

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
1	INTRODUCTION	3
	1.1 Project Overview	3
	1.2 Purpose	3
2	LITERATURE SURVEY	4
	2.1 Existing problem	4
	2.2 References	4
	2.3 Problem Statement Definition	8
3	IDEATION & PROPOSED SOLUTION	9
	3.1 Empathy Map Canvas	9
	3.2 Ideation & Brainstorming	9
	3.3 Proposed Solution	11
	3.4 Problem Solution fit	12
4	REQUIREMENTS ANALYSIS	13
	4.1 Functional requirement	13
	4.2 Non- Functional requirement	13
5	PROJECT DESIGN	15
	5.1 Data Flow Diagrams	15
	5.2 Solution & Technical Architecture	15
	5.3 User Stories	16

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

6	PROJECT PLANNING & SCHEDULING		19
	6.1	Sprint Planning & Estimation	19
	6.2	Sprint Delivery schedule	21
	6.3	Reports from JIRA	22
7	CODING & SOLUTIONING		21
	7.1	Feature 1	24
	7.2	Feature 2	24
8	TESTING		25
	8.1	Test Cases	25
	8.2	User Acceptance Testing	25
	RESULTS		28
9	9.1	Performance Metrics	28
10	ADVANTAGES & DISADVANTANGES		30
11	CONCLUSION		31
12	FUTURE SCOPE		32
13	APPENDIX		33
	13.1	Source Code	33
	13.2	GitHub & Project Demo Link	42

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

CHAPTER-1 INTRODUCTION

1.1 PROJECT OVERVIEW

It is an online software for real time communication for deaf and dumb with the help of convolutional neural networks and tensorflow keras mechanism. It is designed and created for the users to translate and learn American Sign Language (ASL).

1.2 PURPOSE

The purpose of Sign Language Recognition (SLR) systems is to provide an efficient and accurate way to convert sign language into text or voice has aids for the hearing impaired for example, or enabling very young children to interact with computers (recognizing sign language), among others.

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

CHAPTER-2 LITERATURE SURVEY

2.1 EXISTING PROBLEM

The challenge faced by dumb and deaf people while communicating with the system in work place , since they cannot hear it, dangerous to go places alone because they cannot hear car, bikes, or other people coming. Dumb people use hand signs to communicate, hence normal people face problem in recognizing their language by signs made. Hence there is a need of the systems which recognizes the different signs and conveys the information to the normal people.

2.2 REFERENCES

1. . Based Real Time Communication for Physically and Speech Disabled People (Ong Chin Ann, Marlene Valeriu Lu – 2019)

Communication is a social process of exchanging information from one entity to another in verbal and non-verbal form. It defines our existence and it is an important instrument that connects people together. It comes naturally as a raw skill embedded in most people at birth and we acquired the ways of communication through cognitive learning. Communication is the basis, which drives the process of development in all the fields (Manohar, 2008) .

2. Systematic review of computer vision semantic analysis in medical (Antonio Victor Alencar Lundgren, Byron Leite Dantas Bezzerra – 2021)

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

Medical diagnosing techniques have fascinated us for a long time. It has been common for us to use them in our daily life and implement these technologies. Machine learning and especially computer vision contribute a lot in medical science, which make different difficult tasks easy for doctors and more tolerable for patients. They are widely useful in early detection of disease, and hence are a valuable tool to save human life. Cardio graphic techniques are a must for old age and infant safety.

3. A survey on Facial Emotion Recognition Techniques (Felipe Zago Canal, Tobias Rossi Muller, Gustavo Gino Scotton – 2022)

Facial expressions recognition is an ability to recognize people by their facial characteristic and differentiate it with one another. Human is born with the ability to recognize other people easily by identifying their facial features such as shape, appearance, skin texture and skin complexion. Other than that, humans also have the ability to express, interpret and differentiate facial expressions. The regular recurring ones are happiness, anger, disgust, fear, surprise and sad (Ekman & Friesen, 1978). The six facial emotions stated above are important and play a major role in expressing emotion as well as recognising facial expression (Busso, et al, 2004). In real life, inter personal human interaction are performed not only using speech or spoken language, but also non verbal cues for example hand gesture, body gesture, facial expression and tone of the voice.

4. Machine Learning based techniques in data analysis (Lavanya Vemulapalli, Dr.P.Chandra Sekhar – 2018)

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

A lot more applications available for us in play store, app store, amazon, etc., which are dependent machine learning. There are significant number of organizations and startups which turn towards optimum machine learning, and have proved that investing in machine learning is the best in today's world. It is an application from which we can virtually explore streets of cities. It uses a dense geosampling tool to show the streets of cities. Streets are captured through a fleet of vehicles equipped with a specialized camera. After collection of photos, they are digitally processed and combined together and look like a single image. From files reported for privacy, Google pixelated faces of pedestrian and license plate which is captured. Web mapping technologies have been embraced by discipline such as geography, archeology and ecology, but also by several social scientific disciplines.

5. . Survey on Machine Learning Algorithm's (Rekha Nagar, Dr. Yudhvir Singh – 2022)

The subfield of artificial intelligence, machine learning has gained much popularity in last few couple of years. Many tech giants use machine learning algorithms, like Netflix's algorithms to make movie prediction from your previous watched movies. In this section, we would like to present some of the famous algorithms which use frequently.

They are :

- **Naïve-Bayes' algorithm** - This is the algorithm mostly used in machines and hardware. It simply applies Bayes' theorem along with strong independence assumptions. Let's take an example, to mark an email as

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

spam, used for face detection software, etc.

- **k-means clustering algorithm** - This is a type of unsupervised learning which has various uses including business and management. This algorithm also lets us know profit at each stage of the product. It is also referred as Lloyd's algorithm. This algorithm is also used in grouping of features into different labels.
- **Neural Network** - Our neurons in body play a major role in determining the steps to process a single task. Similarly, artificial neurons are those which help the nervous system of transistors in any sequential or combinational circuit to take up a decision and execute it conditionally. This again depends on activity of the neurons. An artificial neuron is an actual piece of hardware machinery which help the system to take up a decision based on the receptors, as such several optoelectronic devices have already been developed. This algorithm helps us to build any machine functioning exactly as human reflex arcs.

6. Survey on application of Artificial Intelligence in Cyber Security (Shidawa Baba Atiku, Achi Unimke Aaron, Fatima Shittu – 2020)

Cyber security refers to protecting your personal computer from malicious software. Machine learning has a lot many algorithms and systems which protect users from threats. Such as the Paypal app which was developed in December 1998, uses machine learning algorithms to protect its users from different threats and online spoofing. It uses three types of machine learning algorithms that are linear, neural network, and deep learning algorithm.

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

They are :

- **Waterhole** - It is like a pit surrounded by greenery. Hackers access other people's information by using sites which are more accessible to the public more than anything else.... for example, networks in a coffee shop is accessed by so many users such that these users load their pc 's with whatsoever data is provided to them. Like this there are so many sites to put on viruses and worms. Machine learning have algorithms that detect path of these malware blocking them with a firewall thereafter.
- **Webshell** - These are piece of code which are loaded into a working device which provokes the user to misjudge and then taking advantage, entry is gained into the full database.
- **Ransomware** - Similar to webshell, but here the user is vulnerably threatened externally by a group of software brokers who have corrupted the users' personal files. Such scenarios can be totally avoided by using machine level language which was early detection.

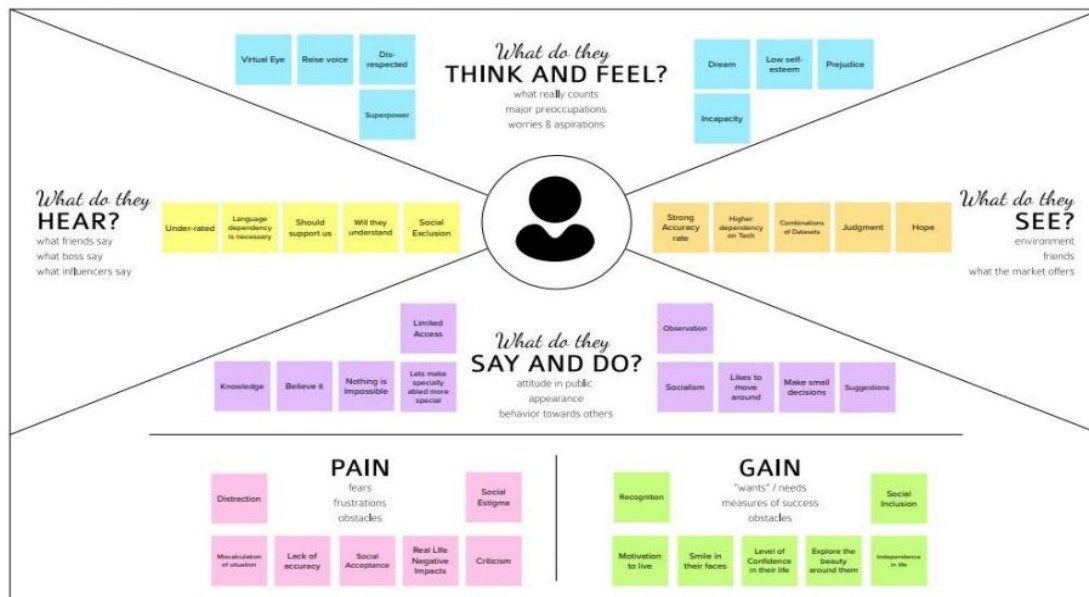
2.3 PROBLEM STATEMENT DEFINITION

There have been several advancements in technology and a lot of research has been done to help the people who are deaf and dumb. Aiding the cause, Deep learning, and computer vision can be used too to make an impact on this cause. This can be very helpful for the deaf and dumb people in communicating with others as knowing sign language is not something that is common to all, moreover, this can be extended to creating automatic editors, where the person can easily write by just their hand gestures.

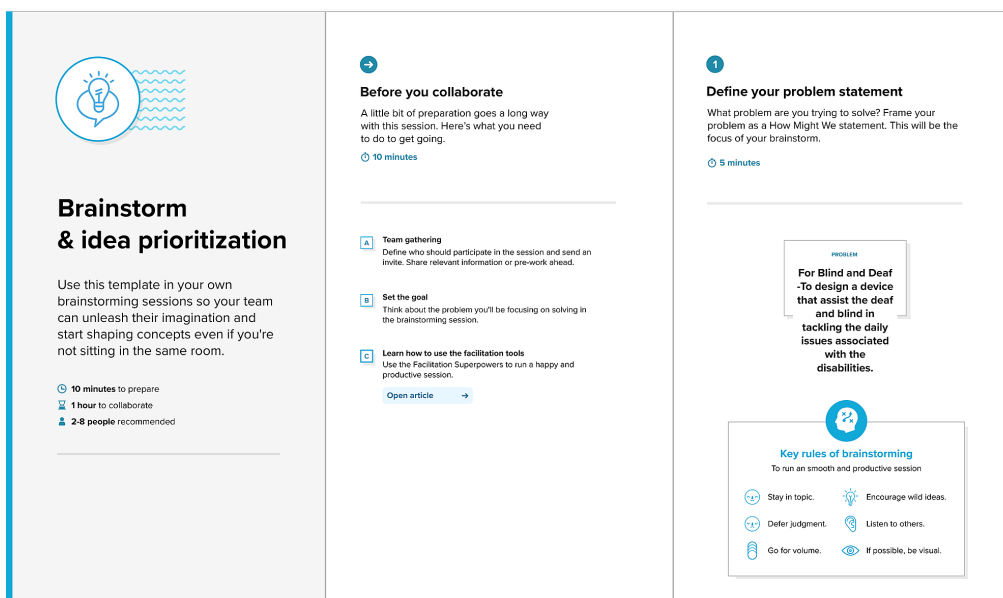
REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

CHAPTER-3 IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP



3.2 IDEATION AND BRAINSTROMING



REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

2

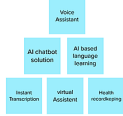
Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP
You can select a sticky note and hit the pencil (switch to sketch) icon to start drawing!

Siva Vinel Rajhen



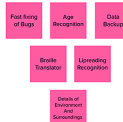
Subiksha S



Praveen Kumar S



Priyasha S



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes

COMPUTER VISION



TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas in themes within your mind.

ACCESSIBILITY FOR THE HEARING IMPAIRED AND DEAF



ACCESSIBILITY FOR THE VISUALLY IMPAIRED AND BLIND



FOR BOTH BLIND AND DEAF



USER FLEXIBILITY



SECURITY

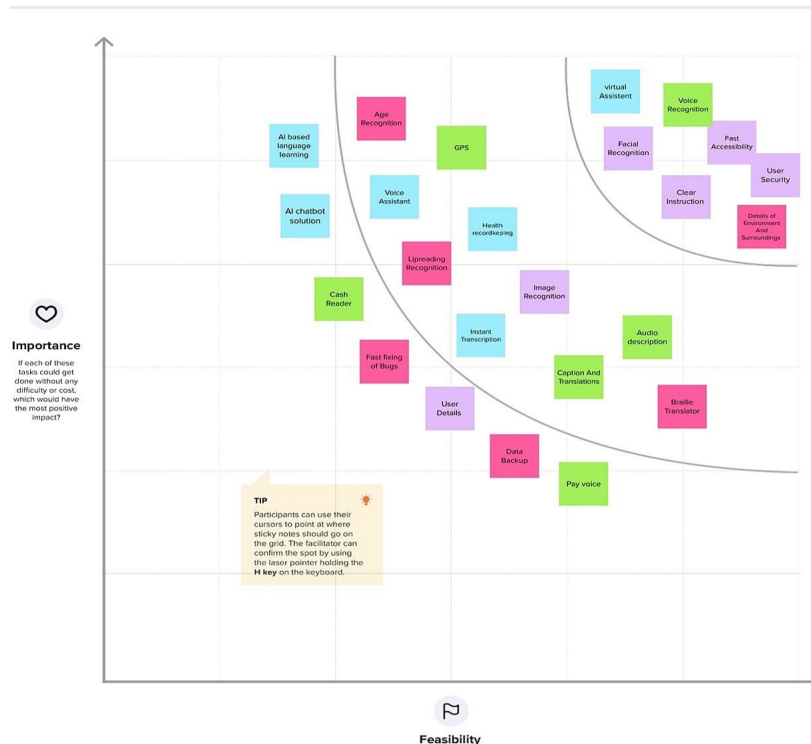


4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes



REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	For a visually impaired to detect their surrounding and for a hearing impaired to understand what someone is saying or to hear is impossible without help. There is a need to understand their social environment for the need of safety and independence from a care taker.
2.	Idea / Solution description	Instead of depending on a guide dog or a caretaker, we introduce an application to perform the role of an eye and ear.
3.	Novelty / Uniqueness	Real-time fast image recognition and speech recognition along with text-to-speech and speech-to-text transmission. Object detection, copy or translation of text, find visually similar image.
4.	Social Impact / Customer Satisfaction	Independent travel is an essential part of daily life for many people who are visually impaired or hearing impaired, but they face a greater risk of bumping into obstacles when they walk on their own.
5.	Business Model (Revenue Model)	With the need of huge data to analyse at real-time, we use cloud storage for preserving the data.
6.	Scalability of the Solution	Every visually impaired and hearing impaired person can use this application to solve their daily life crisis, without depending on another.

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

3.4 PROBLEM SOLUTION FIT

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) Customers are specially abled peoples CS	6. CUSTOMER CONSTRAINTS As originally defined in artificial intelligence, constraints enumerate the possible values a set of variables may take in a given world. A possible world is a total assignment of values to variables representing a way the world (real or imaginary) could be. CC	5. AVAILABLE SOLUTIONS Polly can attach to any wheelchair or bedside, track eye movement and use ML to assist smart prediction of the user's needs and wants. AS	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS AI can be trained to do text summarization, which is helpful for users with cognitive impairments; or to do image and facial recognition, which helps those with visual impairments; or real-time captioning, which helps those with hearing impairment. J&P	9. PROBLEM ROOT CAUSE Notwithstanding the tangible and monetary benefits, AI has various shortfalls and problems which inhibits its large scale adoption. The problems include Safety, Trust, Computation Power, Job Loss concern. RC	7. BEHAVIOUR Artificial intelligence is learning more about how to work with (and on) humans. A recent study has shown how AI can learn to identify vulnerabilities in human habits and behaviours and use them to influence human decision-making BE	
Focus on J&P, tap into BE, understand RC	3. TRIGGERS which are made to interact with humans, can help teach social and educational skills to students of all abilities, including those with attention deficit hyperactivity disorder, hearing impairments, Down syndrome and autism TR	10. YOUR SOLUTION AI technology can empower people living with limited physical mobility. Microsoft's AI for Accessibility program uses the potential of Artificial Intelligence to develop solutions to many physical and cognitive challenges disabled individuals face at work and in daily life to promote social inclusion for them SL	8. CHANNELS of BEHAVIOUR the exploitation of human biases detected by AI algorithms, personalised addictive strategies for consumption of (online) goods, or taking advantage of the emotionally vulnerable state of individuals to promote products and services that match well with their temporary emotions CH	Focus on J&P, tap into BE, understand RC
	4. EMOTIONS: BEFORE / AFTER Nearly two-thirds of people surveyed said they have a good understanding of what artificial intelligence is. Some 60% of people also think that products and services using AI will make their lives easier, with 60% also expecting AI to profoundly change their daily life in the coming years. EM			
Identify strong TR & EM				

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

CHAPTER-4 REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through phone number
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login Using Credentials
FR-4	Authentication	User when he/she tries to log into the system
FR-5	Reporting	Any problems in the application will automatically reported to the developer.
FR-6	Compliance to rules or Laws	Terms and conditions ,Privacy policy, End user licensing agreement

4.2 NON - FUNCTIONAL REQUIREMENT

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Goals are easy to accomplish quickly and with few or no user errors

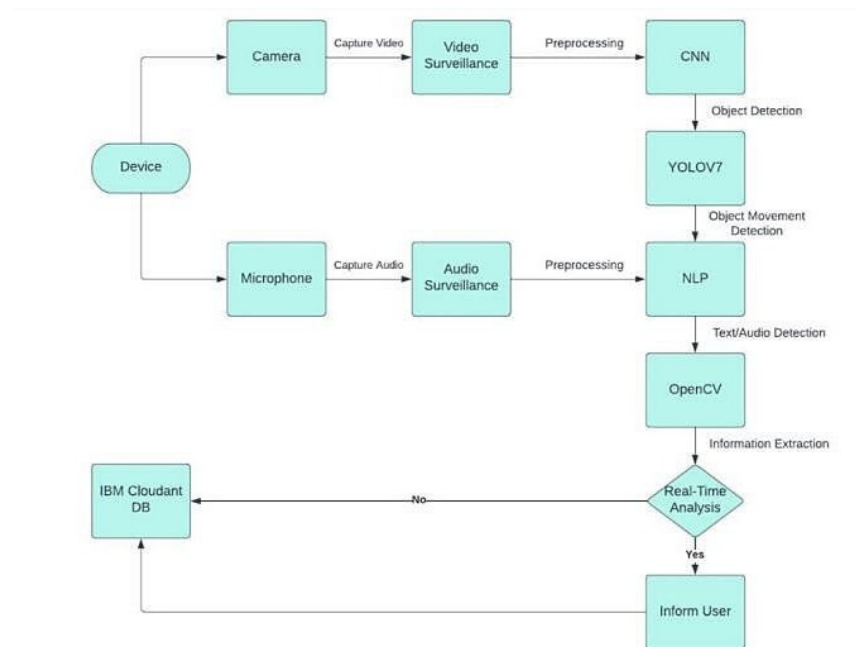
REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

NFR-2	Security	Our device shutdowns automatically when detecting a potential security threat
NFR-3	Reliability	The device must perform without failure in 95% of use cases
NFR-4	Performance	Our device works as quickly as possible to benefit our users
NFR-5	Availability	It will be offered. 24/7
NFR-6	Scalability	Our device is more than capable of supporting numerous tasks concurrently while maintaining a primary task

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

CHAPTER-5 PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

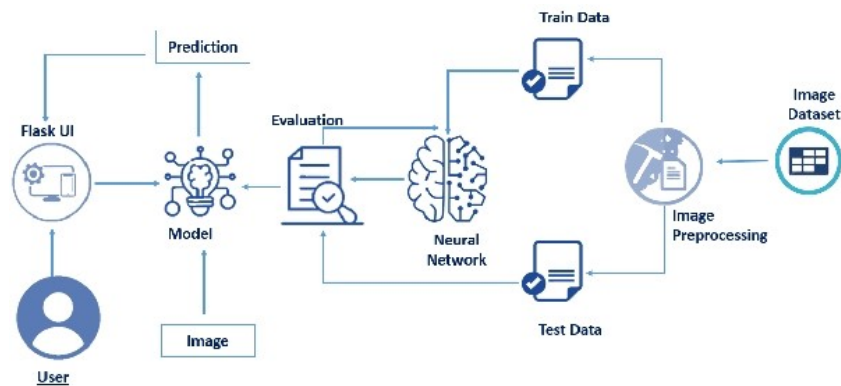


Fig 5.2.1 Solution Architecture

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

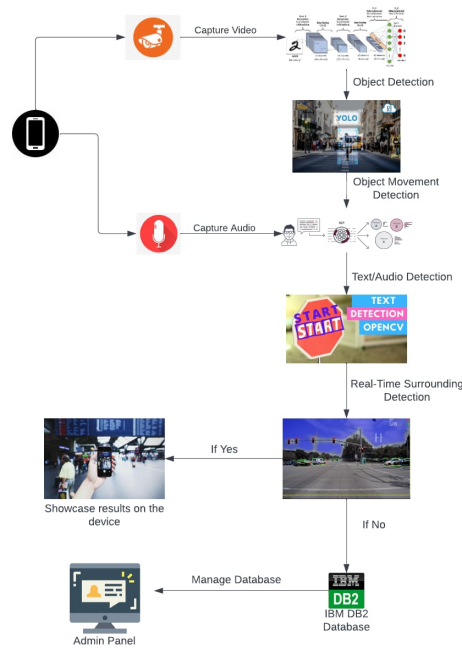


Fig 5.2.2 Technical Architecture

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 usage reason	Camera & microphone are setup according to preference	High	Sprint-1

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

			as for visually impaired or hearing impaired			
	Pre-processing	USN-2	Train and test the model	Train the model according to the datasets received	High	Sprint-1
	Real-Time Surrounding activity detection	USN-3	The user with visually impaired will be informed with activities happening around him/her via their device	Camera surveillance	High	Sprint-1
	Real-Time Surrounding audio detection	USN-4	The user with hearing impaired will be informed with speech to text functionality via their device	Audio surveillance	High	Sprint-1
	Personal Voice Assistant	USN-5	To assist the user who is visually impaired	Alert the user	High	Sprint-2
	Personal Text	USN-6	To assist the user who is hearing	Alert the user	High	Sprint-2

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

	Assistant		impaired			
	User Understanding	USN-7	The user understands the surrounding activity	Understanding	High	Sprint-2
Administrator	Register	USN-8	Register into the application	Admin can access the account	Medium	Sprint-3
	Login	USN-9	Login and manage the application	Manage application	Medium	Sprint-3
		USN-10	Store's in the database	Data Storage	Medium	Sprint-3

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

CHAPTER-6

PROJECT PANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Function al Require ment (Epic)	User Sto ry Num ber	User Story / Task	Sto ry Poi nts	Priori ty	Team Membe rs
Sprint-1	Registrat ion	US N-1	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Subiksha S
Sprint-1		US N-2	As a user, I can register for the application through mobile number.	1	High	Priyasha S
Sprint-1		US N-3	As a user, I can register for the application through Gmail	2	Medi um	Praveen kumar S
Sprint-1	Login	US N-4	As a user, I can log into the application by entering email & password	3	High	Subiksha S

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

Sprint-1	User Details	US N-5	As a user, I can give my details.	2	Medium	Siva vimel Rajhen
Sprint-2	Collection of Data	US N-6	As a admin, I can collect the Data from different sources could be internal and/or external to satisfy the requirements/problems.	8	Medium	Subiksha S
Sprint-2	Clean the data	US N-7	As a admin, I can clean dataset to improve the model	5	Medium	Praveen Kumar S
Sprint-2	Speech / Image Pre-processing	US N-8	As a admin, I can Perform Pre-processing techniques on the dataset	8	Medium	Siva Vimel Rajhen
Sprint-3	Training and testing data	US N-9	As a admin, I can train the model and to test its efficiency and performance of the model	5	High	Priyasha S
Sprint-3	Build model	US N-10	As a admin, I can build a model for selecting correct ML algorithm	13	High	Siva Vimel Rajhen
Sprint-3	Make predictions	US N-11	As a admin, I can Use algorithms to cast predictions and forecast trends	8	High	Praveen Kumar S

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

Sprint-4	Evaluate and improve	US N-12	As a admin, I can test the accuracy and precision of the model and Improving the performance	8	High	Priyasha S
Sprint-4	Model Deploy	US N-13	Deploying ML model in production	5	High	Siva Vimel Rajhen

6.2 SPRINT DELIVERY SCHEDULE

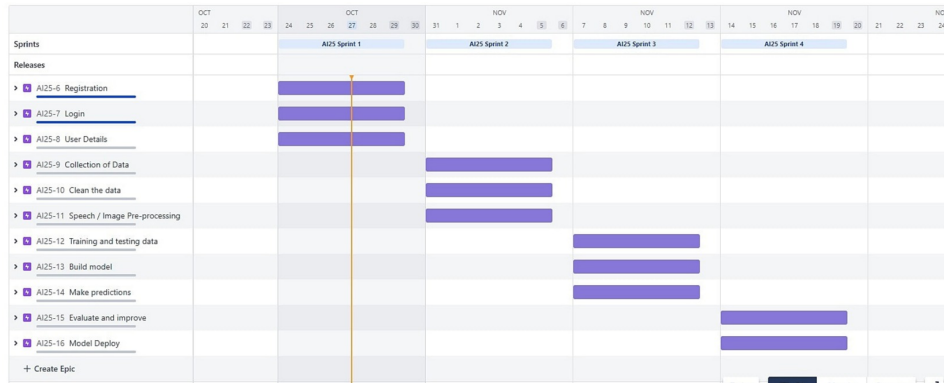
Spri nt	Total Story Poin ts	Dura tion	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Spri nt-1	13	6 Days	24 Oct 2022	29 Oct 2022		
Spri nt-2	21	6 Days	31 Oct 2022	05 Nov 2022		
Spri nt-3	26	6 Days	07 Nov 2022	12 Nov 2022		
Spri nt-4	13	6 Days	14 Nov 2022	19 Nov 2022		

6.3 REPORT FROM JIRA

It shows time taken to completed status of all the issues in the sprint.

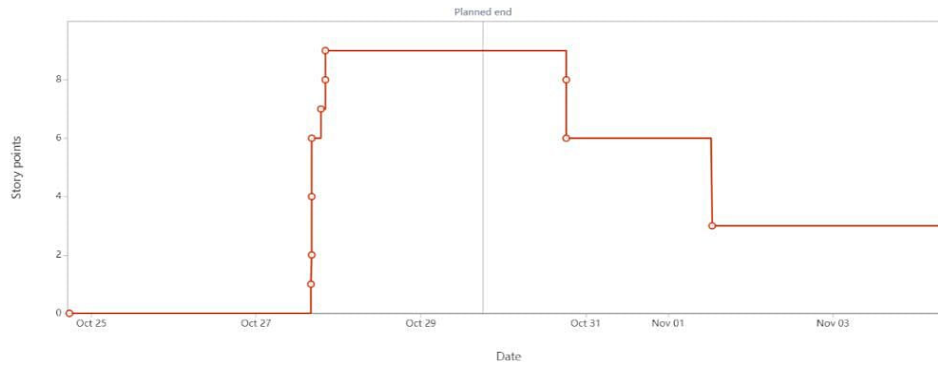
REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

ROADMAP

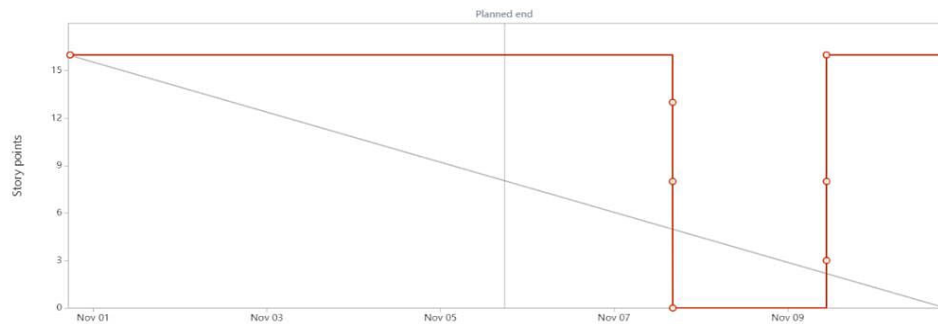


Burndown Chart

Sprint 1

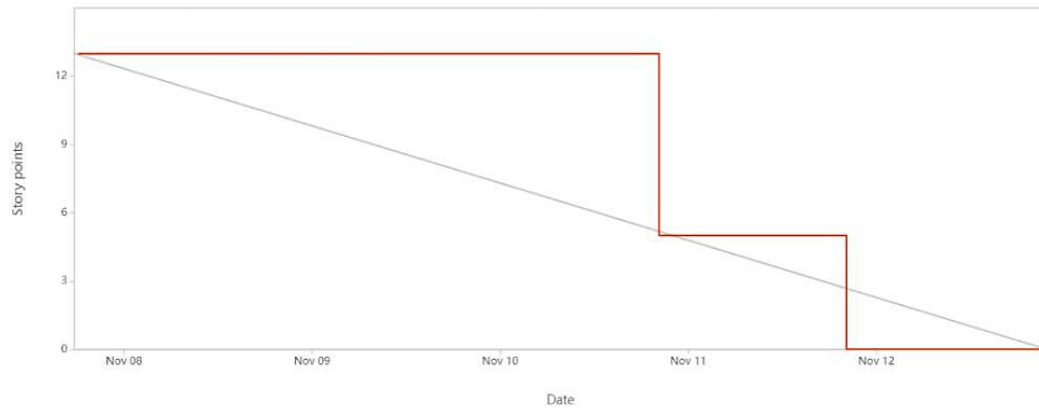


Sprint 2

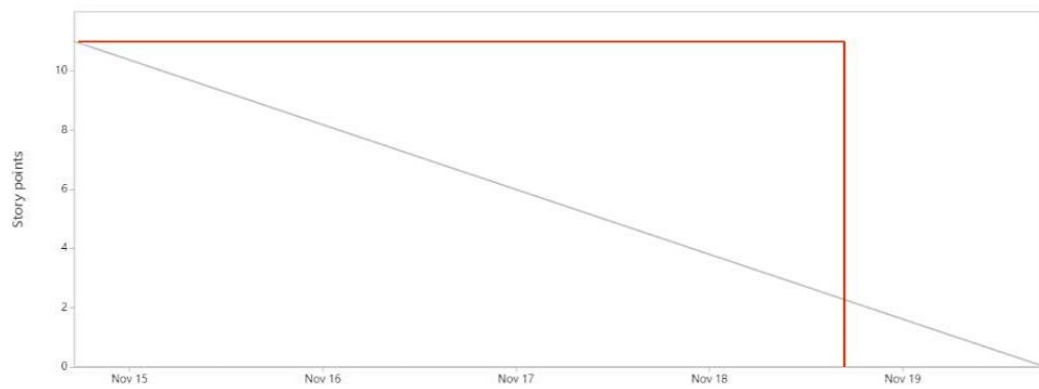


REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

Sprint 3



Sprint 4



REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

CHAPTER-7 CODING & SOLUTION

7.1 FEATURE 1

Sign language recognition includes two main categories, which are isolated sign language recognition and continuous sign language recognition. The supervision information is a key difference between the two categories. While isolated sign language recognition is similar to the action recognition area, the continuous sign language recognition concerns about not only the recognition task but also the accurate alignment between the input video segments and the corresponding sentence-level labels.

7.2 FEATURE 2

American Sign Language, also known as ASL, is a form of communication that contains a complete vocabulary and grammar but is expressed through physical movements of the hands and arms rather than speech. ASL offers an option for deaf, hearing-impaired, and hearing individuals to communicate with each other. In this blog, we will discuss a few interesting facts about ASL.

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

CHAPTER-8 TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
HomePage_TC_001	UI	Home Page	Verify user is able to see the Login/Signup popup when user clicked on the URL.	OS, Browser, Internet Connection	1.Enter URL and click go 2.Verify login/Signup displayed or not?	http://127.0.0.1:5000/	Login/Signup should display	Working as expected	Pass	NA	N	NA	Siva Vimeel Rajhen
HomePage_TC_002	UI	Home Page	Verify the UI elements in Login/Signup page	OS, Browser, Internet Connection	1.Enter URL and click go 2.Verify login/Signup page with below UI elements: a.email test box b.password test box c.Login button d.New? Create account link e.Last password? Recovery password link	http://127.0.0.1:5000/login	Application should show below UI elements: a.email test box b.password test box c.Login button d.New? Create account link e.Last password? Recovery password link	Working as expected	Pass	NA	N	NA	Siva Vimeel Rajhen
LoginPage_TC_003	UI	Home page	Verify user is able to log into application with valid credentials	OS, Browser, Internet Connection	1.Enter URL and click go 2.Click on Login option 3.Enter valid email in Email test box 4.Enter valid password in password test box 5.Click on login button	Email: siva28@gmail.com password: Testing123	User should navigate to user account login page	Working as expected	Pass	NA	N	NA	Siva Vimeel Rajhen
LoginPage_TC_004	Functional	Login page	Verify user is able to log into application with invalid credentials	OS, Browser, Internet Connection	1.Enter URL and click go 2.Click on Login option 3.Enter invalid email in Email test box 4.Enter valid password in password test box 5.Click on login button	Email: siva28@gmail.com password: Testing123	Application should show 'Incorrect email or password' validation message.	Working as expected	Pass	NA	N	NA	Siva Vimeel Rajhen

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
SignIn_Page_TC_005	Functional	SignIn Page	Verify the user redirected to the page which shows prediction	OS, Browser, Internet Connection	1.Enter URL and click go 2.Click on Login option 3.Enter valid email in Email test box 4.Enter valid password in password test box 5.Click on login button 6. As sign language page will display	Click the button	Application should navigate the next page	Working as expected	Pass	NA	N	NA	Siva Vimeel Rajhen
AslPage_TC_006	Functional	Prediction page	Verify user webcam is properly stream frames	OS, Browser, Internet Connection, web cam	1.Enter URL and click go 2.Click on Login option 3.Enter valid email in Email test box 4.Enter valid password in password test box 5.Click on login button 6. As sign language page will display 7. Webcam will Turn on 8. User's video will display on the page	Give access for webcam	User's webcam should turn on and capture the frames	Working as expected	Pass	NA	N	NA	Siva Vimeel Rajhen
AslPage_TC_007	Functional	Prediction page	Verify Hand Sign Predicting	OS, Browser, Internet Connection, web cam	1.Enter URL and click go 2.Click on login button 3. Click to Sign to ASL page 4. Webcam will Turn on 5. User's video will display on the page 6. Show Hand inside the rectangle box 7. Sign letter will be predicted	Show hand inside the rectangle box	Detect the Hand sign letter present inside the rectangle box	Working as expected	Pass	NA	N	NA	Siva Vimeel Rajhen
AslPage_TC_008	Functional	Prediction page	Verify Predicted sign in translated in Text	OS, Browser, Internet Connection, web cam	1.Enter URL and click go 2.Click on login button 3. Click to Sign to ASL page 4. Webcam will Turn on 5. User's video will display on the page 6. Show Hand inside the rectangle box 7. Sign letter will be predicted 8. Displays the letter in the webcam screen	Shows predicted Hand Sign	Detect the Predicted sign and convert into Text	Working as expected	Pass	NA	N	NA	Siva Vimeel Rajhen

8.2 USER ACCEPTANCE TESTING

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).

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

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	7	4	2	3	16
Duplicate	2	0	3	0	5
External	3	3	0	1	7
Fixed	5	2	4	12	23
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	0	0	0	0
Totals	17	9	11	17	54

Test Case 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	7	4	2	3	16
Duplicate	2	0	3	0	5
External	3	3	0	1	7
Fixed	5	2	4	12	23
Not Reproduced	0	0	1	0	1

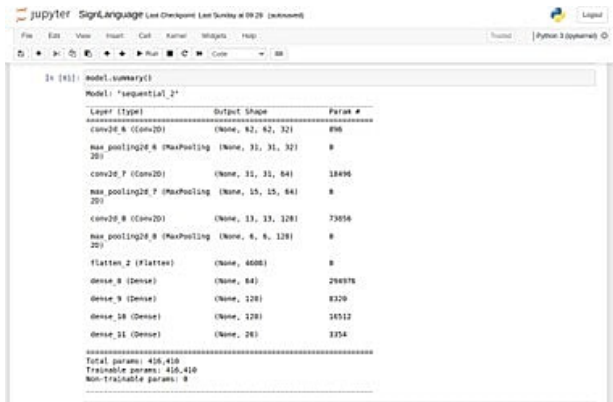
**REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI
FORSPECIALLY ABLED**

Skipped	0	0	1	1	2
Won't Fix	0	0	0	0	0
Totals	17	9	11	17	5 4


REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

CHAPTER-9 RESULTS

9.1 PERFORMANCE METRICS

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Total params:416,410 Trainable params:416,410 Non-Trainable params:0	 <pre> In [81]: model.summary() Model: "sequential_2" Layer (type) Output Shape Param # ----- conv2d_6 (Conv2D) (None, 82, 82, 32) 896 max_pooling2d_6 (MaxPooling2D) (None, 41, 41, 32) 0 conv2d_7 (Conv2D) (None, 41, 41, 64) 18496 max_pooling2d_7 (MaxPooling2D) (None, 20, 20, 64) 0 conv2d_8 (Conv2D) (None, 20, 20, 128) 73856 max_pooling2d_8 (MaxPooling2D) (None, 10, 10, 128) 0 flatten_2 (Flatten) (None, 6000) 0 dense_9 (Dense) (None, 84) 294376 dense_10 (Dense) (None, 128) 8320 dense_11 (Dense) (None, 128) 16312 dense_12 (Dense) (None, 28) 3724 ----- Total params: 416,410 Trainable params: 416,410 Non-trainable params: 0 </pre>

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

2.	Accuracy	<p>Training Accuracy – 100.0%</p> <p>Validation Accuracy – 1.0</p>	 <pre> In [14]: imgs, labels = next(test_batches) In [15]: scores = model.evaluate(imgs, labels, verbose=0) print('Model metrics names: %s' % (model.metrics_names[0] if (scores[0]*100)%1 loss of 0.000432923756380991; accuracy of 100.0% In [16]: model.save('best_model_data\43.15') In [17]: model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy']) reduce_lr = ReduceLRPlateauMonitor(val_loss = 'reduce_lr', patience=2, min_lr=0.0001) early_stop = EarlyStoppingMonitor(val_loss = 'val_loss', patience=2, min_lr=0.0001, mode='auto') In [18]: history2 = model.fit(train_batches, epochs=10, callbacks=[reduce_lr, early_stop], validation_data = test_batches) Epoch 1/10 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v Epoch 2/10 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v Epoch 3/10 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v Epoch 4/10 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v Epoch 5/10 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v Epoch 6/10 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v Epoch 7/10 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v Epoch 8/10 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v Epoch 9/10 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v Epoch 10/10 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v 2011/1111 [=====] - 756 24ms/step - loss: 0.0078 - accuracy: 0.9638 - val_loss: 0.0013 - v In [19]: imgs, labels = next(test_batches) </pre>
----	----------	--	--

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

CHAPTER-10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

Sign language allows deaf and hard of hearing people to communicate quickly and effectively with others who use sign language, or who "sign." Most deaf people use a combination of sign language, lip-reading and written communication to go about their daily lives. Many resources have been developed in America to help deaf people who have live normal lives. Today, ASL is one of the fastest growing language being taught on college campuses.

DISADVANTAGES

Sign language requires the use of hands to make gestures. This can be a problem for people who do not have full use of their hands. Even seemingly manageable disabilities such as Parkinson's or arthritis can be a major problem for people who must communicate using sign language. Having a broken arm or carrying a bag of groceries can, for a deaf person, limit communication. The amount of light in a room also affects the ability to communicate using sign language.

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

CHAPTER-11

CONCLUSION

The aim of this project is to predict the ASL alphanumeric hand-gestures in real time. The above work shows that it can be solved with better accuracy when we actually consider the segmented RGB hand-gestures. By applying depth-based segmentation we remove the overheads of dynamic background. The segmented gray scale hand-gestures were fed to 3 layered CNN for training and testing in real time. We were able to achieve validation accuracy of 1.00 and testing accuracy of 100%. Our model showed good accuracy while predicting results both offline and online

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED

CHAPTER-12 FUTURE SCOPE

1. We can develop a model for ASL word and sentence level recognition. This will require a system that can detect changes with respect to the temporal space.
2. We can develop a complete product that will help the speech and hearing impaired people, and thereby reduce the communication gap.

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI

FORSPECIALLY ABLED

CHAPTER-13

APPENDIX

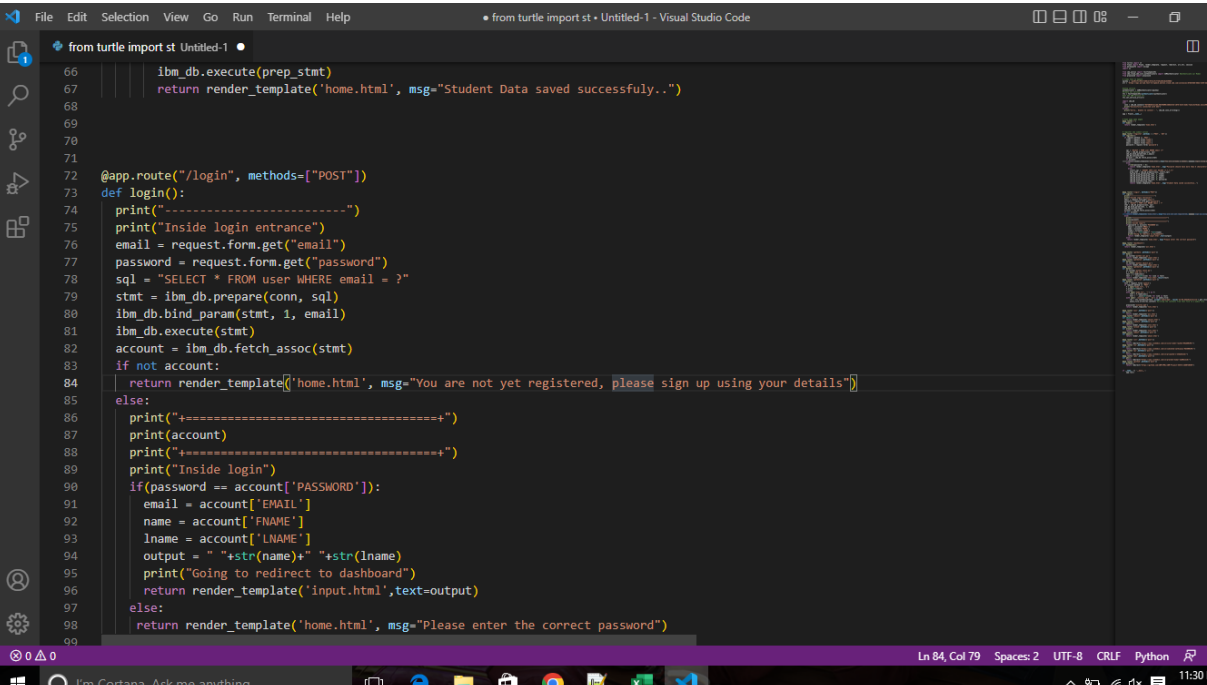
13.1 SOURCE CODE

shlens.py

```
from turtle import st
1 from flask import Flask, render_template, request, redirect, url_for, session
2 from markupsafe import escape
3 import os
4
5
6 from ibm_watson import TextToSpeechV1
7 from ibm_cloud_sdk_core.authenticators import IAMAuthenticator #Authenticate our Model
8 from playsound import playsound
9
10 # Creds Text to Speech
11 apikey = 'qoaM1Vfccc9Vj7lBkKsZr97SnsPcYfb9uK41BuEh980Z'
12 url = 'https://api.au-syd.text-to-speech.watson.cloud.ibm.com/instances/8f6bfb66-68b2-4194-bb43-f8567c8b81d3'
13
14 #setup service
15 authenticator = IAMAuthenticator(apikey)
16 #Create our service
17 tts = TextToSpeechV1(authenticator=authenticator)
18 #set the IBM service url
19 tts.set_service_url(url)
20
21 import ibm_db
22 try:
23     conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=3883e7e4-18f5-4afe-be8c-fa31c41761d2.bs2io90l08kqb1od8l9c.databases.appdomain.cloud;PORT=31498;S
24     print("Successfully connected with db2")
25 except:
26     print("Sorry.. Unable to connect : ", ibm_db.conn_errormsg())
27
28 app = Flask(__name__)
29
30 # Home page open aagum
31 @app.route('/')
32 def home():
```

```
33 @app.route('/')
34 def home():
35     return render_template('home.html')
36
37
38 # register oda submit action
39 @app.route('/register', methods = ['POST', 'GET'])
40 def register():
41     if request.method == 'POST':
42         fname = request.form['fname']
43         lname = request.form['lname']
44         email = request.form['email']
45         password = request.form['password']
46
47
48
49 sql = "SELECT * FROM user WHERE email =?"
50 stmt = ibm_db.prepare(conn, sql)
51 ibm_db.bind_param(stmt, 1, email)
52 ibm_db.execute(stmt)
53 account = ibm_db.fetch_assoc(stmt)
54 if account:
55     return render_template('home.html', msg="You are already a member, please login using your details")
56 else:
57     if(len(password) < 6):
58         return render_template('home.html', msg="Password should have more than 6 characters!!")
59     else:
60         insert_sql = "INSERT INTO user VALUES (?, ?, ?, ?)"
61         prep_stmt = ibm_db.prepare(conn, insert_sql)
62         ibm_db.bind_param(prepare_stmt, 1, fname)
63         ibm_db.bind_param(prepare_stmt, 2, lname)
64         ibm_db.bind_param(prepare_stmt, 3, email)
65         ibm_db.bind_param(prepare_stmt, 4, password)
66         ibm_db.execute(prepare_stmt)
```

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED



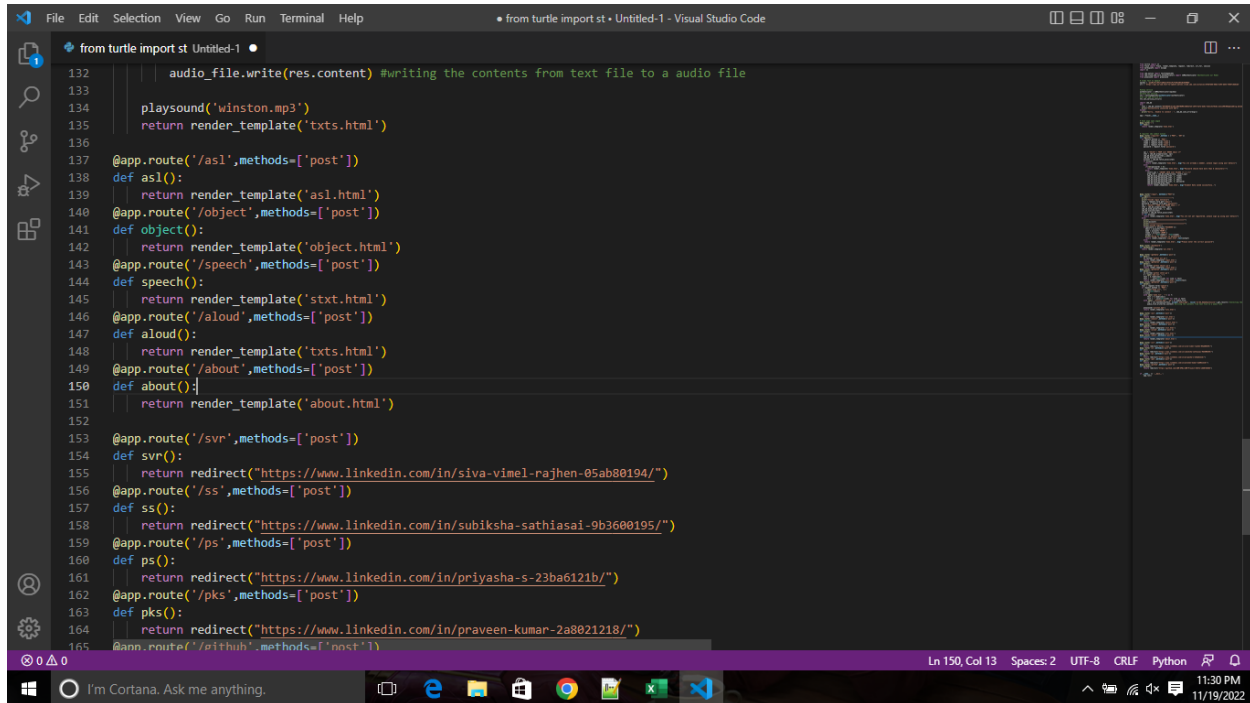
```
from turtle import st
Untitled-1
66 |         ibm_db.execute(prepare_stmt)
67 |         return render_template('home.html', msg="Student Data saved successfully..")
68 |
69 |
70 |
71 |
72 | @app.route("/login", methods=["POST"])
73 | def login():
74 |     print("-----")
75 |     print("Inside login entrance")
76 |     email = request.form.get("email")
77 |     password = request.form.get("password")
78 |     sql = "SELECT * FROM user WHERE email = ?"
79 |     stmt = ibm_db.prepare(conn, sql)
80 |     ibm_db.bind_param(stmt, 1, email)
81 |     ibm_db.execute(stmt)
82 |     account = ibm_db.fetch_assoc(stmt)
83 |     if not account:
84 |         return render_template('home.html', msg="You are not yet registered, please sign up using your details")
85 |     else:
86 |         print("+++++")
87 |         print(account)
88 |         print("+++++")
89 |         print("Inside login")
90 |         if (password == account['PASSWORD']):
91 |             email = account['EMAIL']
92 |             name = account['FNAME']
93 |             lname = account['LNAME']
94 |             output = " " + str(name) + " " + str(lname)
95 |             print("Going to redirect to dashboard")
96 |             return render_template('input.html', text=output)
97 |         else:
98 |             return render_template('home.html', msg="Please enter the correct password")
99 |
```

```

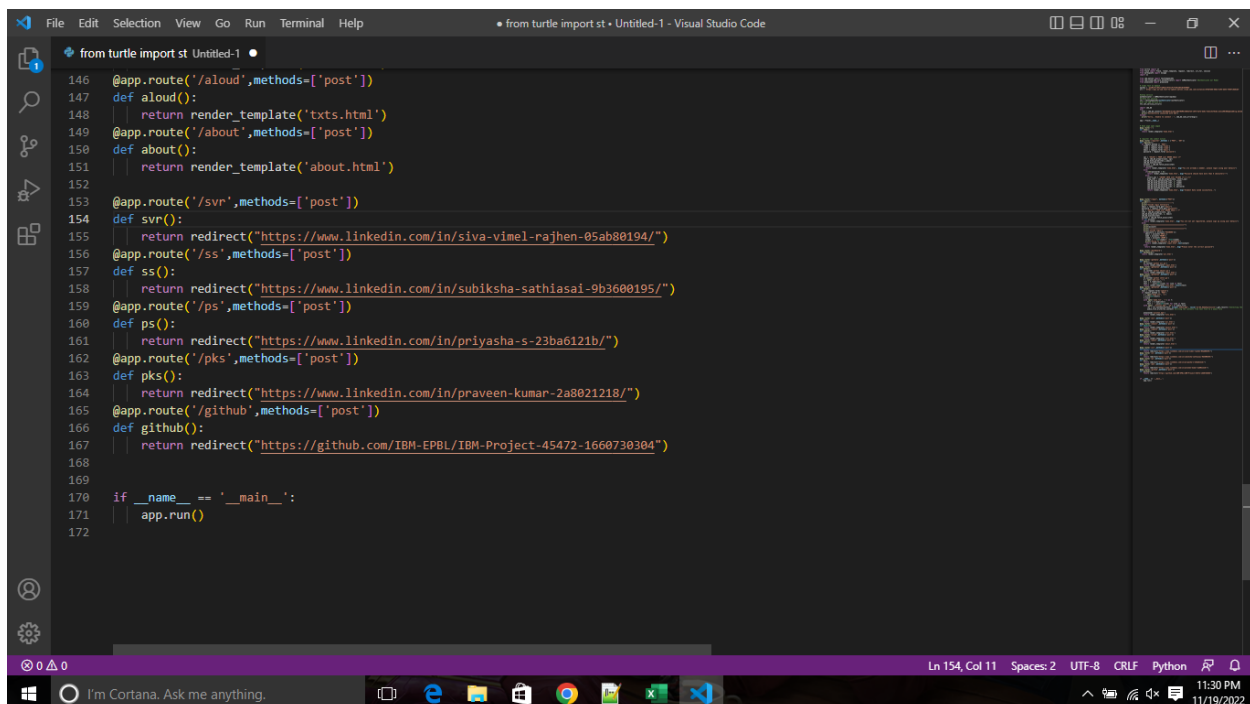
99
100 @app.route('/dashboard')
101 def dashboard():
102     return render_template("asl.html")
103
104
105 @app.route('/getdata', methods=["post"])
106 def data():
107     os.system('python asl.py')
108     return render_template("input.html")
109 @app.route('/getdata2', methods=["post"])
110 def data2():
111     os.system('python object.py')
112     return render_template("input.html")
113 @app.route('/getdata3', methods=["post"])
114 def data3():
115     os.system('python s2txt.py')
116     h = open('mod.txt', 'r')
117     text = h.readlines()
118     text = ''.join(str(line) for line in text)
119     return render_template("stxt.html", result=text)
120 @app.route('/getdata4', methods=["post"])
121 def data4():
122     text = request.form['speech']
123     if request.method == 'POST':
124         f = open('mode.txt', 'w')
125         f.write(str(text))
126         f.close()
127         with open('mode.txt', 'r') as f:
128             text = f.readlines()
129             text = ''.join(str(line) for line in text)
130         with open('./winston.mp3', 'wb') as audio_file:
131             res = tts.synthesize(text, accept='audio/mp3', voice='en-US_EmmaExpressive').get_result() #selecting the audio format and voice
132             audio_file.write(res.content) #writing the contents from text file to a audio file

```

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED



```
from turtle import st
132 | audio_file.write(res.content) #writing the contents from text file to a audio file
133 |
134 | playsound('winston.mp3')
135 | return render_template('txts.html')
136 |
137 | @app.route('/asl', methods=['post'])
138 | def asl():
139 |     return render_template('asl.html')
140 | @app.route('/object', methods=['post'])
141 | def object():
142 |     return render_template('object.html')
143 | @app.route('/speech', methods=['post'])
144 | def speech():
145 |     return render_template('stxt.html')
146 | @app.route('/aloud', methods=['post'])
147 | def aloud():
148 |     return render_template('txts.html')
149 | @app.route('/about', methods=['post'])
150 | def about():
151 |     return render_template('about.html')
152 |
153 | @app.route('/svr', methods=['post'])
154 | def svr():
155 |     return redirect("https://www.linkedin.com/in/siva-vimel-rajhen-05ab80194/")
156 | @app.route('/ss', methods=['post'])
157 | def ss():
158 |     return redirect("https://www.linkedin.com/in/subiksha-sathiasai-9b3600195/")
159 | @app.route('/ps', methods=['post'])
160 | def ps():
161 |     return redirect("https://www.linkedin.com/in/priyasha-s-23ba6121b/")
162 | @app.route('/pks', methods=['post'])
163 | def pks():
164 |     return redirect("https://www.linkedin.com/in/praveen-kumar-2a8021218/")
165 | @app.route('/github', methods=['post'])
```



```
146 | @app.route('/aloud', methods=['post'])
147 | def aloud():
148 |     return render_template('txts.html')
149 | @app.route('/about', methods=['post'])
150 | def about():
151 |     return render_template('about.html')
152 |
153 | @app.route('/svr', methods=['post'])
154 | def svr():
155 |     return redirect("https://www.linkedin.com/in/siva-vimel-rajhen-05ab80194/")
156 | @app.route('/ss', methods=['post'])
157 | def ss():
158 |     return redirect("https://www.linkedin.com/in/subiksha-sathiasai-9b3600195/")
159 | @app.route('/ps', methods=['post'])
160 | def ps():
161 |     return redirect("https://www.linkedin.com/in/priyasha-s-23ba6121b/")
162 | @app.route('/pks', methods=['post'])
163 | def pks():
164 |     return redirect("https://www.linkedin.com/in/praveen-kumar-2a8021218/")
165 | @app.route('/github', methods=['post'])
166 | def github():
167 |     return redirect("https://github.com/IBM-EPBL/IBM-Project-45472-1660738304")
168 |
169 |
170 | if __name__ == '__main__':
171 |     app.run()
172 |
```

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

als.py

```
File Edit Selection View Go Run Terminal Help • import numpy as np • Untitled-1 - Visual Studio Code
import numpy as np
import cv2
import keras
from keras.preprocessing.image import ImageDataGenerator
import tensorflow as tf

model = keras.models.load_model("best_model_dataflair3.h5")
word_dict = {0:'A',1:'B',2:'C',3:'D',4:'E',5:'F',6:'G',7:'H',8:'I',9:'J',10:'K',11:'L',12:'M',13:'N',14:'O',15:'P',17:'Q',18:'R',19:'S',20:'T',21:'U',22:'V',23:'W',24:'X',25:'Y',26:'Z'}
background = None
accumulated_weight = 0.5
ROI_top = 100
ROI_bottom = 300
ROI_right = 150
ROI_left = 350

def cal_accum_avg(frame, accumulated_weight):
    global background
    if background is None:
        background = frame.copy().astype("float")
        return None
    cv2.accumulateWeighted(frame, background, accumulated_weight)

def segment_hand(frame, threshold=25):
    global background
    diff = cv2.absdiff(background.astype("uint8"), frame)
    _, thresholded = cv2.threshold(diff, threshold, 255, cv2.THRESH_BINARY)
    image = cv2.findContours(thresholded.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
    contours, hierarchy = cv2.findContours(thresholded.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

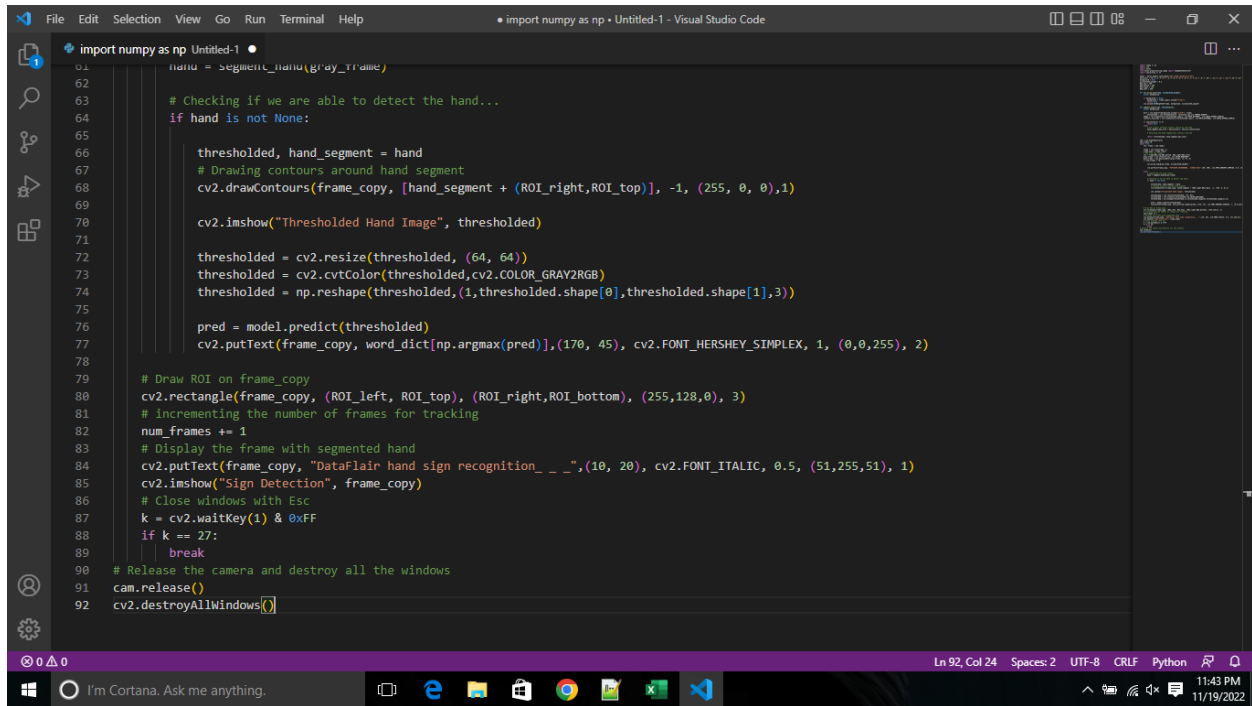
    if len(contours) == 0:
        return None
    else:
        # The largest external contour should be the hand
        hand_segment_max_cont = max(contours, key=cv2.contourArea)
        # Returning the hand segment(max contour) and the
        return (thresholded, hand_segment_max_cont)

cam = cv2.VideoCapture(0)
num_frames = 0
while True:
    ret, frame = cam.read()
    frame = cv2.flip(frame, 1)
    frame_copy = frame.copy()
    # ROI from the frame
    roi = frame[ROI_top:ROI_bottom, ROI_right:ROI_left]
    gray_frame = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
    gray_frame = cv2.GaussianBlur(gray_frame, (9, 9), 0)
    if num_frames < 70:
        cal_accum_avg(gray_frame, accumulated_weight)
        cv2.putText(frame_copy, "FETCHING BACKGROUND...PLEASE WAIT", (80, 400), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 0, 255), 2)
    else:
        # segmenting the hand region
        hand = segment_hand(gray_frame)
        # Checking if we are able to detect the hand...
        if hand is not None:
            thresholded, hand_segment = hand
```

```
File Edit Selection View Go Run Terminal Help • import numpy as np • Untitled-1 - Visual Studio Code
else:
    # The largest external contour should be the hand
    hand_segment_max_cont = max(contours, key=cv2.contourArea)
    # Returning the hand segment(max contour) and the
    return (thresholded, hand_segment_max_cont)

cam = cv2.VideoCapture(0)
num_frames = 0
while True:
    ret, frame = cam.read()
    frame = cv2.flip(frame, 1)
    frame_copy = frame.copy()
    # ROI from the frame
    roi = frame[ROI_top:ROI_bottom, ROI_right:ROI_left]
    gray_frame = cv2.cvtColor(roi, cv2.COLOR_BGR2GRAY)
    gray_frame = cv2.GaussianBlur(gray_frame, (9, 9), 0)
    if num_frames < 70:
        cal_accum_avg(gray_frame, accumulated_weight)
        cv2.putText(frame_copy, "FETCHING BACKGROUND...PLEASE WAIT", (80, 400), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (0, 0, 255), 2)
    else:
        # segmenting the hand region
        hand = segment_hand(gray_frame)
        # Checking if we are able to detect the hand...
        if hand is not None:
            thresholded, hand_segment = hand
```

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED



```
import numpy as np
...
hand = segment_hand(gray_frame)

# Checking if we are able to detect the hand...
if hand is not None:

    thresholded, hand_segment = hand
    # Drawing contours around hand segment
    cv2.drawContours(frame_copy, [hand_segment + (ROI_right, ROI_top)], -1, (255, 0, 0), 1)

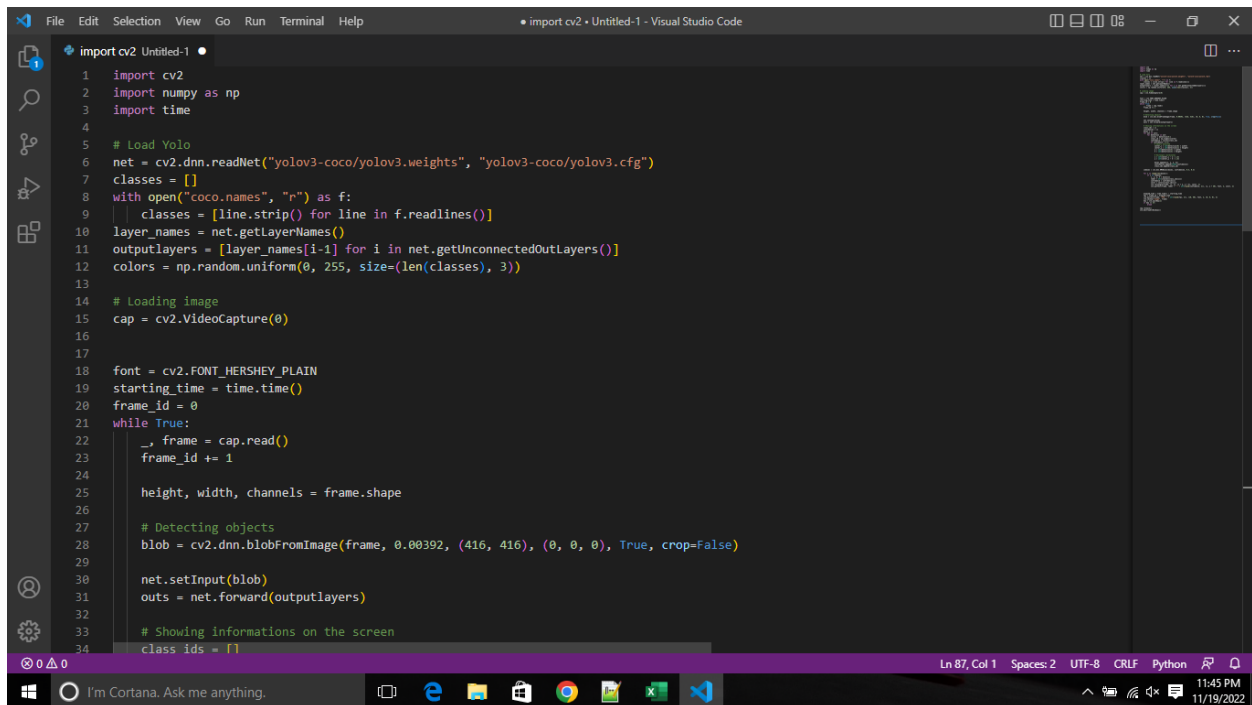
    cv2.imshow("Thresholded Hand Image", thresholded)

    thresholded = cv2.resize(thresholded, (64, 64))
    thresholded = cv2.cvtColor(thresholded, cv2.COLOR_GRAY2RGB)
    thresholded = np.reshape(thresholded, (1, thresholded.shape[0], thresholded.shape[1], 3))

    pred = model.predict(thresholded)
    cv2.putText(frame_copy, word_dict[np.argmax(pred)], (170, 45), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255), 2)

# Draw ROI on frame_copy
cv2.rectangle(frame_copy, (ROI_left, ROI_top), (ROI_right, ROI_bottom), (255, 128, 0), 3)
# Incrementing the number of frames for tracking
num_frames += 1
# Display the frame with segmented hand
cv2.putText(frame_copy, "Datafair hand sign recognition _ _ _", (10, 20), cv2.FONT_ITALIC, 0.5, (51, 255, 51), 1)
cv2.imshow("Sign Detection", frame_copy)
# Close windows with Esc
k = cv2.waitKey(1) & 0xFF
if k == 27:
    break
# Release the camera and destroy all the windows
cam.release()
cv2.destroyAllWindows()
```

object.py



```
import cv2
import numpy as np
import time

# Load Yolo
net = cv2.dnn.readNet("yolov3-coco/yolov3.weights", "yolov3-coco/yolov3.cfg")
classes = []
with open("coco.names", "r") as f:
    classes = [line.strip() for line in f.readlines()]
layer_names = net.getLayerNames()
outputlayers = [layer_names[i-1] for i in net.getUnconnectedOutLayers()]
colors = np.random.uniform(0, 255, size=(len(classes), 3))

# Loading image
cap = cv2.VideoCapture(0)

font = cv2.FONT_HERSHEY_PLAIN
starting_time = time.time()
frame_id = 0
while True:
    frame = cap.read()
    frame_id += 1

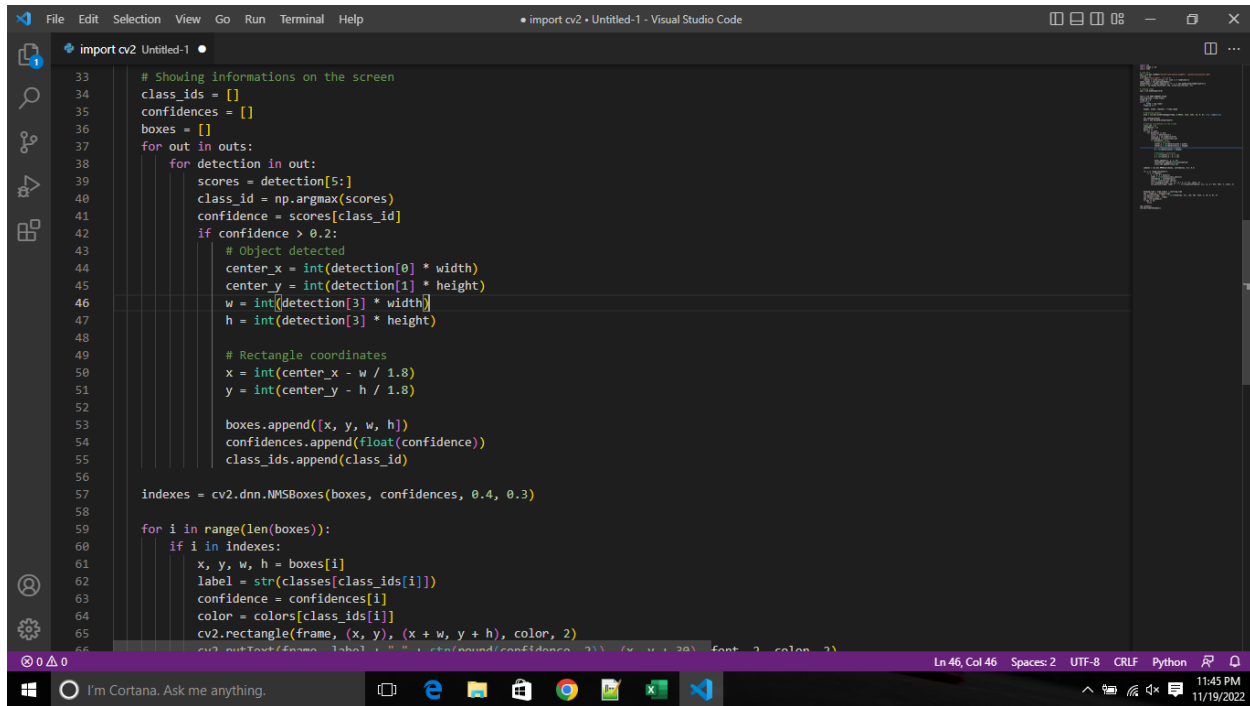
    height, width, channels = frame.shape

    # Detecting objects
    blob = cv2.dnn.blobFromImage(frame, 0.00392, (416, 416), (0, 0, 0), True, crop=False)

    net.setInput(blob)
    outs = net.forward(outputlayers)

    # Showing informations on the screen
    class_ids = []
```

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED



```
import cv2
import numpy as np
import colornames

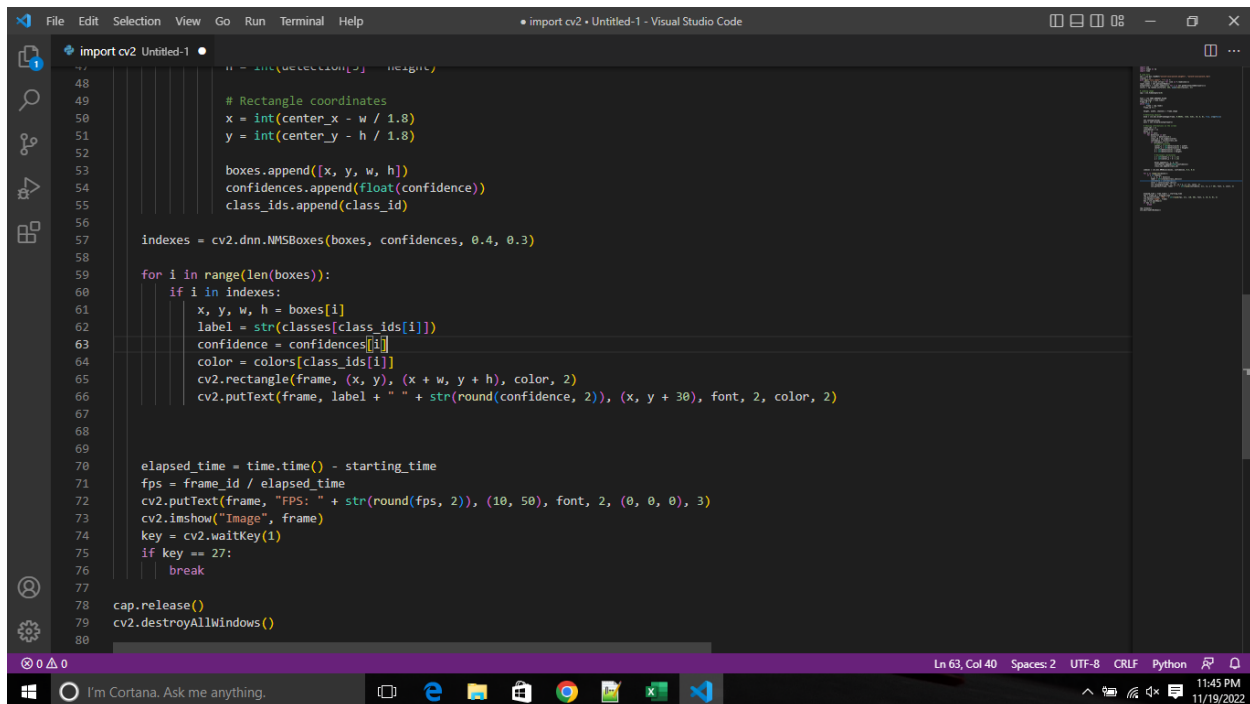
# Showing informations on the screen
class_ids = []
confidences = []
boxes = []
for out in outs:
    for detection in out:
        scores = detection[5:]
        class_id = np.argmax(scores)
        confidence = scores[class_id]
        if confidence > 0.2:
            # Object detected
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * height)
            w = int(detection[3] * width)
            h = int(detection[3] * height)

            # Rectangle coordinates
            x = int(center_x - w / 1.8)
            y = int(center_y - h / 1.8)

            boxes.append([x, y, w, h])
            confidences.append(float(confidence))
            class_ids.append(class_id)

indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.4, 0.3)

for i in range(len(boxes)):
    if i in indexes:
        x, y, w, h = boxes[i]
        label = str(classes[class_ids[i]])
        confidence = confidences[i]
        color = colors[class_ids[i]]
        cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
        cv2.putText(frame, label + " " + str(round(confidence, 2)), (x + w + 30, y + h + 30), font, 2, color, 2)
```



```
indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.4, 0.3)

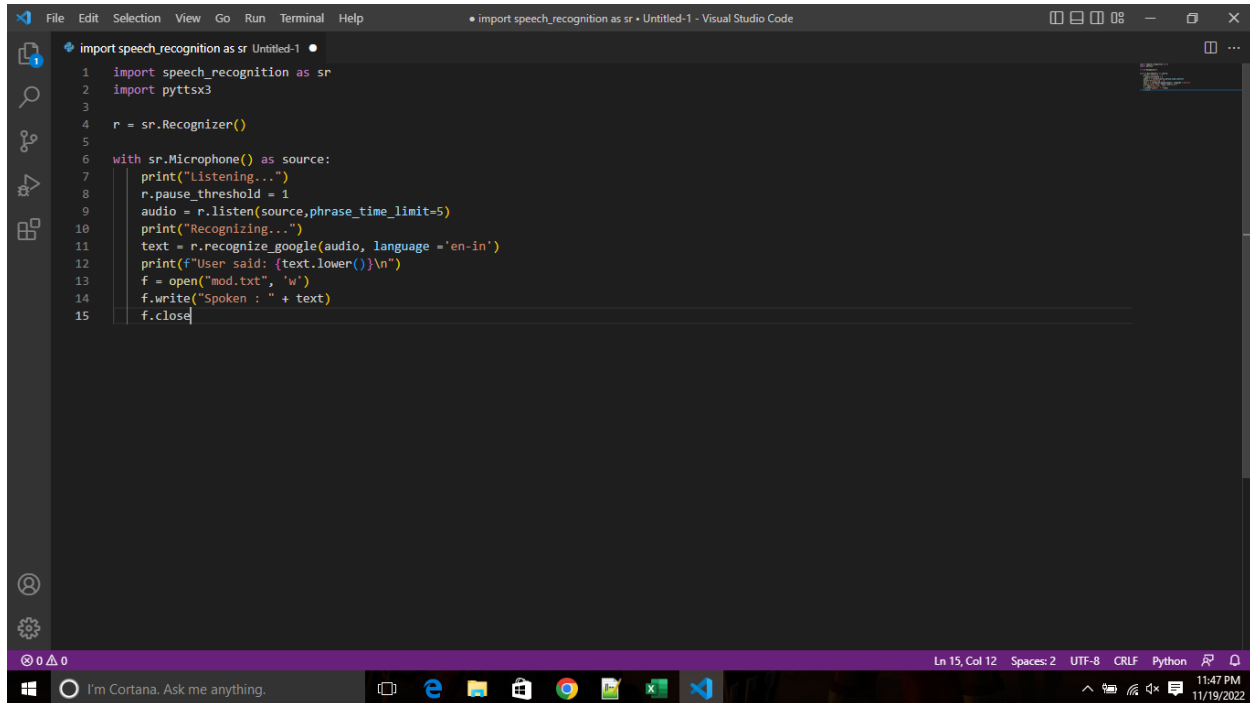
for i in range(len(boxes)):
    if i in indexes:
        x, y, w, h = boxes[i]
        label = str(classes[class_ids[i]])
        confidence = confidences[i]
        color = colors[class_ids[i]]
        cv2.rectangle(frame, (x, y), (x + w, y + h), color, 2)
        cv2.putText(frame, label + " " + str(round(confidence, 2)), (x + w + 30, y + h + 30), font, 2, color, 2)

elapsed_time = time.time() - starting_time
fps = frame_id / elapsed_time
cv2.putText(frame, "FPS: " + str(round(fps, 2)), (10, 50), font, 2, (0, 0, 0), 3)
cv2.imshow("Image", frame)
key = cv2.waitKey(1)
if key == 27:
    break

cap.release()
cv2.destroyAllWindows()
```

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

s2txt.py

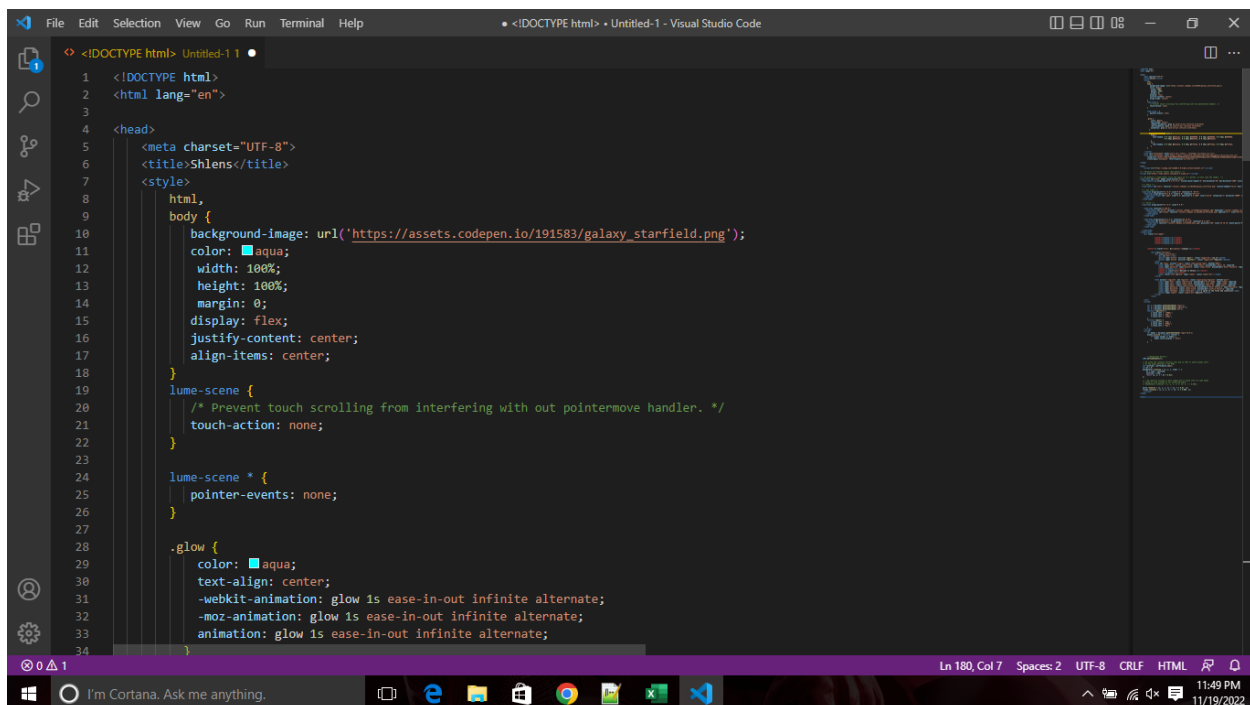


```
import speech_recognition as sr
import pyttsx3

r = sr.Recognizer()

with sr.Microphone() as source:
    print("Listening...")
    r.pause_threshold = 1
    audio = r.listen(source, phrase_time_limit=5)
    print("Recognizing...")
    text = r.recognize_google(audio, language='en-in')
    print(f"User said: {text.lower()}")
    f = open("mod.txt", 'w')
    f.write("Spoken : " + text)
    f.close()
```

home.html

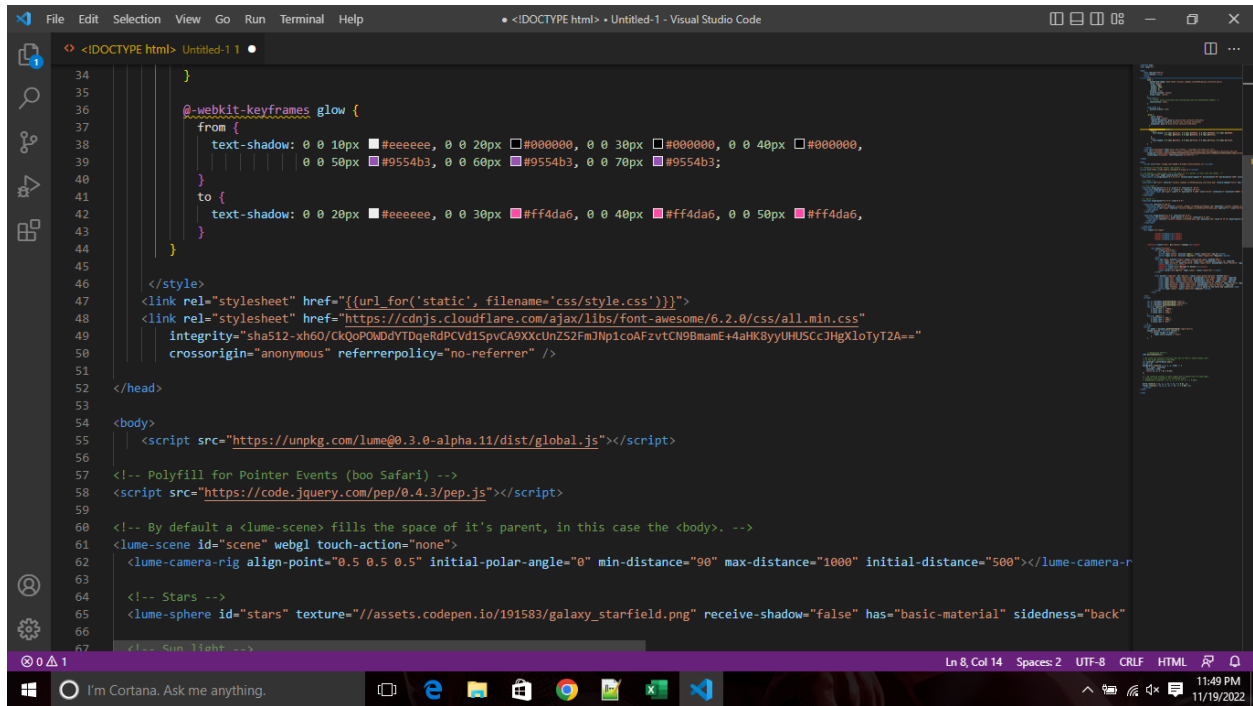


```
<!DOCTYPE html>
<html lang="en">

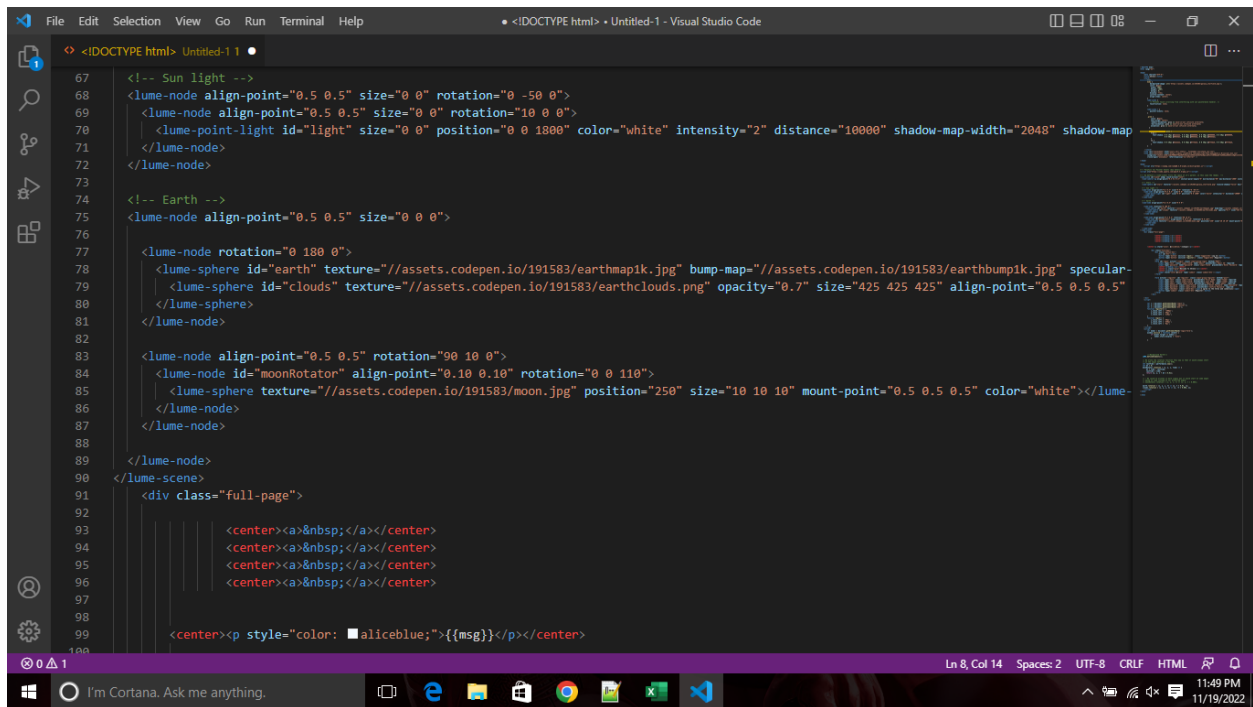
<head>
  <meta charset="UTF-8">
  <title>Shlens</title>
  <style>
    html,
    body {
      background-image: url('https://assets.codepen.io/191583/galaxy_starfield.png');
      color: aqua;
      width: 100%;
      height: 100%;
      margin: 0;
      display: flex;
      justify-content: center;
      align-items: center;
    }
    lume-scene {
      /* Prevent touch scrolling from interfering with out pointermove handler. */
      touch-action: none;
    }
    lume-scene * {
      pointer-events: none;
    }
    .glow {
      color: aqua;
      text-align: center;
      -webkit-animation: glow 1s ease-in-out infinite alternate;
      -moz-animation: glow 1s ease-in-out infinite alternate;
      animation: glow 1s ease-in-out infinite alternate;
    }
  </style>
</head>

<body>
  <div>
    <h1>Shlens</h1>
  </div>
</body>
</html>
```

REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED



```
34 }
35
36 @-webkit-keyframes glow {
37   from {
38     text-shadow: 0 0 10px #ffffff, 0 0 20px #000000, 0 0 30px #000000, 0 0 40px #000000,
39                 0 0 50px #9554b3, 0 0 60px #9554b3, 0 0 70px #9554b3;
40   }
41   to {
42     text-shadow: 0 0 20px #ffffff, 0 0 30px #ff4da6, 0 0 40px #ff4da6, 0 0 50px #ff4da6,
43                 0 0 60px #ff4da6, 0 0 70px #ff4da6, 0 0 80px #ff4da6;
44   }
45 }
46
47 </style>
48 <link rel="stylesheet" href="{{url_for('static', filename='css/style.css')}}">
49 <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.2.0/css/all.min.css"
50       integrity="sha512-xh60/CkQoPOWdYTDqRdPCVd1SpvCA9XXcUnZS2FmJNp1coAFzvtCN9BmamE+4aHK8yyUHUSCcJHgXloT2A=="
51       crossorigin="anonymous" referrerpolicy="no-referrer" />
52 </head>
53
54 <body>
55   <script src="https://unpkg.com/lume@0.3.0-alpha.11/dist/global.js"></script>
56
57   <!-- Polyfill for Pointer Events (boo Safari) -->
58   <script src="https://code.jquery.com/jquery/3.4.3/jquery.js"></script>
59
60   <!-- By default a <lume-scene> fills the space of it's parent, in this case the <body>. -->
61   <lume-scene id="scene" webgl touch-action="none">
62     <lume-camera-rig align-point="0.5 0.5 0.5" initial-polar-angle="0" min-distance="90" max-distance="1000" initial-distance="500"></lume-camera-rig>
63
64     <!-- Stars -->
65     <lume-sphere id="stars" texture="//assets.codepen.io/191583/galaxy_starfield.png" receive-shadow="false" has="basic-material" sidedness="back"
66               ></lume-sphere>
67     <!-- Sun light -->
```



```
67 <!-- Sun light -->
68 <lume-node align-point="0.5 0.5" size="0 0" rotation="0 -50 0">
69   <lume-node align-point="0.5 0.5" size="0 0" rotation="10 0 0">
70     <lume-point-light id="light" size="0 0" position="0 0 1000" color="white" intensity="2" distance="10000" shadow-map-width="2048" shadow-map-size="1024 1024"></lume-point-light>
71   </lume-node>
72 </lume-node>
73
74 <!-- Earth -->
75 <lume-node align-point="0.5 0.5" size="0 0">
76
77   <lume-node rotation="0 180 0">
78     <lume-sphere id="earth" texture="//assets.codepen.io/191583/earthmap1k.jpg" bump-map="//assets.codepen.io/191583/earthbump1k.jpg" specular-roughness="0.5" opacity="1"></lume-sphere>
79     <lume-sphere id="clouds" texture="//assets.codepen.io/191583/earthclouds.png" opacity="0.7" size="425 425 425" align-point="0.5 0.5 0.5"></lume-sphere>
80   </lume-node>
81
82   <lume-node align-point="0.5 0.5" rotation="90 10 0">
83     <lume-node id="moonRotator" align-point="0.10 0.10" rotation="0 0 110">
84       <lume-sphere texture="//assets.codepen.io/191583/moon.jpg" position="250" size="10 10 10" mount-point="0.5 0.5 0.5" color="white"></lume-sphere>
85     </lume-node>
86   </lume-node>
87 </lume-node>
88
89 </lume-scene>
90 <div class="full-page">
91
92   <center><a>&nbsp;</a></center>
93   <center><a>&nbsp;</a></center>
94   <center><a>&nbsp;</a></center>
95   <center><a>&nbsp;</a></center>
96
97   <center><p style="color: #aliceblue;">{{msg}}</p></center>
98
99 </div>
```


REAL-TIME COMMUNICATION SYSTEM POWERED BY AI FOR SPECIALLY ABLED

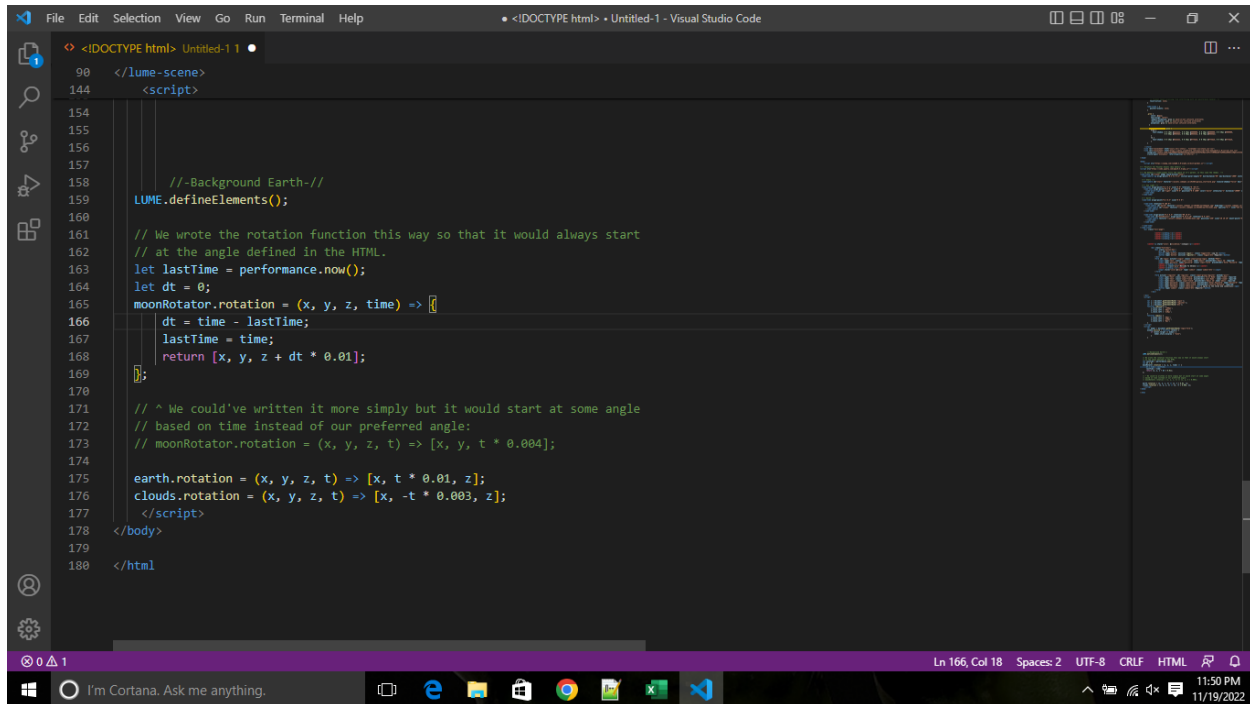
```
File Edit Selection View Go Run Terminal Help • <!DOCTYPE html> • Untitled-1 - Visual Studio Code

100
101
102 <div class="form-box">
103   <div class="button-box">
104     <div id="btn"></div>
105     <button type="button" onclick="login()" class="toggle-btn">Log In</button>
106     <button type="button" onclick="register()" class="toggle-btn">Register</button>
107   </div>
108   <form id="login" action="/login" class="input-group-login" method="POST">
109     <input type="text" class="input-field" name="email" placeholder="Email Id" required>
110     <input type="password" name="password" class="input-field" placeholder="Enter Password" required>
111     <center><a>&nbsp;</a></center>
112     <center><a class="glow">Welcome To Shlens</a></center>
113     <center><a>&nbsp;</a></center>
114     <input style="color: black" type="submit" class="submit-btn"></input>
115   </form>
116
117   <form action="/register" id="register" class="input-group-register" method="post">
118     <input type="text" class="input-field" placeholder="First Name" name="fname" required>
119     <input type="text" class="input-field" placeholder="Last Name" name="lname" required>
120     <input type="email" class="input-field" placeholder="Email Id" name="email" required>
121     <input type="password" class="input-field" placeholder="Enter Password" name="password" required>
122     <input type="password" class="input-field" placeholder="Confirm Password" required>
123     <input type="checkbox" class="checkbox" ><span>I agree to the terms and conditions</span>
124     <button type="submit" class="submit-btn">Register</button>
125   </form>
126 </div>
127
128 <script>
129
130   var x = document.getElementById('login');
131   var y = document.getElementById('register');
132   var z = document.getElementById('btn');
133   function register() {
```

```
File Edit Selection View Go Run Terminal Help • <!DOCTYPE html> • Untitled-1 - Visual Studio Code

90 </lume-scene>
128 <script>
133   function register() {
134     x.style.left = '400px';
135     y.style.left = '50px';
136     z.style.left = '110px';
137   }
138   function login() {
139     x.style.left = '50px';
140     y.style.left = '450px';
141     z.style.left = '0px';
142   }
143 </script>
144 <script>
145   var modal = document.getElementById('login-form');
146   window.onclick = function (event) {
147     if (event.target == modal) {
148       modal.style.display = "none";
149     }
150   }
151
152
153
154
155
156
157
158 //Background Earth//
159 LUME.defineElements();
160
161 // We wrote the rotation function this way so that it would always start
162 // at the angle defined in the HTML.
163 let lastTime = performance.now();
164 let dt = 0;
```

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED



The screenshot shows a Visual Studio Code editor window with a file named 'Untitled-1.html'. The code is written in HTML and JavaScript. It includes a LUMEScene and a moonRotator. The code is as follows:

```
90 </lume-scene>
144 <script>
154
155
156
157
158 //Background Earth-//
159 LUME.defineElements();
160
161 // We wrote the rotation function this way so that it would always start
162 // at the angle defined in the HTML.
163 let lastTime = performance.now();
164 let dt = 0;
165 moonRotator.rotation = (x, y, z, time) => {
166   dt = time - lastTime;
167   lastTime = time;
168   return [x, y, z + dt * 0.01];
169 };
170
171 // ^ We could've written it more simply but it would start at some angle
172 // based on time instead of our preferred angle:
173 // moonRotator.rotation = (x, y, z, t) => [x, y, t * 0.004];
174
175 earth.rotation = (x, y, z, t) => [x, t * 0.01, z];
176 clouds.rotation = (x, y, z, t) => [x, -t * 0.003, z];
177 </script>
178 </body>
179
180 </html>
```

13.2 GITHUB AND PROJECT DEMO LINK

Github link: <https://github.com/IBM-EPBL/IBM-Project-45472-1660730304>

Project demo link:

<https://drive.google.com/file/d/1S4gESFaB01Texde3RPhcmoDyJ2uP24b-/view?usp=sharing>

REAL-TIME COMMUNICATIONSYSTEM POWERED BY AI FORSPECIALLY ABLED