
Zoom Lang Buddy

Team Members:

1. Ravi Rajappa
2. Maanya S
3. Shraddha Pandey
4. Gautham Vijayaraj



A project pitch by DIGITAL DEVILS 🐱🔥
(or at least that's what it's supposed to be)

Breaking The Sound Barrier in Zoom Classes

Imagine being in a lecture... but the only thing you hear is your stomach growling.

Many Deaf and Hard-of-Hearing students still miss context, emotion, and nuance in lectures.

Live captions ≠ ASL understanding.

Or Imagine if English isn't your first language. And you can't keep up with the classes.

Result? Confusion 🤪, disengagement 😞, and graduation delays 🧑🏫💔.



SCENARIO 1

 zoom

Meet Alex – CS Major, Passionate, Intelligent but **Deaf**

So Alex is a CS Student attending lectures on Zoom.

He couldn't keep track of the close captions. And guess what?

He missed technical terms like “binary trees” and “quicksort.”

And now **nothing makes sense to him.**



zoom

SCENARIO 2

 zoom

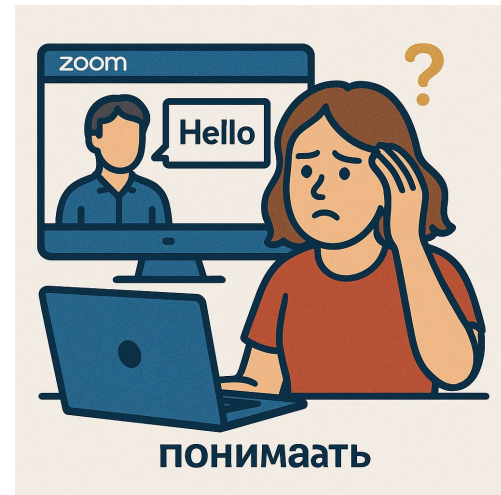
Meet Alena – CS Major, Russian, Smart but **NOT FLUENT IN ENGLISH**

English is not Alena's first language.

She needs to wait till she gets the zoom class's entire recordings and translate the transcripts.

She is new to this country and she is finding it hard to keep up with the professors.

She needs a fix that can help her keep up with the classes and feel like home in a new country.



zoom

SOLUTION??

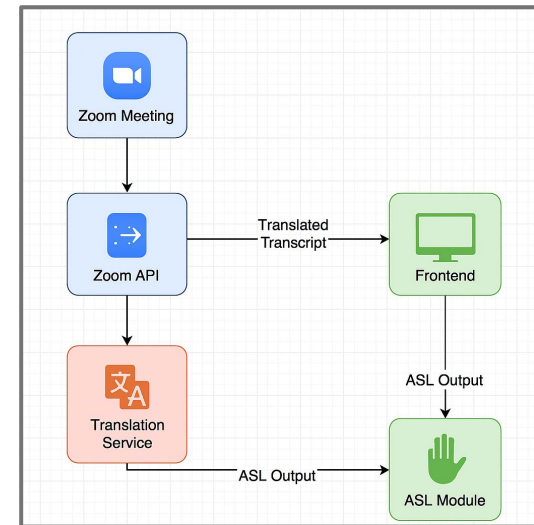


LANG BUDDY – The Real MVP 💪

Presenting to you proudly Lang buddy - your new sleek companion window

A Real-time solution that captures prof's voice → live transcript → matches keywords to ASL signs.

And not only that, we can also translate those transcripts to any language that the student is prefers.



Built on the shoulders of the RTSM

Our project began with the **RTSM (Real-Time Speech-to-Meaning)** base, which captures live Zoom audio and transcribes it instantly.

But we wanted more than just captions. We wanted it to **talk in hands and be multilingual.**

Zoom Speech → RTSM Event → Translation/ASL Conversion → Live Display



THE IMPLEMENTATION

- So we hooked into the RTSM WebSockets and routing logic (**rtms.js**) to fetch and stream live Zoom transcription events.
- Redirected this to a Flask backend (**app.py**) that:
 - Handles translation (with Google Translator).
 - Converts speech into cleaned, displayable ASL letters.
- Built a responsive frontend (**index.html**) where users can choose the language that they prefer, "ASL" for example, and watch the magic unfold, character-by-character



Add Ons

Added a **new Flask route** (`/stream`) to dynamically return real-time translated text.

Introduced **ASL character matching** using preloaded images (`asl_images/*.jpeg`).

Designed a playful **JS animation loop** that:

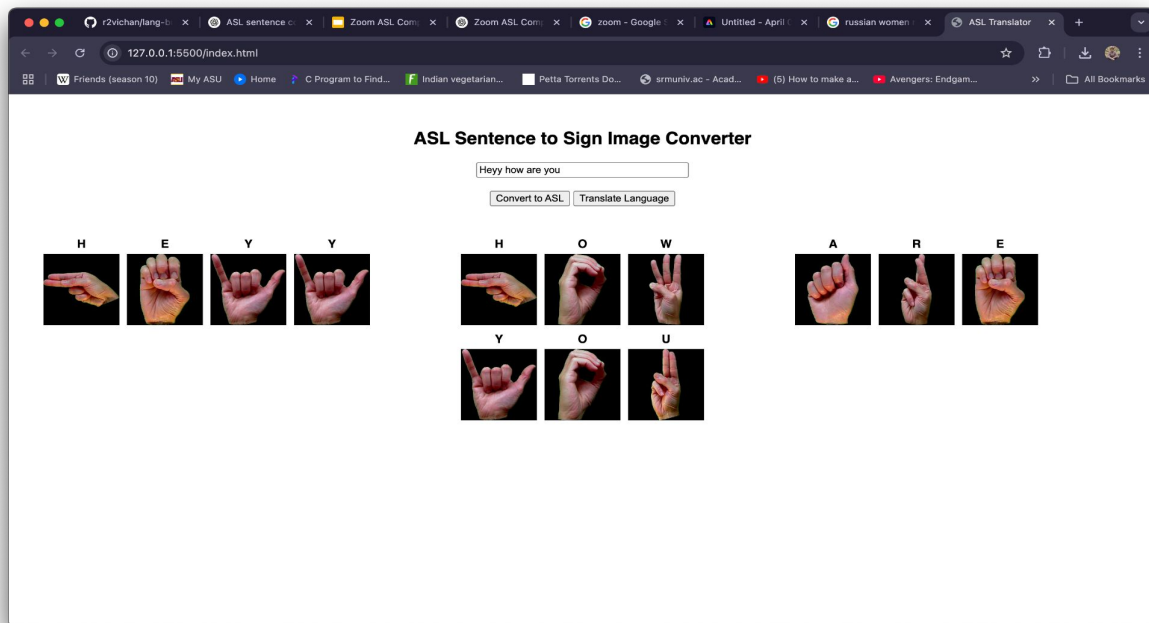
- Cleans the text.
- Renders images with slight popping animations for each character — like ASL karaoke



RESULTS

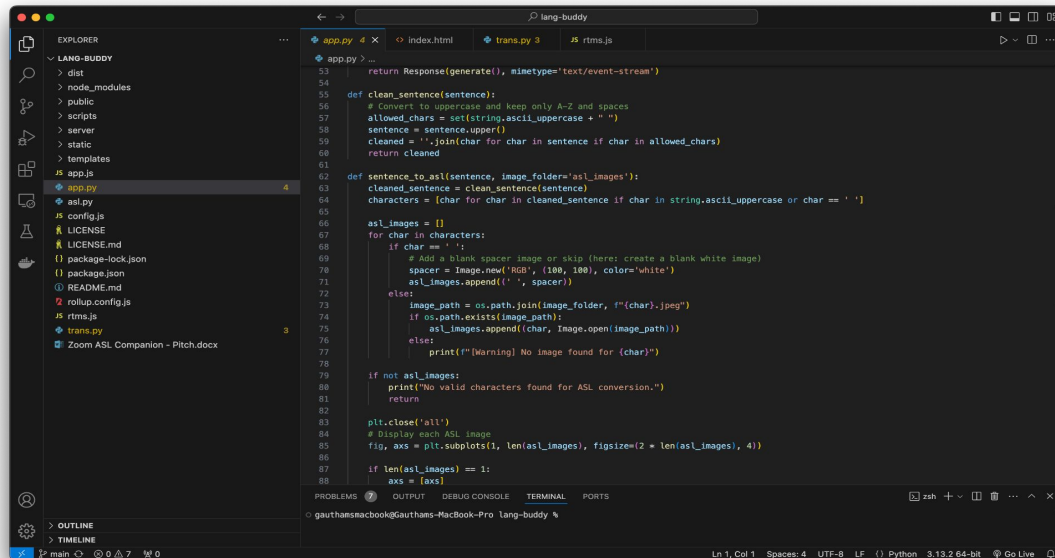


DUMMY CONCEPT (skip to the next)



zoom

THE SOURCE CODES (check the repo for more)



```
53 return Response(generate(), mimetype='text/event-stream')
54
55 def clean_sentence(sentence):
56     # Convert to uppercase and keep only A-Z and spaces
57     allowed_chars = set(string.ascii_uppercase + " ")
58     sentence = sentence.upper()
59     cleaned = ''.join(char for char in sentence if char in allowed_chars)
60     return cleaned
61
62 def sentence_to_asl(sentence, image_folder='asl_images'):
63     cleaned_sentence = clean_sentence(sentence)
64     characters = [char for char in cleaned_sentence if char in string.ascii_uppercase or char == ' ']
65
66     asl_images = []
67     for char in characters:
68         if char == ' ':
69             # Add a blank spacer image or skip (here: create a blank white image)
70             spacer = Image.new('RGB', (100, 100), color='white')
71             asl_images.append(' ', spacer)
72         else:
73             image_path = os.path.join(image_folder, f"{char}.jpeg")
74             if os.path.exists(image_path):
75                 asl_images.append(char, Image.open(image_path))
76             else:
77                 print(f"[Warning] No image found for {char}")
78
79     if not asl_images:
80         print("No valid characters found for ASL conversion.")
81         return
82
83     plt.close('all')
84     # Display each ASL image
85     fig, axes = plt.subplots(1, len(asl_images), figsize=(2 * len(asl_images), 4))
86
87     if len(asl_images) == 1:
88         axes = [axes]
```

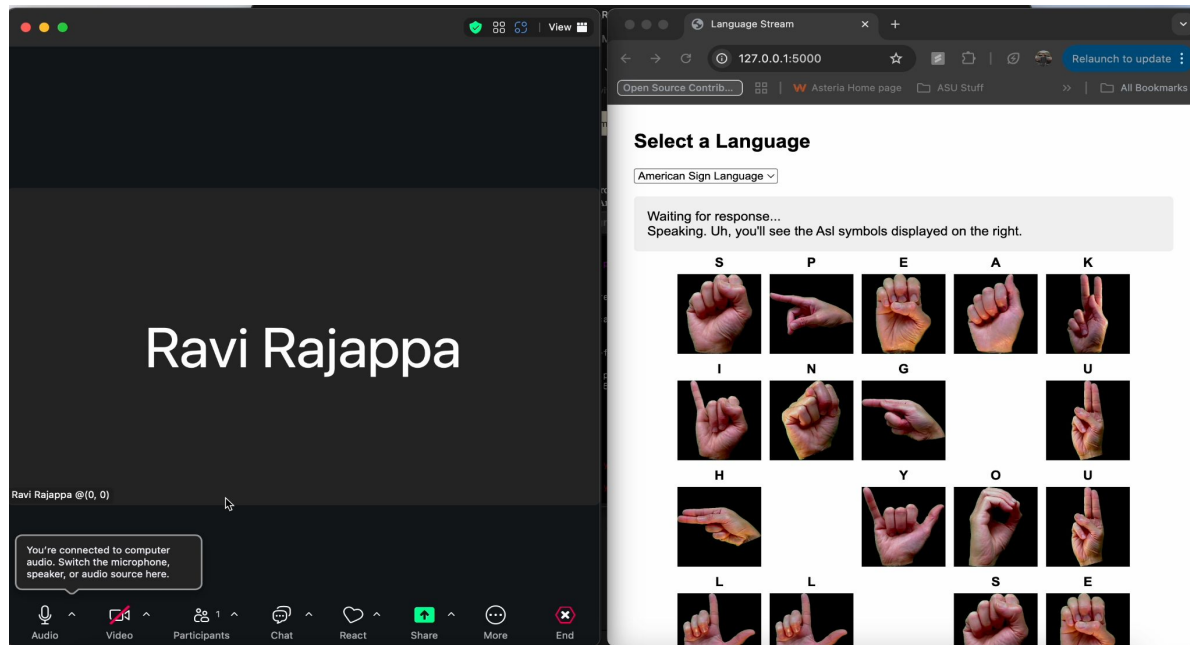


Repo Link: <https://github.com/r2vichan/lang-buddy>

Demo Link: https://www.youtube.com/watch?v=Eyd3IrurDew&ab_channel=RavichandranRajappa

Full Video Link: https://www.youtube.com/watch?v=eL800AUZlGY&ab_channel=RavichandranRajappa

Proof of Working Concept



zoom

THANK YOU !!

zoom