

Software Engineering Department  
Braude College

Capstone Project Phase A – 61998

**SoundSigns: Translating English Audio to Sign Language**

**25-1-D-19**

**Table of Contents**

[**Abstract 4**](#_1t3h5sf)

[**1. Introduction 4**](#_4d34og8)

[**2. Related Work 6**](#_17dp8vu)

[**3. Background 7**](#_3rdcrjn)

[3.1. American Sign Language (ASL) 7](#_26in1rg)

[3.2. ASL Gloss 7](#_lnxbz9)

[3.3. Speech-to-Text Conversion Technologies 7](#_35nkun2)

[3.4. Machine Learning and Natural Language Processing 7](#_1ksv4uv)

[3.5. Mediapipe 8](#_44sinio)

**3.6. WLASL Dataset 8**

[3.7. ChatGPT 9](#_2jxsxqh)

[3.8. 3D Avatars 9](#_z337ya)

[**4. Expected Achievements 9**](#_3j2qqm3)

[4.1. Outcomes 9](#_1y810tw)

[4.2. Unique Features 10](#_4i7ojhp)

[- 4.2.1. Real-Time Audio-to-Sign Translation 10](#_2xcytpi)

**- 4.2.2. Enhanced Gloss Transformation and Sentence Structuring 10**

[- 4.2.3. Multi-Platform Accessibility 10](#_3whwml4)

[- 4.2.4. User-Centric Interface 10](#_2bn6wsx)

[4.3. Criteria for Success 11](#_qsh70q)

[**5. The Process 11**](#_3as4poj)

[5.1 Transforming English Text to ASL Gloss 11](#_1pxezwc)

[- 5.1.1 Constraints and Challenges - Transforming English Text to ASL Gloss 11](#_49x2ik5)

[- 5.1.2 Use in Project - Transforming English Text to ASL Gloss 12](#_2p2csry)

[5.2 Motion Capture 12](#_147n2zr)

[- 5.2.1 Constraints and Challenges - Motion Capture 14](#_3o7alnk)

**- 5.2.2 Use in Project - Motion Capture 14**

[5.3 Avatars with Motion Capture 15](#_23ckvvd)

[5.3.1 Constraints and Challenges - Avatars with Motion Capture 15](#_ihv636)

[- 5.3.2 Use in Project - Avatars with Motion Capture 16](#_32hioqz)

[5.4 Methodology and Development Process 16](#_1hmsyys)

**6. The Product 17**

[6.1. Architecture Overview 17](#_41mghml)

[- 6.1.1 SoundSigns Application 18](#_2grqrue)

[Components of the SoundSigns System: 18](#_vx1227)

[- 6.1.2 User interface 19](#_3fwokq0)

**6.2. Requirements 19**

[- 6.2.1 Functional 19](#_4f1mdlm)

[- 6.2.2 Non-Functional 20](#_2u6wntf)

[6.3. Diagrams 22](#_19c6y18)

[- 6.3.1 Use Case 22](#_3tbugp1)

**- 6.3.2 Activity Diagram link 23**

[- 6.3.3 Sequence Diagram link 24](#_28h4qwu)

[**7. Verification and Evaluation 24**](#_37m2jsg)

[7.1. Evaluation 24](#_1mrcu09)

**7.2. Verification 25**

[- 7.2.1 Testing 25](#_46r0co2)

**8. References 27**

# 

# 

# 

# 

# 

# 

# 

# Abstract

In today's world, significant progress has been made to aid the deaf community in communicating with non-deaf individuals. However, there remains a gap in accessibility to media content that typically requires professional sign language interpreters. To address this, we propose a digital tool designed to replace live interpreters using American Sign Language (ASL). Our web-based solution captures English speech via a microphone, processes it through the Whisper model for speech-to-text conversion, and utilizes ChatGPT to transform the text into ASL gloss. This text is then mapped onto a 3D animated avatar using a database of JSON files containing gesture movements. The avatar performs the necessary signs, providing a real-time interpretation of the audio input. This approach leverages MediaPipe for efficient motion capture, ensuring accurate and natural avatar animations. By focusing on adaptability and accessibility, this tool aims to deliver real-time sign language translations, enhancing media content accessibility for the deaf community. Our solution addresses the shortage of interpreters while serving as an educational resource for learning ASL. Through iterative development and user feedback, we aim to refine our tool's functionality and effectiveness, ultimately fostering greater inclusivity and connectivity.

# 1. Introduction

In the modern age human society has made crucial advancements to enhance deaf peoples' ability to communicate with non-deaf people, such as creating hearing aids, and the invention of sign language.

For deaf individuals who cannot use hearing aids, consuming media can be challenging, the primary methods for accessing audio content are through subtitles or professional sign language interpreters. Subtitles are typically provided in regular TV shows and movies, while interpreters are often expected for global news broadcasts. However, these solutions are not always readily available.

For typical social media content, creating subtitles is a time-consuming and challenging task. Hiring a sign language interpreter is also costly and not feasible for many content creators worldwide. We propose a solution to this problem by developing a tool capable of generating sign language like that of a professional interpreter. Our tool will initially support only American English.

The tool interface will be in a website with a microphone button. When clicked, it will record English vocal sounds. The audio recordings will be converted into English text that will be sent to the backend server for processing. The backend server will use the ChatGPT API for transforming the text into English gloss. The resulting terms will be translated into movements using a database containing thousands of JSON files. each representing captured movements of sign language words. These JSON files will be mapped onto a 3D animated avatar, which will display the sign language phrase to the user.

This tool is a step toward addressing the shortage of professional sign language interpreters, providing the deaf community with a primary means for consuming online media and content. Additionally, this website can serve as a platform to teach sign language, helping users learn how to perform signs.

Our project leverages existing tools, including one for converting English speech to text, an animated 3D model for visual representation, and a dataset containing videos of professional sign language interpreters using American Sign Language (ASL).

The syntax of ASL differs significantly from that of English. Therefore, we will use ChatGPT to convert the English text syntax into ASL gloss format.

At the time of writing this project, a few existing sign language tools align with the vision of our website. Unfortunately, these tools often require manual setup, such as downloading datasets and libraries, or involve using complex, non-user-friendly applications. Additionally, many of them lack a high-quality animated avatar.

The core feature of the project is real-time sign language interpretation. This functionality ensures that as audio is recorded, the avatar animates in sync with it, effectively mimicking the actions of a human sign language interpreter.

# 

# 2. Related Work

Recent technological advancements have greatly aided the development of systems that translate spoken English into American Sign Language (ASL), enhancing accessibility for the deaf and hard-of-hearing communities. The SoundSigns project is inspired by several existing projects and research contributions.

Prior research has extensively explored the usage of 3D avatars for sign language interpretation. Sanaullah et al. (2022)[4] discuss the implementation of neural networks to improve real-time translation of text into sign language, highlighting the effectiveness of automated systems in enhancing accessibility for the deaf.

The systematic review by Kahlon and Singh (2023)[5] provides an overview of different systems and approaches used in textual-to-sign-language translation, emphasizing the challenges and solutions in developing effective translation models.

Sáfár and Marshall (2001)[6] delve into the design of systems that convert English text into sign language, focusing on the architectural framework necessary for efficient translation, which is relevant to the syntactic processing in our project.

Moryossef (2023) [7] highlights recent developments in real-time translation capabilities using multilingual models, which are directly applicable to enhancing the live interaction potential of the SoundSigns platform.

Krňoul et al. (2008) [8] explore the adaptation of text-to-sign systems for the Czech language, offering insights into extending similar systems to other languages, such as expanding SoundSigns beyond ASL.

Finally, Filhol et al. (2016) [9] address the linguistic processing involved in converting text to sign language through a rule-based approach, providing a foundational understanding of automated translation techniques within digital applications.

These projects and papers illustrate the growing field of digital sign language translation tools, providing valuable methodologies and technological guidance for SoundSigns. By integrating advanced datasets like WLASL [2], implementing Whisper from ChatGPT for speech-to-text conversion, then using the ChatGPT API for converting English text to gloss [3], and utilizing 3D avatar creation technologies like Ready Player Me [10], SoundSigns aspires to contribute meaningfully to media accessibility for the deaf community.

# 3. Background

## 3.1. American Sign Language (ASL)

American Sign Language (ASL) is a visual language primarily used by the deaf community across the United States and parts of Canada. ASL is a fully developed, natural language with unique grammar and syntax that differs from spoken English. It utilizes manual gestures, facial expressions, and body movements to express meaning and emotion, serving as a vital component in communication for the deaf and hard of hearing. Grasping ASL's structural and semantic elements is crucial for creating effective translation systems that help the communication between hearing individuals and the non-hearing ones.

## 3.2. ASL Gloss

Glossing is a visual method used in linguistics to represent token signs. The ASL gloss is the translation of the ASL's visual-spatial components into a written format. This representation is not a direct English translation. It is a stripped-down version devoid of English grammatical nuances, focusing on the essential elements conveyed in the original ASL sentence. The ASL Gloss helps both interpreters and machines understanding of the unique aspects of the sign language.

## 3.3. Speech-to-Text Conversion Technologies

Speech-to-text conversion technologies play a pivotal role in facilitating communication between hearing and non-hearing individuals by transcribing spoken language into written form. These systems employ advanced algorithms and machine learning models, allowing them to recognize and process human speech with high accuracy and speed. In the context of the SoundSigns project, technologies such as Whisper by OpenAI are crucial as they form the first step in the translation pipeline, converting the audio input captured by the microphone into text form. This transcription serves as the groundwork for subsequent transformation into sign language gloss, ensuring a seamless experience for users interacting with the system.

## 3.4. Machine Learning and Natural Language Processing

Machine learning and natural language processing (NLP) are technologies that automate the language translations, such as English to ASL. By using models like ChatGPT, complex algorithms analyze and transform conventional English text into sign language glosses. This involves understanding the semantic and syntactic nuances of both languages to produce accurate translations. Employing transformer models and other advanced architectures helps in managing tasks such as context recognition, sentence restructuring, and cultural subtleties, thereby improving the system's overall reliability and effectiveness in real-world applications.

## 3.5. Mediapipe

The Google Mediapipe, is a tool for hand tracking, face detection, pose estimation, and object detection. The tool can be used to create avatars that mimic the movements of the person captured in the video or webcam, by extracting defined landmarks .

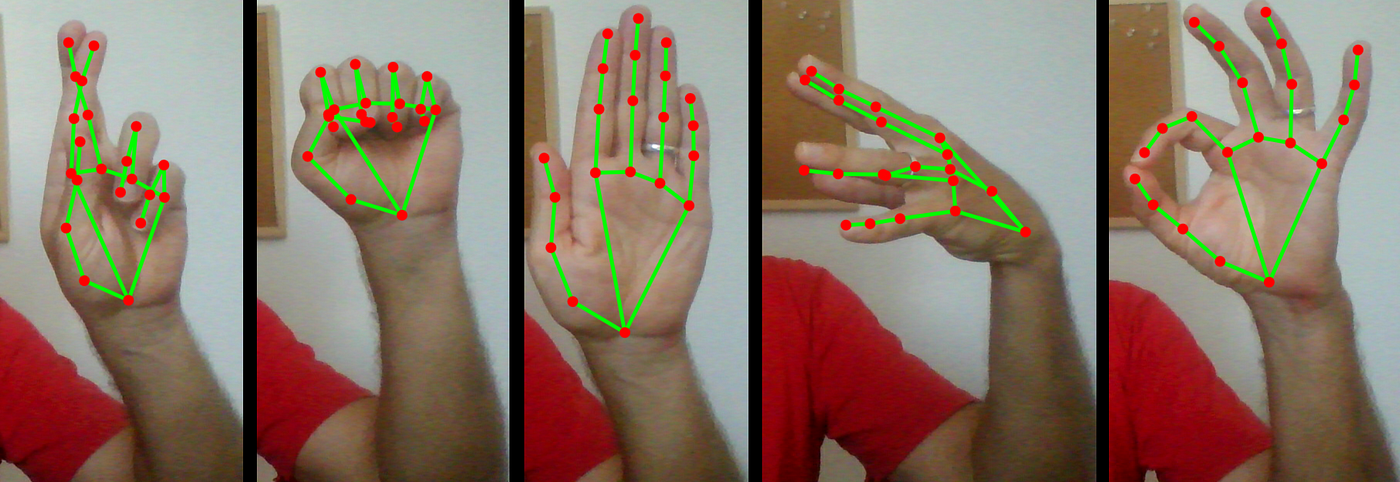


Figure 1. Extracting hand landmarks from the webcam.

It is particularly beneficial in projects like SoundSigns, where accurate gesture interpretation is required. By leveraging its robust real-time tracking capabilities, that the tool can be used to create 3D avatar that effectively simulates the nuances of sign language. The framework's modularity allows easy integration with other tools and systems, enhancing its utility across various applications. Furthermore, MediaPipe's ability to detect and process multiple landmarks simultaneously enables high-fidelity motion capture, crucial for replicating sign language's complex hand and body gestures. This enhances digital interpreters' overall realism and effectiveness, supporting more natural interactions between users and the system.

## 3.6. WLASL Dataset

WLASL [2] is currently the largest video dataset available for Word-Level American Sign Language (ASL) recognition. It encompasses 2,000 commonly used words in ASL, making it a valuable resource for advancing research in this field. The dataset includes two main components: the WLASL\_v0.3.json file, which provides a detailed glossary and metadata for the videos, and a "videos" folder containing approximately 12,000 video clips. Each video is named according to its corresponding video\_id, ensuring straightforward indexing and accessibility.

## 3.7. ChatGPT

ChatGPT, developed by OpenAI, is a powerful AI language model designed for natural, human-like conversations. It is built on the Generative Pre-trained Transformer (GPT) architecture, which leverages deep learning to generate coherent and contextually relevant text based on user inputs. Trained on an extensive dataset encompassing diverse textual information, ChatGPT can effectively assist with answering questions, drafting both formal and creative content, and offering detailed explanations. Its wide-ranging applications include customer support, where it can manage inquiries and provide solutions, and personal assistants, aiding in everyday tasks. By continuously learning from user interactions and updates, ChatGPT evolves to enhance its performance, making it a versatile tool in various fields that require advanced language capabilities.

## 3.8. 3D Avatars

3D avatars offer a dynamic and engaging way to present sign language interpretations through digital characters. Ready Player Me [10] is an advanced platform that enables to create highly customizable 3D avatars that can be integrated into various applications, enhancing user interaction and engagement. These avatars can mimic human sign language interpreters by replicating the necessary gestures and facial expressions accurately, providing a virtual solution for interpretive communication needs. The use of 3D avatars bridges realism and accessibility, making digital platforms more inclusive for the deaf community.

# 4. Expected Achievements

## 4.1. Outcomes

We anticipate that SoundSigns will deliver an intuitive online platform capable of translating spoken English into American Sign Language (ASL) in real time. The platform will feature a 3D animated avatar capable of performing accurate sign language interpretations, ensuring that media and live conversations are accessible to the deaf and hard-of-hearing communities. By prioritizing a seamless and engaging user experience, the platform will facilitate natural interactions and comprehension. Users can expect an easily navigable website with features designed to accommodate varying levels of technical proficiency, enhancing inclusivity and satisfaction.

## 4.2. Unique Features

### 4.2.1. Real-Time Audio-to-Sign Translation

At the core of SoundSigns is the real-time translation of audio input to sign language using 3D avatars. This feature involves integrating cutting-edge speech-to-text technology with natural language processing models like ChatGPT, followed by the animation of sign language via detailed motion capture data. The challenge lies in optimizing these processes to minimize latency and ensure accurate gesture representation.

### 4.2.2. Enhanced Gloss Transformation and Sentence Structuring

SoundSigns will leverage the ChatGPT API to perform enhanced gloss transformation, converting conventional English text into ASL gloss with accuracy and precision. This feature involves advanced natural language processing to dynamically reorder words and adjust grammar, ensuring the output aligns with ASL's unique syntax and preserves the original meaning and nuance. By maintaining context and addressing variations in tense, plurality, and emphasis, the system will deliver authentic and meaningful sign language translations.

### 4.2.3. Multi-Platform Accessibility

SoundSigns will be designed for multi-platform accessibility, ensuring that users can access the platform via various devices, including desktops, tablets, and smartphones. This flexibility allows users to interact with the sign language translation service wherever they are, contributing to broader reach and convenience.

### 4.2.4. User-Centric Interface

SoundSigns will feature a user-centric interface that emphasizes accessibility and ease of use. The platform will incorporate real-time feedback mechanisms, allowing users to see immediate results as their audio input is translated into sign language animations. Customizable settings will enable users to personalize their experience by selecting different avatar appearances, adding a layer of personalization to their interactions. Additionally, the platform will provide easily accessible help and support resources, offering guidance and answering common questions to enhance the overall user experience.

## 4.3. Criteria for Success

* **Accurate Real-Time Translation:** Achieve precise translation of spoken English into ASL signs, ensuring clarity and accuracy in gesture representation.
* **Effective Gloss Conversion:** Successfully convert English text to ASL gloss, maintaining the integrity and nuances of the language.
* **Multi-Platform Functionality:** Ensure smooth operation and accessibility across a range of devices and platforms, including desktops, tablets, and smartphones.
* **Intuitive User Interface:** Develop an interface that is accessible and easy to use, catering to individuals with diverse technical proficiencies.

# 5. The Process

## 5.1 Transforming English Text to ASL Gloss

The first step in transforming an English voice into an animated avatar using American Sign Language (ASL) is converting English text into ASL gloss. This step involves the stripping of words down to their base forms and adapting the structure of the English sentences to align with the ASL grammar and syntax.

Additionally, non-English words are represented through fingerspelling (SABINE ARNAUD 2019) [1], a method of spelling out each letter of a word using specific hand signs. This process ensures a more accurate and natural representation of ASL, highlighting the language's distinct linguistic characteristics and grammar rules.

### 5.1.1 Constraints and Challenges - Transforming English Text to ASL Gloss

ASL uses a subject-object-verb (SOV) structure, unlike English’s subject-verb-object (SVO) order. Converting sentences like “The dog chased the cat” into the appropriate ASL gloss structure “DOG CHASE CAT” requires a precise understanding of syntax transformation. For example, “She ate the apple” would become “SHE EAT APPLE” in ASL gloss, where the auxiliary verb “ate” is omitted, and the sentence structure aligns with ASL’s more direct and efficient form. If indicating a specific time, such as "yesterday," the gloss might include the time element at the beginning, resulting in "YESTERDAY SHE EAT APPLE," which shows how time indicators are managed in ASL to set the context.

Articles (e.g., “the,” “a”), auxiliary verbs (e.g., “is,” “are”), and certain prepositions are typically omitted in ASL gloss. Determining which words to omit without losing meaning requires linguistic expertise. For instance, in ASL gloss, "I am happy" would be simplified to "I HAPPY," eliminating the auxiliary verb “am” for clarity and precision.

Words such as names, technical terms, and foreign phrases must be converted to fingerspelling. Managing this dynamically, especially with varying handshape recognition, is a key challenge. For example, “Einstein” would be fingerspelled as “E-I-N-S-T-E-I-N,” which requires a clear and accurate representation of each letter.

English words often depend on context for meaning. For example, “run” can mean to sprint, manage (e.g., run a company), or operate (e.g., run a machine). Accurately discerning context is critical for appropriate ASL gloss translation. In ASL, “run” as in sprinting would be signed differently from “run” in managing a business, emphasizing the need for contextual understanding.

Transforming compound or complex sentences with multiple clauses requires breaking them into simpler, conceptually coherent segments for ASL gloss. A sentence like “Although she was tired, she finished the task” might be broken down into “SHE TIRED FINISH TASK,” simplifying the structure while preserving the meaning.

### 5.1.2 Use in Project - Transforming English Text to ASL Gloss

ASL Gloss has syntax and a unique structure that is different from the plain regular English. it is a distinct and visually-based language that relies on facial expressions, handshapes, and spatial relationships to convey meaning effectively.

We use ChatGPT, to transform English text into ASL gloss. ChatGPT can effectively translate simple sentences and basic concepts. However, a limitation we face is that tools capable of translating complex syntax and dynamic contexts with high accuracy are currently nonexistent.

## 5.2 Motion Capture

The phrase "Motion Capture" is well-known in the world of cinema because it is integral to creating Computer-Generated Imagery (CGI). CGI is generated by capturing a person's movements and using specialized tools to extract landmarks from the footage. These movements are then recorded and used within a rendering engine, such as ThreeJS or Unity, to produce lifelike animations. ThreeJS is a popular JavaScript library that simplifies the creation and display of animated 3D graphics directly in web browsers. It provides tools and features that allow developers to render complex 3D scenes efficiently, making it accessible for projects that require intricate visual effects without the need for extensive plugin installations. CGI, facilitated by tools like ThreeJS and Unity, allows filmmakers to create complex visual effects and realistic characters, enhancing storytelling by blending seamlessly with live-action footage.

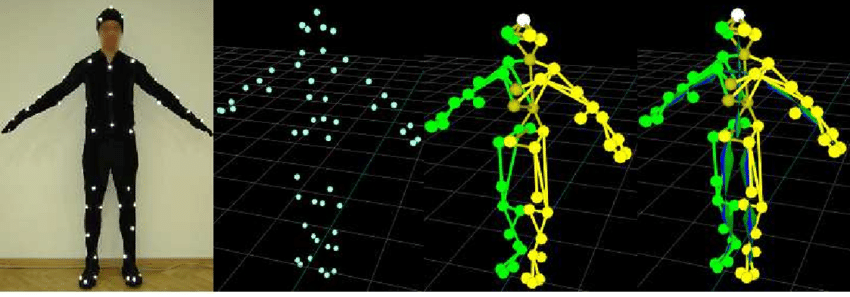


Figure 2. Motion Capture Suit to Skeleton made from Landmarks

We will use MediaPipe, a free software that can take video input and produce the coordinates of each body part to be recorded.

MediaPipe employs several methods to capture various human motions, including:

1. Hand Tracking
2. Pose Estimation
3. Face Mesh
4. Holistic
5. Objectron
6. BlazePose
7. Gesture Recognition
8. Eye Tracking

We will focus on two key areas: Hand Tracking and Pose Estimation. Face Mesh is not necessary because it is hyper-detailed, and we do not require such intricate details for our subsequent steps.

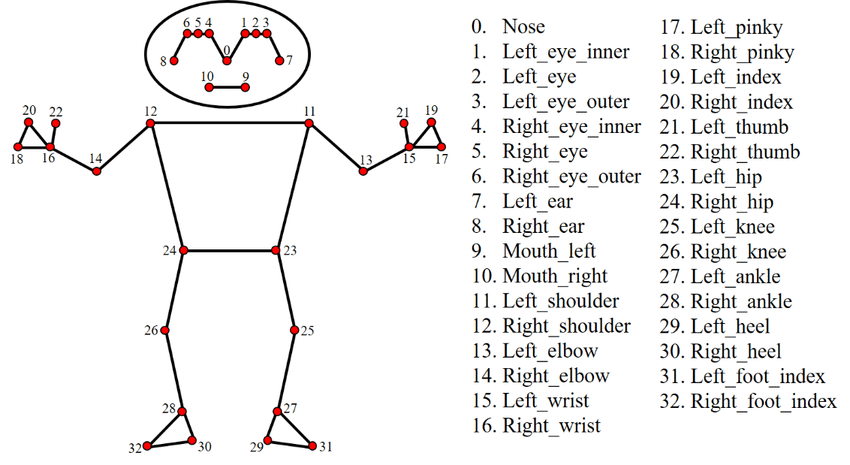


Figure 3. Landmarks of Pose Estimation by MediaPipe

### 5.2.1 Constraints and Challenges - Motion Capture

MediaPipe offers pose estimation and hand tracking for capturing motion, and there is some overlap between the two. Pose Estimation detects landmarks on the entire body, including the hands. However, since Hand Tracking provides more detailed and accurate tracking of the hands, we need to exclude the hand landmarks from Pose Estimation and rely on Hand Tracking for better hand detection and tracking.

Another challenge is the need to capture the motion of individuals performing ASL. Fortunately, there are many datasets available featuring professional signers. The dataset we will use, as mentioned, contains 5 GB of videos that can be converted into motion capture data.

### 5.2.2 Use in Project - Motion Capture

In conclusion, MediaPipe provides an accessible and cost-effective solution for motion capture, offering several modules that can track human motion in real-time. For our purposes, Hand Tracking and Pose Estimation are the most relevant features, as they capture detailed body movements without requiring expensive equipment or specialized software.

While Pose Estimation offers comprehensive tracking of body landmarks, including those for the hands, Hand Tracking delivers superior precision and detail specifically for hand movements, which is crucial for capturing American Sign Language (ASL). Therefore, we will exclude the hand landmarks from Pose Estimation and rely solely on Hand Tracking to achieve better accuracy in ASL motion capture.

Moreover, the availability of datasets featuring professional signers performing ASL enables us to effectively use MediaPipe for motion capture. These datasets allow us to convert video recordings into accurate movement data, suitable for animation, augmented reality, or further analysis applications. This approach presents an efficient and budget-friendly alternative to traditional motion capture techniques used in CGI production.

## 5.3 Avatars with Motion Capture

After obtaining the motion capture data from MediaPipe, the next step is the translating of the motion data into a "live" avatar. This process, which involves mapping the captured motion data to a digital model, has been widely used in various industries, from gaming to film production. Many applications and technologies exist to bring motion capture data to life in virtual environments.

This step involves the mapping of the landmarks obtained from Pose Estimation and Hand Tracking to an avatar’s corresponding body parts. This is typically done using rigging, where a 3D model is set up with a skeleton structure that can move in response to the input data. The captured coordinates, such as joint positions and hand movements, are used to animate the model in real time.

Unity and Unreal Engine are popular platforms for real-time avatar rendering. Both support integration with motion capture data and allow for smooth avatar animation in response to live input. Three.js, a JavaScript library for 3D graphics, can also be used for web-based applications to animate avatars using motion capture data.

Our project will use an avatar that can effectively mimic the hand and body movements of individuals performing American Sign Language (ASL). This will allow for seamless communication with virtual avatars, thus enhancing user interaction in educational and assistive contexts.

### 5.3.1 Constraints and Challenges - Avatars with Motion Capture

We will use Three.js, a JavaScript library for 3D rendering to render the motion in a web-based environment. Although Three.js is required to animate the avatar in real-time within a browser, allowing users to see their movements reflected in the digital model.

Additionally, we will create a pipeline between the backend and frontend parts to transfer the motion captured data efficiently. This will ensure a real-time control over the avatar’s movements, using the data processed by the backend and visualized through Three.js on the frontend.

### 5.3.2 Use in Project - Avatars with Motion Capture

Translating MediaPipe motion capture data into live avatars is vital for creating interactive, realistic experiences. Accurate mapping of Pose Estimation and Hand Tracking data onto 3D avatars ensures smooth, natural movements, crucial for ASL applications. Our project uses Three.js for web-based avatar rendering, a JavaScript library that simplifies 3D graphics creation and manipulation directly in the browser. Three.js handles avatar animation, rendering, and lighting, enhancing realism and accessibility without extra plugins. Integrating Three.js with MediaPipe's capabilities provides a seamless system for ASL education and assistive applications.

## 5.4 Methodology and Development Process

We will adopt an Agile methodology, which is crucial for our success, to guide our development process. By working in iterative cycles, we will break down feature delivery into smaller, manageable components. This approach allows us maximum flexibility and ease in implementing changes. The development process will be divided into the following stages:

1. Transforming English Voice to English Text: Using speech-to-text technologies like Whisper, we will convert spoken English input into written text format for further processing.

2. Transforming English Text into ASL Gloss: We will employ ChatGPT to translate the English text into ASL gloss. This involves adjusting the syntax to fit ASL’s unique grammatical structure.

3. Using MediaPipe for Motion Capture: MediaPipe will be used to accurately capture motion by analyzing video footage and extracting coordinates that represent body and hand movements.

4. Capturing Video Data with Python Script: A Python script will automate the processing of all videos in our dataset, converting them into motion capture data for analysis and use in subsequent stages.

5. Creating a Database for Captured Data: We will develop a structured database to store the labeled motion capture data. This database will serve as a reference for mapping ASL gestures.

6. Preparing a 3D Avatar: A 3D avatar will be prepared and customized for use in our application, capable of realistically performing ASL gestures based on the captured data.

7. Writing a Three.js Script: We will write a script using Three.js to animate the avatar. This script will use the captured motion data to drive the avatar's movements accurately.

8. Utilizing ASL Gloss to Pull Data: By mapping ASL gloss terms to our database, we will retrieve the appropriate motion data required for each gesture.

9. Developing Frontend to Backend Pipelines: We will establish pipelines to transmit English text from the frontend interface to the backend for processing.

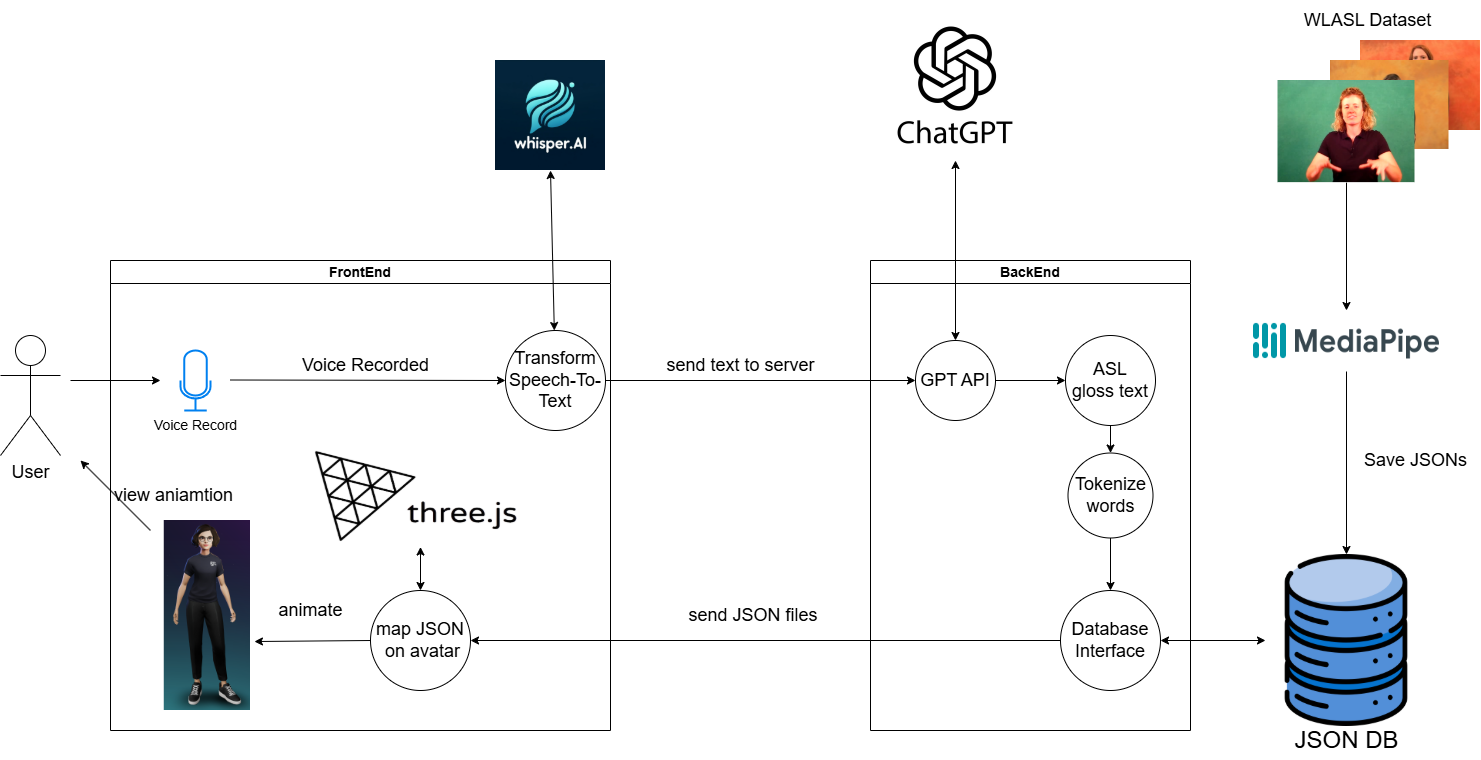
10. Developing Backend to Frontend Pipelines: Similarly, we will create pipelines to send the processed motion capture data back to the frontend for avatar animation.

11. Creating a Web Application: Finally, we will integrate all these components into a cohesive web application, ensuring seamless operation and user interaction.

At the end of each iteration, we will conduct evaluations and implement any necessary changes based on user feedback. Our primary focus will be delivering functional software while continuously improving based on insights gained during testing.

# 6. The Product

## 6.1. Architecture Overview



Our architecture consists of several key components, following the process order:

1. Voice Record: The process begins with the user recording their voice input through the microphone on the frontend interface.

2. Transform Speech-To-Text: Using WhisperAI, the recorded voice is converted into text in the frontend, minimizing the computational load on the backend.

3. GPT API: The text is sent to the backend, where the ChatGPT API translates it into ASL gloss using a carefully crafted prompt designed for this purpose.

4. Tokenize Words: The ASL gloss text is tokenized to prepare it for mapping, ensuring each word is ready for its corresponding gesture.

5. JSON DB: MediaPipe processes professional ASL videos to extract landmarks and coordinates, saving this data as JSON files in a MongoDB database. Each file corresponds to a specific ASL sign.

6. Save JSONs: The JSON format data is securely stored in the database for quick retrieval and use.

7. Map JSON on Avatar: The 3D avatar is animated using Three.js, guided by the JSON files to accurately replicate the ASL gestures.

8. View Animation: Finally, the user can view the 3D avatar performing the animations, which mirror the ASL translations, allowing for real-time interaction and visualization.

### 6.1.1 SoundSigns Application

SoundSigns is an innovative accessibility web application designed to bridge the communication gap between spoken language and American Sign Language (ASL). SoundSigns works by capturing the user’s voice, transcribing the speech into text, converting the text into ASL gloss (a simplified ASL structure), and animating a 3D avatar to perform the corresponding signs. The app architecture uses cloud-based services and client-side rendering to ensure fast and reliable performance.

#### **Components of the SoundSigns System:**

1. **Frontend:**
   * **Voice Recording Module**: Captures the user’s speech and prepares it to be processed.
   * **Speech-to-Text Service (Whisper API)**: Converts recorded speech into text with high accuracy, even in noisy environments.
   * **3D Animation Module**: Uses Three.js to render a customizable 3D avatar that performs ASL gestures based on JSON animation data.
2. **Backend:**
   * **Language Processing Service (GPT API)**: Transforms transcribed text into ASL gloss, maintaining context and grammatical nuances specific to ASL.
   * **Tokenization and Animation Mapping**: Tokenizes the ASL gloss and retrieves pre-built JSON animation files corresponding to each word or phrase from the database.
   * **Database Service (JSON DB)**: Stores animation data (JSON format), ensuring quick access during live sessions.
3. **Avatar Animation and Control:**
   * **Three.js Integration**: Maps JSON animation data onto the avatar in real-time, ensuring fluid and expressive signing movements.
4. **Real-Time Workflow:**
   * **API Integration**: Relies on external APIs (Whisper and GPT) for speech processing and language transformation.
   * **Fast Data Exchange**: Optimized communication between frontend and backend ensures low latency, enabling near real-time ASL animations.

### 6.1.2 User interface

The user interface will consist solely of an avatar performing ASL and a microphone button for recording audio. A text bubble will display the Speech-To-Text results.



## 

## 6.2. Requirements

### 6.2.1 Functional

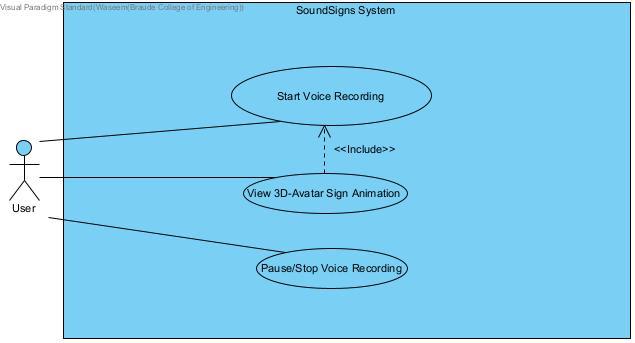
|  |  |
| --- | --- |
| ID | Description |
| 1 | The system shall allow users to access the platform via a web interface. |
| 2 | The system shall capture audio input from the user via a microphone for processing. |
| 3 | The system provides a visual indicator to notify users when speech is being recorded. |
| 4 | The system allows users to submit feedback on the tool’s performance, including avatar accuracy and usability. |
| 5 | The system shall display the 3D avatar performing the sign language animations in real-time. |
| 6 | The system shall allow users to replay sign language translations for review and learning purposes. |
| 7 | The system provides a responsive interface that works seamlessly on both mobile and desktop devices. |
| 8 | During the sign language interpretation session, the system shall support basic audio controls such as play, pause, and stop. |

### 6.2.2 Non-Functional

|  |  |
| --- | --- |
| ID | Description |
| 1 | The system shall provide a responsive and intuitive user interface to ensure ease of use for users of varying technical abilities. |
| 2 | The system shall perform translations and animations with minimal latency to provide a seamless real-time experience. |
| 3 | The system shall be designed to handle multiple simultaneous user sessions efficiently without degradation in performance. |
| 4 | The system shall ensure the security and privacy of user data, complying with applicable regulations and standards. |
| 5 | The system shall be built on a scalable architecture to accommodate future additions of sign languages and feature enhancements. |
| 6 | The system shall offer cross-browser compatibility to ensure accessibility across different web browsers. |
| 7 | The system shall be robust, ensuring consistent operation under typical and peak usage scenarios. |
| 8 | The system shall provide comprehensive error handling and user feedback mechanisms to inform users of any issues encountered. |

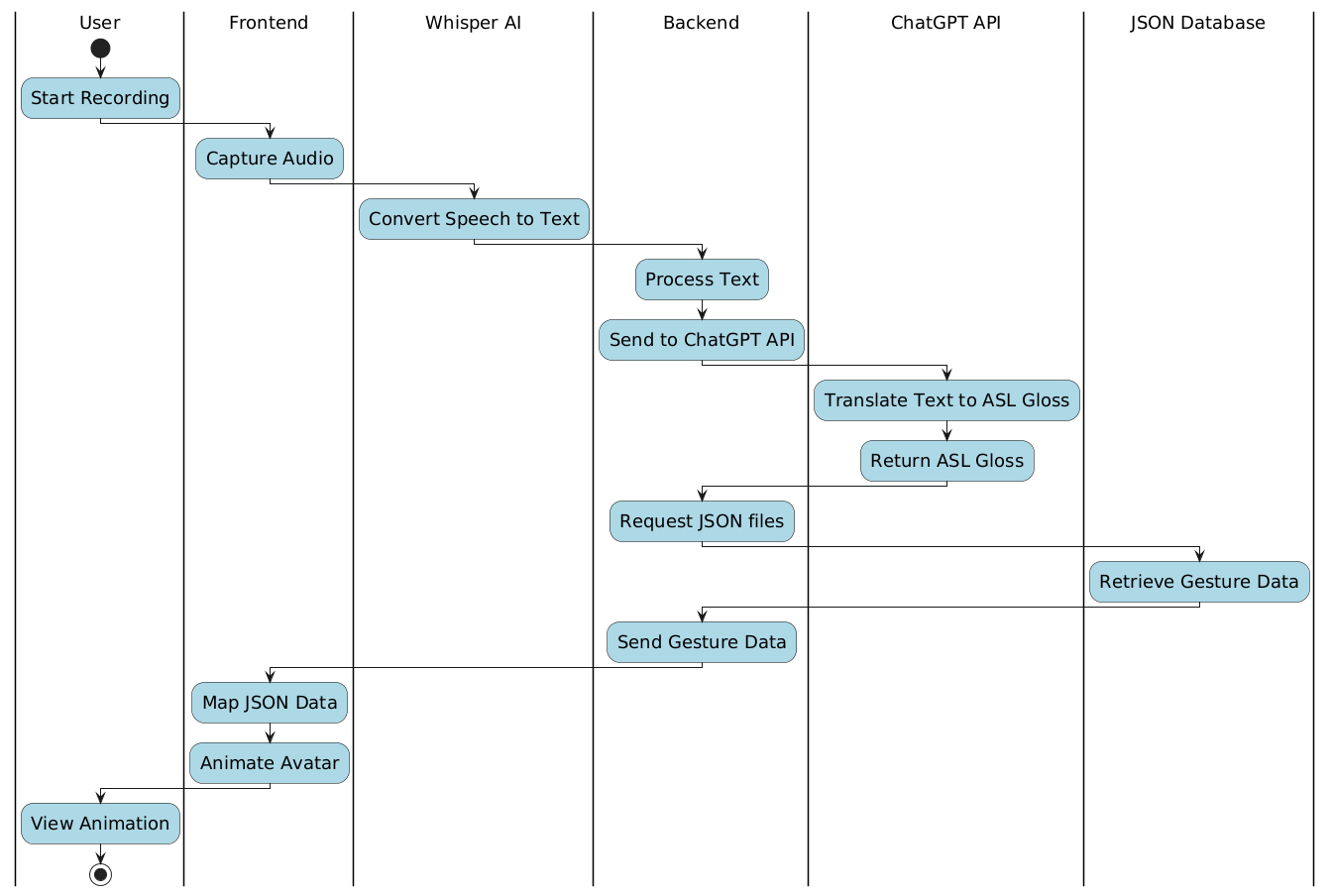
## 6.3. Diagrams

### 6.3.1 Use Case



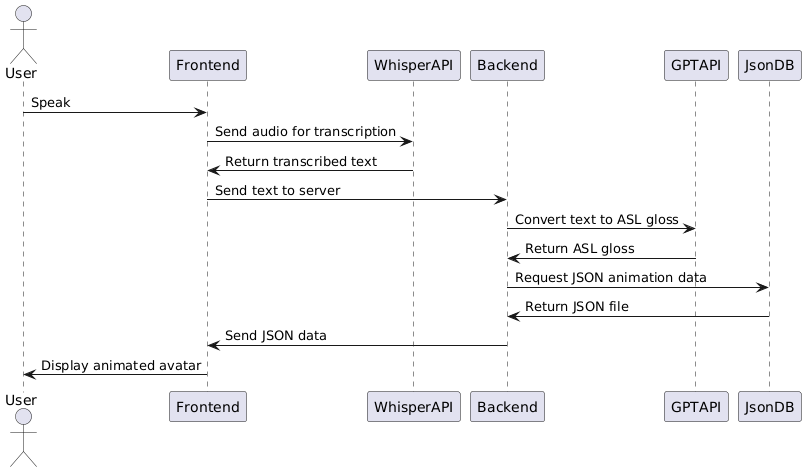
This use case diagram for the SoundSigns System illustrates the user interactions required for voice recording and avatar animation. The user initiates the process by selecting "Start Voice Recording," which serves as a precondition for "Pause/Stop Voice Recording." This sequential flow ensures that recording can only be paused or stopped after it has started. Additionally, the "View 3D-Avatar Sign Animation" is dependent on the inclusion of voice recording initiation, meaning the user must begin recording before the 3D avatar can display the sign animation. This structured process allows for a seamless user experience in synchronizing audio input with visual output.

### 6.3.2 Activity Diagram



This diagram illustrates the workflow for converting spoken input into ASL animations. The user records their voice, which is processed by the frontend to capture audio and then transcribed into text using Whisper AI. The backend sends the text to the ChatGPT API, which translates it into ASL gloss. The gloss is mapped to JSON data retrieved from a database, and the avatar animates the corresponding ASL gestures for the user to view.

### 6.3.3 Sequence Diagram

****

This diagram outlines the sequence of interactions in converting spoken input into ASL animations. The user speaks, and the frontend sends the audio to Whisper API for transcription. Once the text is returned, it is sent to the backend, which uses the GPT API to translate the text into ASL gloss. The backend requests JSON animation data from the database (JsonDB), which is returned and sent to the frontend to display the corresponding ASL animation on an avatar.

# 7. Verification and Evaluation

## 7.1. Evaluation

Involving the deaf community in our evaluation of SoundSigns will provide crucial insights and firsthand experiences with sign language interpretation. users can offer valuable feedback on translation accuracy, helping identify any discrepancies in ASL conversions. They can also evaluate the quality of the 3D avatar animations, assessing whether the movements are smooth, natural, and accurately represent ASL gestures. Their feedback on user experience, including ease of use and responsiveness, will help us to identify areas for improvement, such as reducing latency or enhancing accessibility. This input will ensure that SoundSigns will be an effective and practical tool for real-world applications, allowing us to refine the platform to better meet user needs and enhance overall satisfaction.

## 7.2. Verification

### 7.2.1 Testing

We will conduct a series of tests to evaluate SoundSigns's core functionalities, focusing on translation accuracy, animation quality, and system responsiveness. Each module, such as speech-to-text, gloss transformation, and sign animation, will be tested individually before performing end-to-end testing.

|  |  |  |  |
| --- | --- | --- | --- |
| Test ID | Module | Tested Function | Expected Result |
| 1 | User Interface | UI responsiveness and navigation | UI is responsive, easy to navigate, and works seamlessly across devices. |
| 2 | Audio Capture | Quality of audio input | Audio is captured clearly with minimal distortion. |
| 3 | Speech-to-Text Module | Accuracy of speech recognition | Transcribe speech into text accurately with minimal errors. |
| 4 | Speech-to-Text Module | Latency of speech recognition | Processes speech quickly with low latency. |
| 5 | Gloss Transformation | Accuracy of text-to-gloss | Converts English text to ASL gloss accurately, preserving meaning. |
| 6 | Gloss Transformation | Latency of gloss conversion | Converts text to gloss with minimal delay. |
| 7 | Animation Mapping | Accuracy of gesture animation | Animates gestures accurately according to gloss input. |
| 8 | Animation Mapping | Latency of animation rendering | Renders animations in real-time with minimal delay. |
| 9 | Multi-Device Support | Functionality across devices | Operates smoothly across desktops, tablets, and smartphones. |
| 10 | User Settings | Avatar customization | Users can change avatar appearances without issues. |
| 11 | Overall System | End-to-end accuracy | The system accurately integrates all components for a seamless experience. |

# 

# 

# 8. References

* [1] ARNAUD, S. (2019). Fingerspelling and the Appropriation of Language: The Shifting Stakes of a Practice of Signs. *Sign Language Studies*, *19*(4), 565–605. <https://www.jstor.org/stable/26786328>
* [2] Dongxu Li, & Hongdong Li. (2021). \*WLASL: Word-Level American Sign Language Video\* [Data set]. The Australian National University. Licensed under the Computational Use of Data Agreement (C-UDA). <https://www.kaggle.com/datasets/risangbaskoro/wlasl-processed/data>
* [3] OpenAI. (2024). *ChatGPT* [Large language model]. <https://chat.openai.com/chat>
* [4] Sanaullah, M., Ahmad, B., Kashif, M., Safdar, T., Hassan, M., Hasan, M. H., & Aziz, N. (2022). A real-time automatic translation of text to sign language. Computers, Materials & Continua, 70(2). <https://cdn.techscience.cn/ueditor/files/cmc/TSP_CMC_70-2/TSP_CMC_19420/TSP_CMC_19420.pdf>
* [5] Kahlon, N.K., Singh, W. Machine translation from text to sign language: a systematic review. *Univ Access Inf Soc* **22**, 1–35 (2023). <https://doi.org/10.1007/s10209-021-00823-1>
* [6] Sáfár, É., & Marshall, I. (2001, September). The architecture of an English-text-to-Sign-Languages translation system. In *Recent advances in natural language processing (RANLP)* (pp. 223-228). Tzigov Chark Bulgaria.‏<https://www.visicast.cmp.uea.ac.uk/Papers/confbulgarianew.pdf>
* [7] Moryossef, A. (2023). sign. mt: Real-Time Multilingual Sign Language Translation Application. *arXiv preprint arXiv:2310.05064*.‏ <https://arxiv.org/pdf/2310.05064>
* [8] Krňoul, Z., Kanis, J., Železný, M., Müller, L. (2008). Czech Text-to-Sign Speech Synthesizer. In: Popescu-Belis, A., Renals, S., Bourlard, H. (eds) Machine Learning for Multimodal Interaction. MLMI 2007. Lecture Notes in Computer Science, vol 4892. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-540-78155-4_16>
* [9] Filhol, M., Hadjadj, M.N. & Testu, B. A rule triggering system for automatic text-to-sign translation. *Univ Access Inf Soc* **15**, 487–498 (2016). <https://link.springer.com/article/10.1007/s10209-015-0413-4>
* [10] Ready Player Me. \*Avatar Creator for Developers\*. Wolf3D. <https://readyplayer.me>