Project ID	Student Name(s)	Registration Number	Section (A/B)	Instructor Remarks
	Mayank V Chopra	229309229	C	
	Kunal Yadav	229309237	C	

Finalized Domain Name:
Software

Finalized Problem Statement:
Real-Time Sign Language Translation System
To develop a real-time system that converts speech or text into sign language gestures, ensuring seamless communication between non-sign language users and individuals who rely on sign language. The system will integrate deep learning techniques for speech recognition, natural language processing (NLP), and 3D gesture visualization.

Finalized Dataset:
a) Dataset Size:
Size: 290GB
Frames: Around 13.5 million frames extracted
Data Split: ~80% for training, ~10% for validation, ~10% for testing

b) Dataset Link:
https://how2sign.github.io/

c) Dataset features:
Video, Depth Information, Hand Landmarks, Translation, Pose Data, Audio

Finalized block diagram/model architecture:

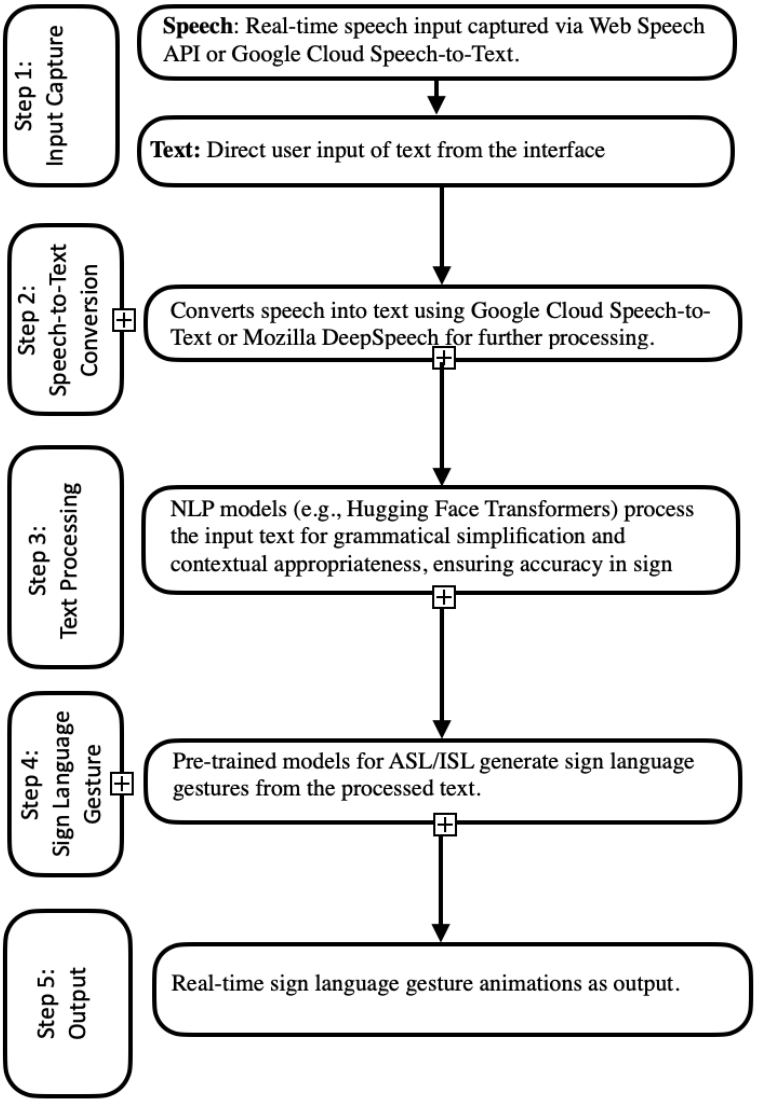

Fig.2 Flow Chart of Text/Speech to Sign Language

1. Platform/ libraries:

- **Speech Recognition:** Google Cloud Speech-to-Text, Mozilla DeepSpeech, Web Speech API
- **NLP:** Hugging Face Transformers, TensorFlow, PyTorch
- **Gesture Rendering:** Three.js, Unity (for 3D avatars)
- **Frontend:** React.js, Flutter
- **Backend:** Python (Flask/Django), WebSockets for real-time communication
- **Cloud:** AWS, Google Cloud
- **Containerization:** Docker, Kubernetes
- **Database:** MongoDB, PostgreSQL (for user preferences, logs)

2. Tools and Techniques: Enlist all the techniques data preprocessing and processing techniques (ML/DL/Optimization/ or any other) that you are using in your experiment.

2.1 Data Pre-processing Techniques

S.No.	Name of the Technique	2-3 line description	Justification for incorporating this technique in your model
1.	Text Normalization	Simplifies text by removing punctuation, lowercasing words, and handling abbreviations for uniform processing.	This ensures that the input text is in a clean, consistent format, minimizing errors in NLP models and improving translation accuracy.
2.	Speech-to-Text Conversion	Converts real-time speech input into text using automatic speech recognition (ASR) models like Google Cloud Speech-to-Text.	This enables the system to handle speech inputs efficiently, translating spoken language into a format suitable for NLP and gesture generation.
3.	Tokenization	Breaks down input text into smaller components (words, sentences) for better processing by NLP models.	Tokenization allows the system to process text more granularly, helping to identify key parts for gesture translation.

2.2 Model Training and Testing Techniques or any other:

S.No.	Name of the Technique	2-3 line description	Justification for incorporating this technique in your model
1.	Sequence-to-Sequence (Seq2Seq) Translation	A Seq2Seq model translates the processed input text into a sequence of sign language gestures. It utilizes an encoder-decoder framework to map sentences to gesture sequences.	Seq2Seq models are highly effective for language translation tasks, ensuring that the gestures correspond to entire sentences, not just individual words.
2.	Long Short-Term Memory (LSTM)	LSTMs are used to capture long-term dependencies in text, ensuring context is	LSTMs are essential for maintaining the contextual flow of text across multiple words, ensuring accurate gesture

		maintained when generating sign language gestures.	translations for complex sentences.
3.	Gesture Synthesis and Rendering	3D avatars are used to generate and render the sign language gestures using **Three.js** or **Unity** for real-time animation.	Gesture synthesis enables the translation of text into sign language in a way that is visually understandable for sign language users.
4.	Automatic Speech Recognition (ASR)	Converts speech input into text, allowing the system to process spoken language in real time.	ASR models provide an additional input modality, enhancing the system's versatility by supporting both speech and text inputs.

3. Experimentation:

Input:

- **Speech:** Real-time speech input using Web Speech API or Google Cloud Speech-to-Text.
- **Text:** Written sentences in English or other languages.

Expected Output:
Accurate generation of sign language gestures (in ASL/ISL) visualized through 3D animated avatars.

Coding Status:

- **Speech-to-Text Conversion**: 50% complete (Google Cloud Speech-to-Text integration in progress).
- **Text Preprocessing and Tokenization**: Ongoing (Basic tokenization complete).
- **Model Development**: Pending (Seq2Seq and LSTM model exploration ongoing).
- **Gesture Rendering**: Planned (Integration with Three.js for 3D avatar rendering).
- **Real-Time System Integration**: Planned.