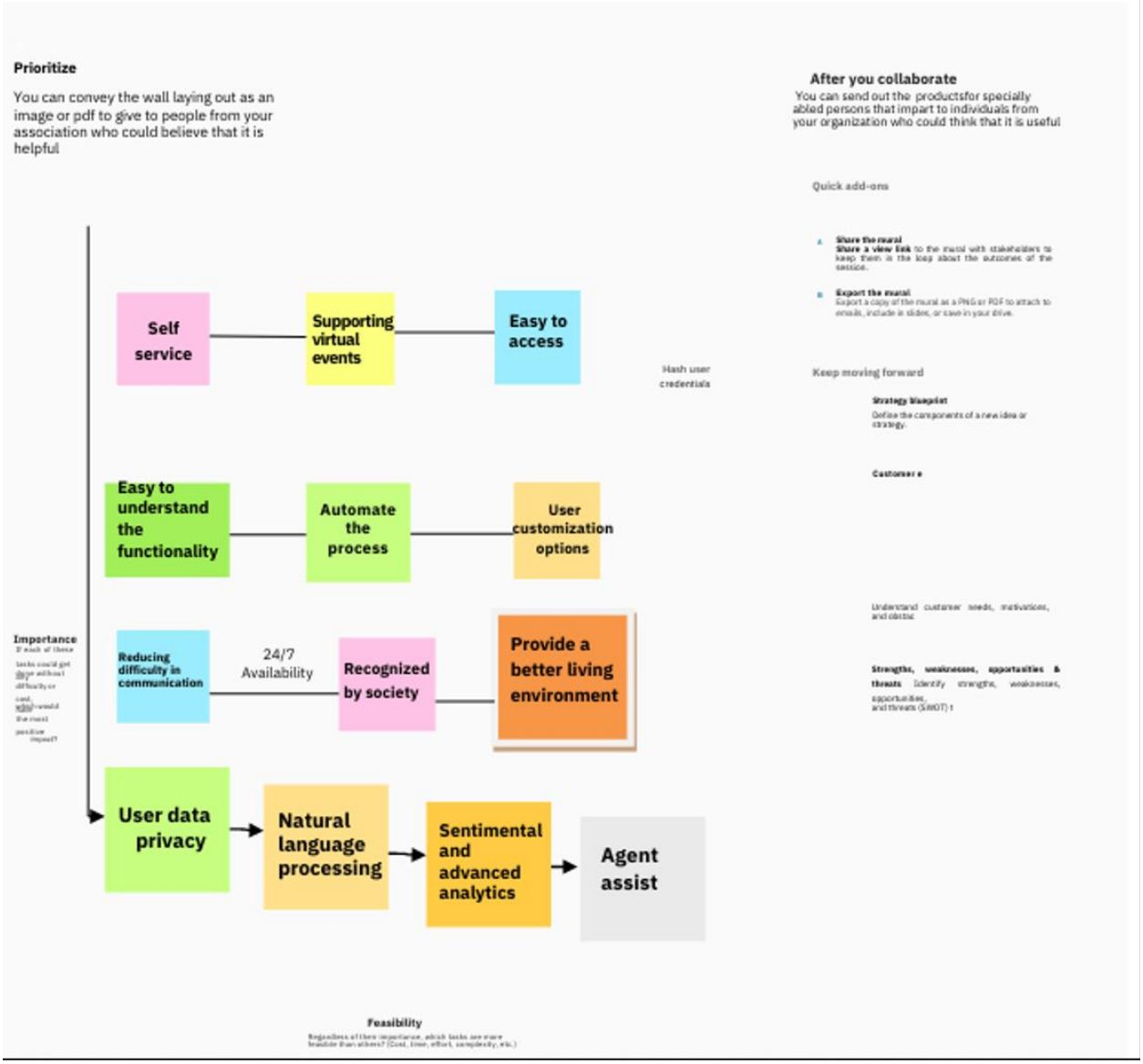
Choice based result

Providing automated services

Catagory 03

Secure and trusted ecosystem

Write maintainable code



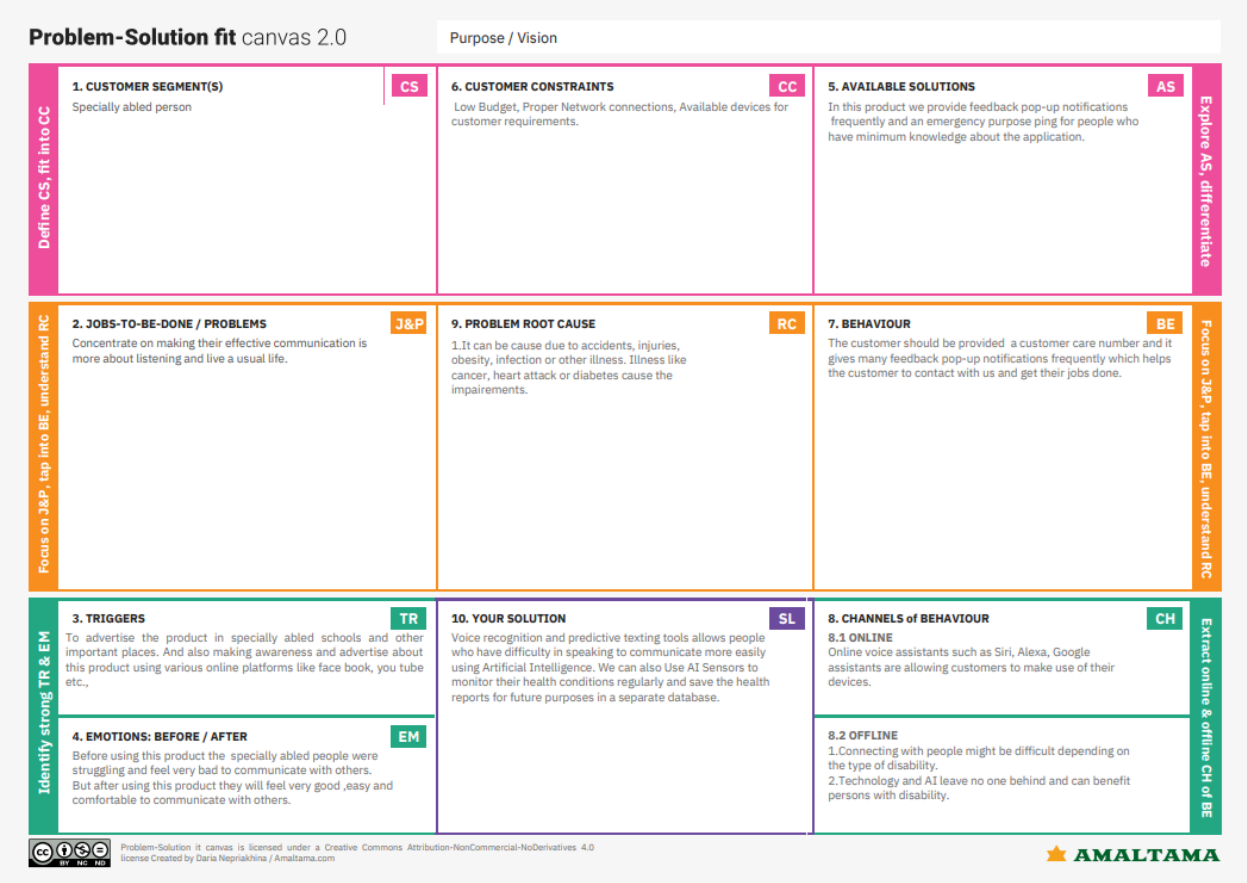
c.<u>Proposed Solution</u>

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	News organizations whose mobile apps only provide users with their articles or videos are missing a big opportunity. An application, by definition, should be applied to perform a task, to solve a problem. Most news doesn't do that. Rather than just feed readers recent stories you wrote *about* their problems, apps can provide tools and data that enable users to actually *solve* their problems. When you solve problems, you get more loyal users and a chance to make more money.
2.	Idea / Solution description	Start with an audience-first focus. Instead of thinking about what your organization produces (news articles) and how to fit that into an app, start from scratch and consider your audience. Who are they, and what problems do they face? Use your imagination, do some role playing with colleagues, and actually talk to some real people.
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 all about solving problems and enabling the user to take an action — go to this bar, shop at that boutique, rent this apartment. These are the apps that build loyal audiences because they help people get things done instead of just presenting another thing they feel obligated to do (keep up with news stories). That is valuable not only for users, but for advertisers as well.
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	News tracker application can handle data across machines and data that will not fit into memory. It supports clusters and can handle machine failures, rebuilding machines easily.

d. Problem Solutionfit



4.REQUIREMENT ANALYSIS

a.Functional requirement

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Authentication	Confirmation through Facial acknowledgment Confirmation through secret kay verification convention
FR-4	External interfaces	Robots and different apparatuses give locally situated care also, other help, permitting individuals with handicaps to freely live
FR-5	Transaction processing	Many application can use to interpret the communication through signing like D talk in the framework
FR-6	Reporting	There is a developing indication that we want to accomplish more, to assist make the existences of individuals with handicaps more straightforward

b.Non-Functional requirements

Non-functional Requirements:

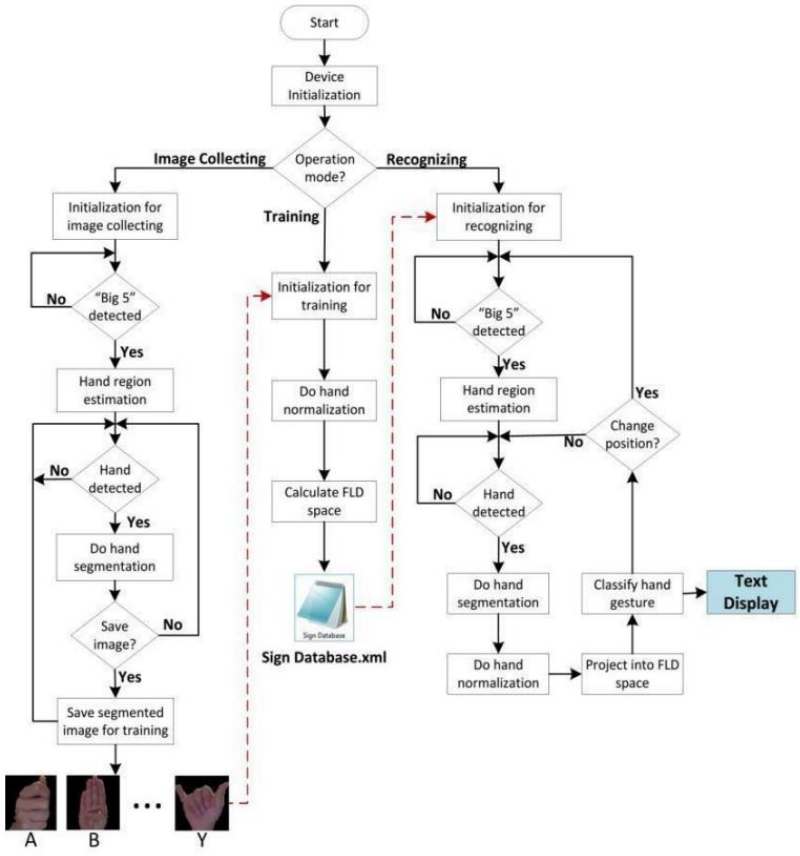
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The application and the product we are developing will be enabled with the facilities of voice engines which helps the user to analyse the surroundings and act accordingly. It also includes features such as speech to text and speech to signs and vice versa which allows deaf people to communicate with the outside world.
NFR-2	**Security**	This application provides a highly confidential platform when it comes to user security. It stores each and every detail of the users in a highly secured database which is impossible to access by the third parties.
NFR-3	**Reliability**	Our application provides quality customer service with high security so this will be trustworthy.
NFR-4	**Performance**	Speed, Accuracy, Efficiency, User-friendly.
NFR-5	**Availability**	Since our application does not contain any premium customers option it will be affordable for all the users. The users can always download it from google play store or apple store or from any other online application platforms.
NFR-6	**Scalability**	It is a single user entity. On the basis of performance this application will work efficiently even with less bandwidth of internet.

5.PROJECT DESIGN

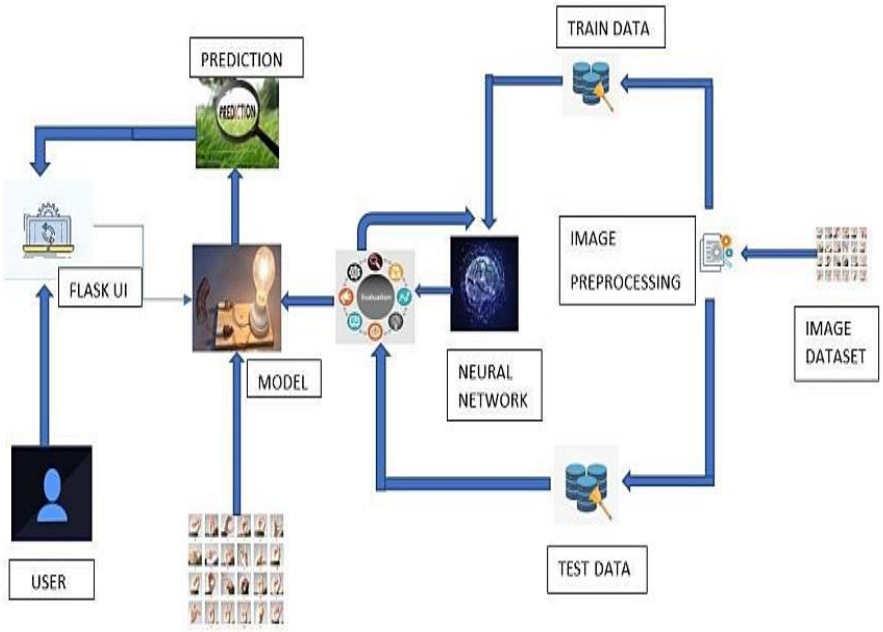
a.Data Flow Diagrams

Dataflow Diagram:



b.Solution & TechnicalArchitecture

Technical Architecture:



c.<u>User Stories</u>

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer *(Desktop user)*	Registration	USN-1	**Not Required**	I can access my account dashboard	High	Sprint-1
	Login	USN-2	**Not Required**		High	Sprint-1
	Dashboard	USN-3	**Not Required**			
Customer *(Desktop user)*	**Main page**	USN-4	As a User, **I can enter the web page once clicked, which provides be the Guidelines** to use the app	I can **enter the web page** once clicked	Medium	Sprint-1
Customer (Desktop user)	**Guidelines**	USN-5	As a User, **I can give a read through the guidelines to understand the functioning of the app.**	I can **give a read through the guidelines.**	Medium	Sprint-1
Customer *(Desktop user)*	**Convert Sign**	USN-6	As a User, **I can click the button Convert sign** , which directs me towards the Main screen	I can **click the button Convert sign and directed me to main screen.**	Medium	Sprint-2
Customer *(Desktop user)*	**Camera (Hand movement detection)**	USN-7	As a User, **I can show my hand sign towards the camera which converts them into text manner.**	**I can show my hand sign** towards the camera accurately.	High	Sprint-2
Customer *(Desktop user)*	**Voice mode**	USN-8	Once the text is obtained, As a User **I can click on the voice mode which provides the text in the form of speech.**	**I can click on the voice mode** which provides the text in the form of speech.	High	Sprint-2

6.PROJECT PLANNING &SCHEDULING

a.Sprint Planning & Estimation

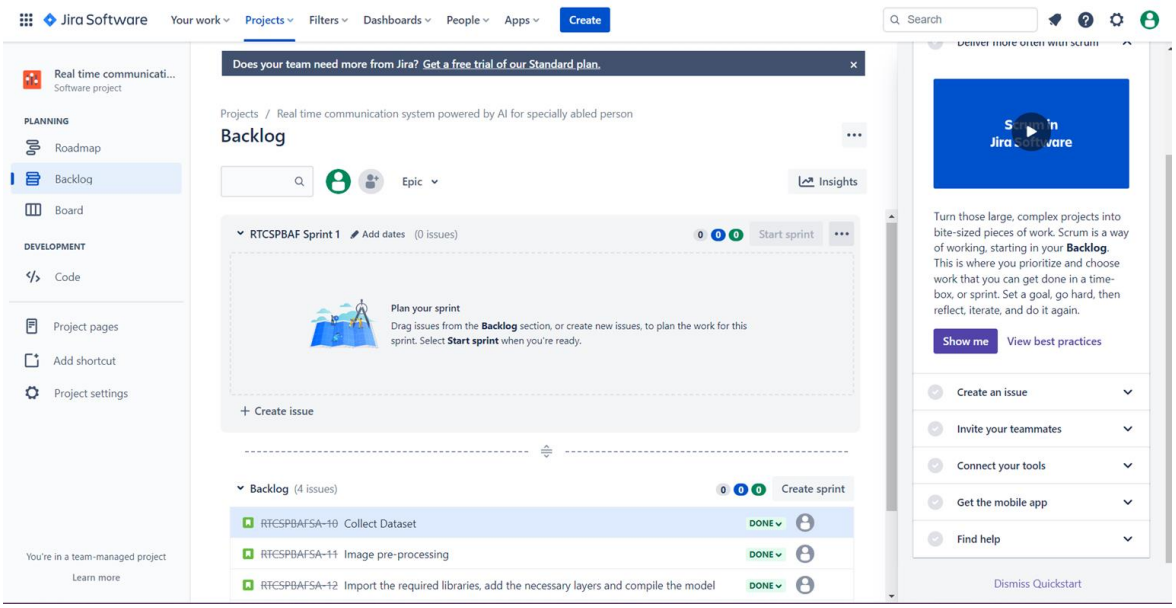
Milestone List:

Milestone Number	Milestone Name	Duration	Submission Dates
MN – 01	Ideation Phase	3 Weeks	17 September, 2022
MN – 02	Project Design Phase – I	2 Weeks	01 October, 2022
MN – 03	Project Design Phase – II	2 Weeks	15 October, 2022
MN – 04	Project Planning Phase	1 Week	22 October, 2022
MN – 05	Project Development Phase	3 Weeks	18 November, 2022
MN – 06	Pre-requisites	1 Week	30 September, 2022
MN – 07	Project Structure	1 Week	02 October, 2022
MN – 08	Data Collection	2 Days	04 October, 2022
MN – 09	Image Pre-processing	4 Days	08 October, 2022
MN – 10	Model Building	1 Week	19 October, 2022
MN – 11	Test the model	2 Days	20 October, 2022
MN – 12	Application Building	1 Week	22 October, 2022
MN – 13	Train CNN Model on IBM	2 Days	30 October, 2022

b.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.	3	High	PAVITHRA V
Sprint – 1	Authentication	USN – 2	As a user, I will receive OTP to confirm details.	2	High	PAVITHRA V
Sprint – 1	Registration	USN – 3	As a user, I will receive confirmation email once I have registered for the application.	1	Low	KIRUTHIKA PARAMESHWARI M
Sprint – 1	Login	USN – 4	As a user, I can log into the application by entering email & password.	2	High	KIRUTHIKA PARAMESHWARI M
Sprint – 2	Dashboard	USN – 5	As a user, I must have one place to explore all available features.	3	High	KAVIPRIYA T
Sprint – 2	Login	USN – 6	As a user, If I forget my password, I must get an auto-generated password to reset my password.	2	Medium	MADHUMIDHA M
Sprint – 3	Help	USN – 7	As a user, I must be able to reach out to the Support Team to get my issues resolved.	1	Low	PAVITHRA V
Sprint – 3	Management	USN – 8	As a user, I can access the site using mobile/ desktop.	3	High	KIRUTHIKA PARAMESHWARI M
Sprint – 4	System	USN – 9	As a user, I must have access to previous usage history.	2	Medium	KAVIPRIYA T
Sprint – 4	System	USN – 10	As a user, I can have audio output as well as text output.	3	High	MADHUMIDHA M

c.Reports from JIRA



7.CODING &SOLUTIONING (Explain the features addedinthe projectalong with code)

a.Model Building

Project: Real-Time Communication system powered by AI for specially abled

Model Building

Import The Required Model Building Libraries

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

In [ ]:
```
#training datagen
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
```

In [ ]:
```
#testing datagen
test_datagen=ImageDataGenerator(rescale=1./255)
```

IMPORTING tensorflow

In [ ]:
```
import tensorflow as tf
import os
```

Initialize The Model

In [ ]:
```
#create model
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [ ]:
```
import numpy as np
import matplotlib.pyplot as plt #to view graph in colab itself
import IPython.display as display
from PIL import Image
import pathlib
```

Unzipping the dataset

In [ ]:

```
#create model
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [ ]:

```
import numpy as np
import matplotlib.pyplot as plt #to view graph in colab itself
import IPython.display as display
from PIL import Image
import pathlib
```

Unzipping the dataset

In [ ]:

```
!unzip '/content/drive/MyDrive/Colab Notebooks/conversation engine for deaf and dumb.zip'
```

```
Streaming output truncated to the last 5000 lines.
 extracting: Dataset/training_set/G/1225.png
 extracting: Dataset/training_set/G/1226.png
 extracting: Dataset/training_set/G/1227.png
 extracting: Dataset/training_set/G/1228.png
 extracting: Dataset/training_set/G/1229.png
  inflating: Dataset/training_set/G/123.png
 extracting: Dataset/training_set/G/1230.png
 extracting: Dataset/training_set/G/1231.png
 extracting: Dataset/training_set/G/1232.png
  inflating: Dataset/training_set/G/1233.png
  inflating: Dataset/training_set/G/1234.png
  inflating: Dataset/training_set/G/1235.png
  inflating: Dataset/training_set/G/1236.png
  inflating: Dataset/training_set/G/1237.png
  inflating: Dataset/training_set/G/1238.png
  inflating: Dataset/training_set/G/1239.png
  inflating: Dataset/training_set/G/124.png
  inflating: Dataset/training_set/G/1240.png
  inflating: Dataset/training_set/G/1241.png
  inflating: Dataset/training_set/G/1242.png
  inflating: Dataset/training_set/G/1243.png
  inflating: Dataset/training_set/G/1244.png
  inflating: Dataset/training_set/G/1245.png
 extracting: Dataset/training_set/G/1246.png
  inflating: Dataset/training_set/G/1247.png
  inflating: Dataset/training_set/G/1248.png
  inflating: Dataset/training_set/G/1249.png
  inflating: Dataset/training_set/G/125.png
  inflating: Dataset/training_set/G/1250.png
  inflating: Dataset/training_set/G/1251.png
  inflating: Dataset/training_set/G/1252.png
  inflating: Dataset/training_set/G/1253.png
  inflating: Dataset/training_set/G/1254.png
  inflating: Dataset/training_set/G/1255.png
```

Applying ImageDataGenerator to training set

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

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

Applying ImageDataGenerator to test set

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

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

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

Length of training set

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

```
79
```

Length of test set

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

```
12
```

Add Layers

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

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

```
79
```

Length of test set

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

```
12
```

Add Layers

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

Add The Convolution Layer

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

Add Pooling Layer

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

Add The Flatten Layer

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

Adding 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'))
```

Adding 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 The Mode

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

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future versio
n. Please use `Model.fit`, which supports generators.
  """Entry point for launching an IPython kernel.
Epoch 1/10
79/79 [==============================] - 89s 1s/step - loss: 0.4318 - accuracy: 0.8572 - val_loss: 0.2326 - val_accuracy: 0.9471
Epoch 2/10
79/79 [==============================] - 87s 1s/step - loss: 0.0413 - accuracy: 0.9886 - val_loss: 0.1702 - val_accuracy: 0.9773
Epoch 3/10
79/79 [==============================] - 88s 1s/step - loss: 0.0253 - accuracy: 0.9933 - val_loss: 0.1599 - val_accuracy: 0.9764
Epoch 4/10
79/79 [==============================] - 86s 1s/step - loss: 0.0086 - accuracy: 0.9979 - val_loss: 0.1979 - val_accuracy: 0.9733
Epoch 5/10
79/79 [==============================] - 87s 1s/step - loss: 0.0097 - accuracy: 0.9975 - val_loss: 0.1815 - val_accuracy: 0.9782
Epoch 6/10
79/79 [==============================] - 86s 1s/step - loss: 0.0067 - accuracy: 0.9982 - val_loss: 0.2445 - val_accuracy: 0.9782
Epoch 7/10
79/79 [==============================] - 84s 1s/step - loss: 0.0045 - accuracy: 0.9988 - val_loss: 0.2291 - val_accuracy: 0.9782
Epoch 8/10
79/79 [==============================] - 84s 1s/step - loss: 0.0083 - accuracy: 0.9973 - val_loss: 0.1956 - val_accuracy: 0.9782
Epoch 9/10
79/79 [==============================] - 83s 1s/step - loss: 0.0029 - accuracy: 0.9995 - val_loss: 0.2011 - val_accuracy: 0.9773
Epoch 10/10
79/79 [==============================] - 84s 1s/step - loss: 0.0027 - accuracy: 0.9991 - val_loss: 0.2367 - val_accuracy: 0.9778
```

Out[ ]:

Save The Model

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

8.TESTING

a.Test Cases

Adding 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 The Mode

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

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future versio
n. Please use `Model.fit`, which supports generators.
  """Entry point for launching an IPython kernel.
Epoch 1/10
79/79 [==============================] - 89s 1s/step - loss: 0.4318 - accuracy: 0.8572 - val_loss: 0.2326 - val_accuracy: 0.9471
Epoch 2/10
79/79 [==============================] - 87s 1s/step - loss: 0.0413 - accuracy: 0.9886 - val_loss: 0.1702 - val_accuracy: 0.9773
Epoch 3/10
79/79 [==============================] - 88s 1s/step - loss: 0.0253 - accuracy: 0.9933 - val_loss: 0.1599 - val_accuracy: 0.9764
Epoch 4/10
79/79 [==============================] - 86s 1s/step - loss: 0.0086 - accuracy: 0.9979 - val_loss: 0.1979 - val_accuracy: 0.9733
Epoch 5/10
79/79 [==============================] - 87s 1s/step - loss: 0.0097 - accuracy: 0.9975 - val_loss: 0.1815 - val_accuracy: 0.9782
Epoch 6/10
79/79 [==============================] - 86s 1s/step - loss: 0.0067 - accuracy: 0.9982 - val_loss: 0.2445 - val_accuracy: 0.9782
Epoch 7/10
79/79 [==============================] - 84s 1s/step - loss: 0.0045 - accuracy: 0.9988 - val_loss: 0.2291 - val_accuracy: 0.9782
Epoch 8/10
79/79 [==============================] - 84s 1s/step - loss: 0.0083 - accuracy: 0.9973 - val_loss: 0.1956 - val_accuracy: 0.9782
Epoch 9/10
79/79 [==============================] - 83s 1s/step - loss: 0.0029 - accuracy: 0.9995 - val_loss: 0.2011 - val_accuracy: 0.9773
Epoch 10/10
79/79 [==============================] - 84s 1s/step - loss: 0.0027 - accuracy: 0.9991 - val_loss: 0.2367 - val_accuracy: 0.9778
```

Out[ ]:

a.User Acceptance Testing

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] 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 level, 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 shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	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

a.<u>Performance Metrics</u>

10.<u>ADVANTAGES & DISADVANTAGES</u>

a.Advantages:

- • It enables workers from all over the world to connect with one another 24 hours a day and share ideas or swiftly solve problems.
- • It is an affordable solution to bring multiple people from different locations to meetings and conferences.

b.Disadvantages:

- Also accuracy dependsupon distance betweencamera andobject.
- It takes a lot of time to listen, speak, read, or write to someone.

11.CONCLUSION

- When bridging the communication gap between two cultures, the proposed communication method between Deaf and Dumb persons and regular people is aiming for it. It enables effective two-way communication between the able-bodied and able-bodied individuals.
- A translator of sign language is needed to facilitate communication between a deaf person and a second person. A mediator must, however, be familiar with the deaf person's sign language. However, given that there are numerous sign languages for numerous languages, this is not always practical.
- This technique is therefore suggested in order for hearing individuals to understand all sign languages and hand gestures used by deaf persons.

12.FUTURE SCOPE

- The speech-to-text and text-to-speech technologies aided persons who had trouble expressing themselves or communicating with others in a usual way.
- As a result, there is less of a communication barrier between those with special needs and those without.
- It is simple to comprehend the context of items and clearly explains it to the individuals who use it for communication using picture pre-processing and artificial intelligence.

13.APPENDIX

a.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>
```

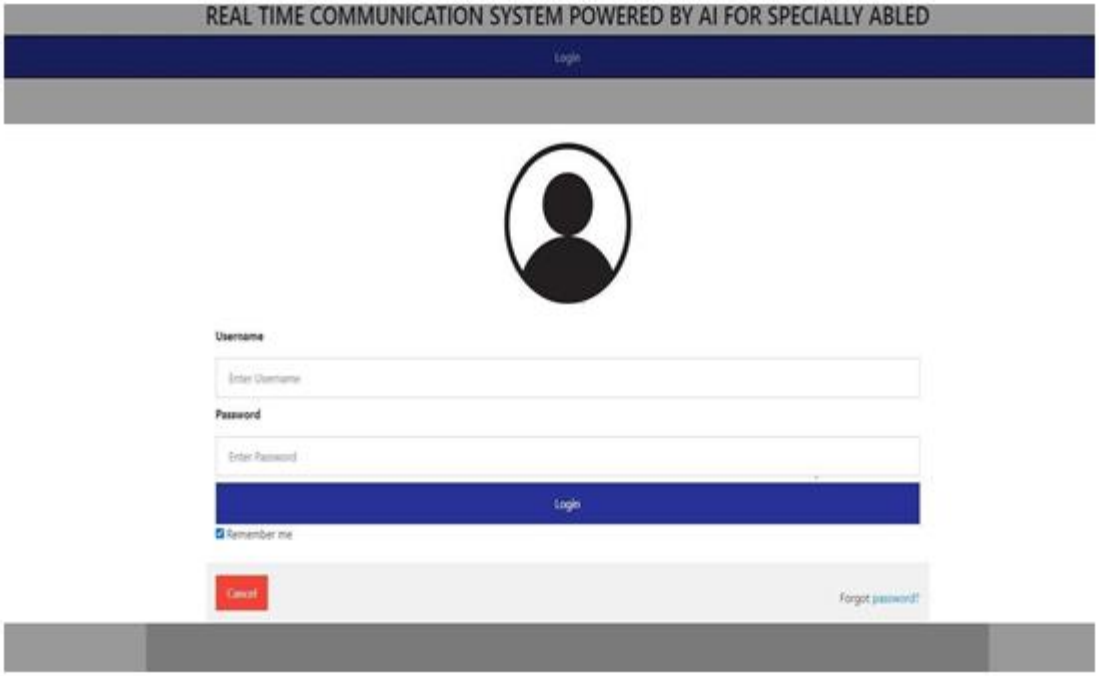
```
<!DOCTYPE html>
<html lang="en">

<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>
```

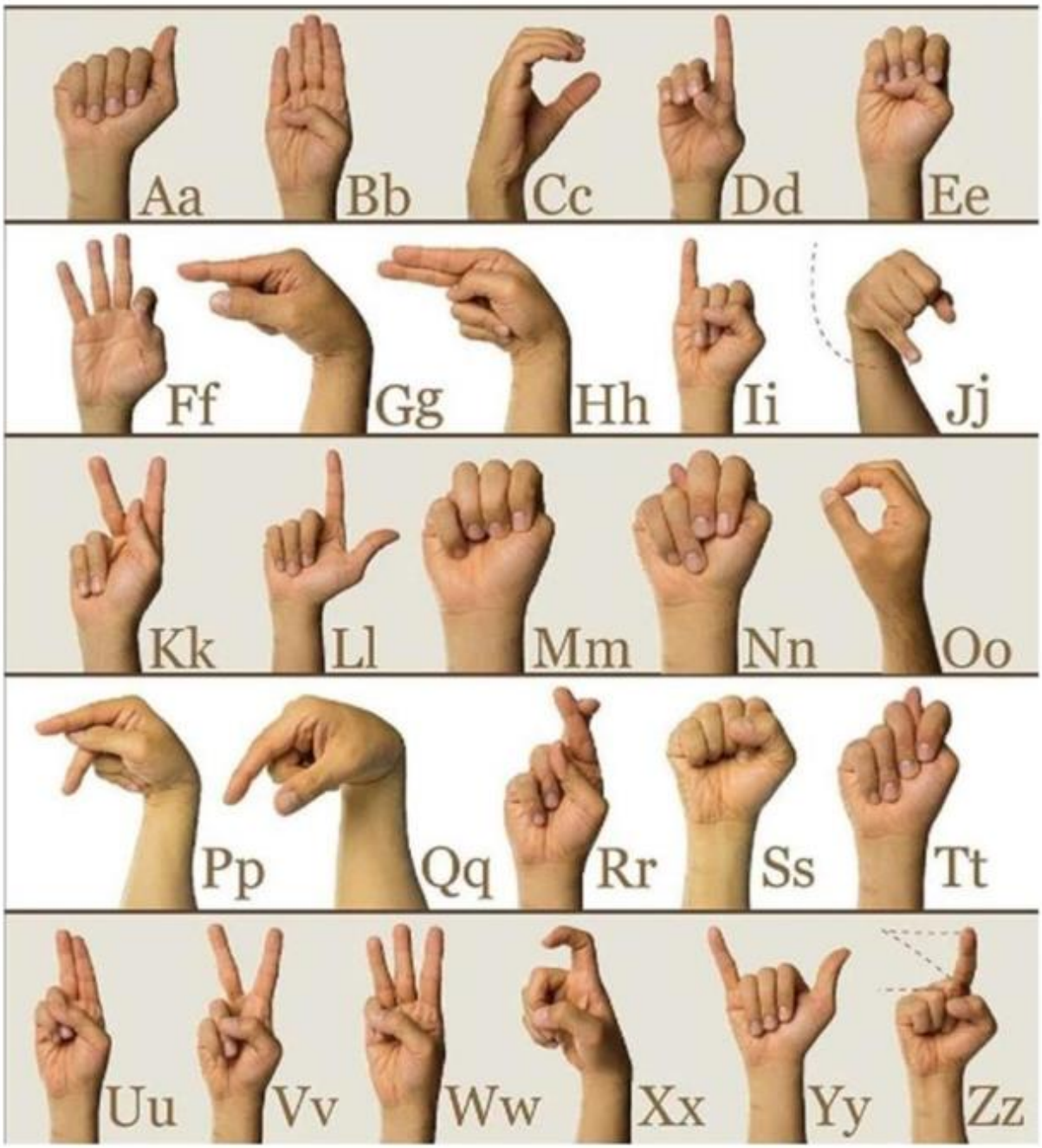
```
        <div class="container">
            <div class="accordion text-white" role="tablist" id="accordion-1">
                <div class="accordion-item" style="background: rgb(33,37,41);">
                    <h2 class="accordion-header" role="tab"><button class="accordion-button" data-bs-toggle="collapse"
                            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.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%;
}
```




Aa
Bb
Cc
Dd
Ee
Ff
Gg
Hh
Ii
Jj
Kk
Ll
Mm
Nn
Oo
Pp
Qq
Rr
Ss
Tt
Uu
Vv
Ww
Xx
Yy
Zz

14.SOURCECODE AND GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-41008-1660638331