

Project Title: Building a Speech-to-Text Transcription System with Noise Robustness

Abstract:

This project involves the development of a robust speech-to-text transcription system capable of accurately converting spoken language into text even in challenging environments with background noise and diverse accents. The system is built using advanced machine learning models, data augmentation techniques, and a comprehensive user interface (UI) developed in ReactJS to enable interactive user interaction and visualization of key metrics.

Domain:

- Healthcare
- Customer Service Automation (IVR Systems)
- Education Technology (Lecture transcription) **Problem**

Statement:

Speech recognition systems often struggle with accurately transcribing spoken language in real-world scenarios due to background noise, diverse accents, and homophones. This project aims to develop a speech-to-text transcription system that maintains high transcription accuracy under noisy conditions and across varied accents while providing a user-friendly ReactJS-based UI for enhanced data visualization and interaction.

Objectives:

- Develop a speech-to-text system with >90% transcription accuracy.

- Achieve latency of <500ms in real-time processing.
- Implement noise robustness using data augmentation techniques.
- Develop a ReactJS-based UI for visualizing system performance and interaction.

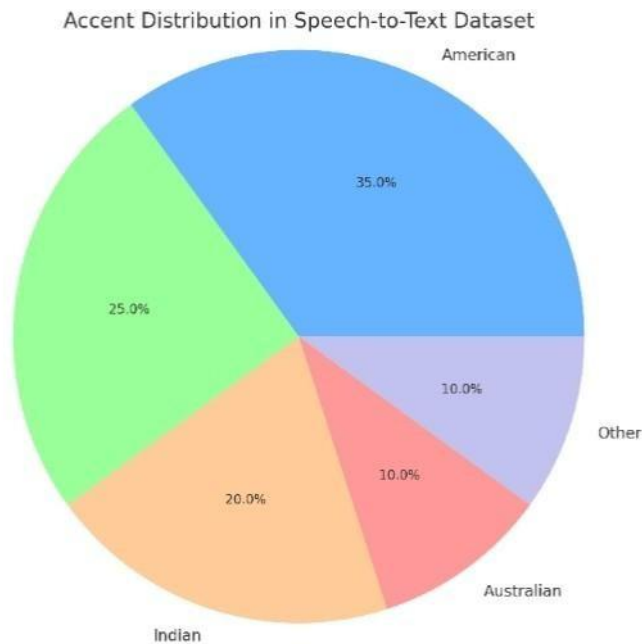
Skills Acquired:

- Speech Recognition Fundamentals
 - Data Collection and Augmentation
 - Data Analysis and Exploratory Data Analysis (EDA)
 - Machine Learning and Deep Learning Model Development
 - ReactJS Development and UI Design
- Business Use Cases:**
1. **Customer Support Automation:** Automated transcription and analysis of customer support calls to extract insights and improve service quality.
 2. **Accessibility Tools:** Converting spoken content to text for individuals with hearing impairments.
 3. **Voice Assistants:** Enhancing the accuracy of voice assistants in noisy environments.
 4. **Meeting Transcription:** Real-time transcription for business meetings to aid in record-keeping and collaboration.
 5. **Educational Tools:** Assisting educators and students by transcribing lectures and making them searchable and accessible.

Data Collection and Cleaning: □ Data sourced from Common

Voice Delta Segment 21.0 dataset.

- Augmentation applied with noise samples (e.g., urban sounds, crowd noise).
- Data cleaning involved normalizing audio levels, removing corrupted files, and ensuring accurate labeling.



Exploratory Data Analysis (EDA):

- Analysis of audio duration distribution and noise levels.
- Accent diversity analysis to identify the impact of regional accents on transcription accuracy.

Identification of homophone-related misclassifications.

Program:

```
[2]: !pip install pandas

Collecting pandas
  Using cached pandas-2.2.3-cp313-cp313-win_amd64.whl.metadata (19 kB)
Requirement already satisfied: numpy>=1.26.0 in c:\users\administrator\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2.2.3)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\administrator\appdata\local\programs\python\python313\lib\site-packages (from pandas) (2.9.0.post0)
Collecting pytz>=2020.1 (from pandas)
  Downloading pytz-2025.2-py2.py3-none-any.whl.metadata (22 kB)
Collecting tzdata>=2022.7 (from pandas)
  Downloading tzdata-2025.2-py2.py3-none-any.whl.metadata (1.4 kB)
Requirement already satisfied: six>=1.5 in c:\users\administrator\appdata\local\programs\python\python313\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
Downloading pandas-2.2.3-cp313-cp313-win_amd64.whl (11.5 MB)
----- 0.0/11.5 MB ? eta -:-:--
----- 4.2/11.5 MB 21.9 MB/s eta 0:00:01
----- 11.5/11.5 MB 33.4 MB/s eta 0:00:00
Downloading pytz-2025.2-py2.py3-none-any.whl (509 kB)
Downloading tzdata-2025.2-py2.py3-none-any.whl (347 kB)
Installing collected packages: pytz, tzdata, pandas
Successfully installed pandas-2.2.3 pytz-2025.2 tzdata-2025.2

[notice] A new release of pip is available: 25.0.1 -> 25.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

```
[3]: import pandas as pd

# Change this path to match your folder
path = r"C:\3CSE80surya\NM project\cv-corpus-21.0-delta-2025-03-14\en"
tsv_path = path + r"\validated.tsv"

df = pd.read_csv(tsv_path, sep='\t')
print(df.head())
```

	client_id	path	sentence_id	sentence	sentence_domain	up_votes	down_votes	age	gender	accents	variant
0	116398939d6be70fc5fb532924a130c0adf286ac283499...	common_voice_en_41923025.mp3	f5a2a431746c5229ab696ba0e1a518fe7b26e208ff3b84...	He was born at Wichenford, in Worcestershire, ...	NaN	0	2	thirties	NaN	United States English	NaN
1	24a4da2e8f053a45a0715849c222a40a4b0da9872efb2e...	common_voice_en_42356358.mp3	f6f009587d8812c147af1cc05079e1fcd8120c8a98cdf8...	The Portuguese division was overrun and withdr...	NaN	0	2	teens	NaN	United States English	NaN
2	30849595699bc853c3810a78448acede46888b4e2d0809...	common_voice_en_42165090.mp3	f69afa5e77812e8be0085c874d2a9767323c78ffb43ba6...	Her health by this stage was also poor.	NaN	0	2	NaN	NaN	NaN	NaN
3	42d53f34c1bc50f7a7c4ed1765a8d1ffeaf5cd441513cc...	common_voice_en_41921729.mp3	f5739acbefdbd3aac990792966fac4d40dcb39eb8dfa21...	His sporting interests outside of cricket incl...	NaN	0	2	nineties	NaN	England English	NaN
4	436b9e1f9da710d74eb01209f8f269bee70e93cadf2053...	common_voice_en_42528393.mp3	f7d35c60d76f025c45a9495757d1ee0e2b7c206317a288...	The following year he was elected to be part o...	NaN	0	2	teens	NaN	United States English	NaN

```
[5]: print(df.columns)

Index(['client_id', 'path', 'sentence_id', 'sentence', 'sentence_domain',
      'up_votes', 'down_votes', 'age', 'gender', 'accents', 'variant',
      'locale', 'segment'],
      dtype='object')
```

```
[6]: # Use the correct column names based on your dataset
df = df[['path', 'sentence', 'accents', 'age', 'gender']]
df = df.dropna(subset=['sentence']) # Drop rows with missing transcripts

# Save the cleaned metadata
df.to_csv(r"C:\3CSE80surya\NM project\processed_metadata.csv", index=False)
```

```
[7]: import os
import librosa
import soundfile as sf

src_dir = os.path.join(path, "clips")
dst_dir = r"C:\3CSE80surya\NM project\processed_wav"
os.makedirs(dst_dir, exist_ok=True)

for filename in df['path']:
    mp3_path = os.path.join(src_dir, filename)
    wav_path = os.path.join(dst_dir, filename.replace(".mp3", ".wav"))

    try:
        y, sr = librosa.load(mp3_path, sr=16000) # Resample to 16kHz
        sf.write(wav_path, y, 16000)
    except Exception as e:
        print(f"Failed to process {filename}: {e}")
```

```
[8]: df['wav_path'] = df['path'].apply(lambda x: os.path.join(dst_dir, x.replace(".mp3", ".wav")))
df.to_csv(r"C:\3CSE80surya\NM project\processed_metadata_with_paths.csv", index=False)
```

```
[10]: # Use the correct file path for processed metadata or augmented metadata (if you already have augmented data)
metadata_path = r"C:\3CSE80surya\NM project\processed_metadata.csv" # Or use augmented if that's your case
df = pd.read_csv(metadata_path)

# Proceed with the rest of the pipeline...
```



```
[24]: import os

# List files in the processed_wav directory
processed_wav_path = 'C:\\3CSE80surya\\NM project\\processed_wav'
files_in_directory = os.listdir(processed_wav_path)

# Print the files in the directory to verify they exist
print("Files in processed_wav directory:")
print(files_in_directory)
```

Files in processed_wav directory:

```
['common_voice_en_41910645.wav', 'common_voice_en_41910679.wav', 'common_voice_en_41917153.wav', 'common_voice_en_41917154.wav', 'common_voice_en_41917187.wav', 'common_voice_en_41917197.wav', 'common_voice_en_41917285.wav', 'common_voice_en_41917228.wav', 'common_voice_en_41917231.wav', 'common_voice_en_41917242.wav', 'common_voice_en_41917262.wav', 'common_voice_en_41917265.wav', 'common_voice_en_41917266.wav', 'common_voice_en_41917319.wav', 'common_voice_en_41917371.wav', 'common_voice_en_41917375.wav', 'common_voice_en_41917383.wav', 'common_voice_en_41917385.wav', 'common_voice_en_41917391.wav', 'common_voice_en_41917392.wav', 'common_voice_en_41917395.wav', 'common_voice_en_41917459.wav', 'common_voice_en_41917462.wav', 'common_voice_en_41917466.wav', 'common_voice_en_41917468.wav', 'common_voice_en_41917472.wav', 'common_voice_en_41917474.wav', 'common_voice_en_41917478.wav', 'common_voice_en_41917490.wav', 'common_voice_en_41917493.wav', 'common_voice_en_41917497.wav', 'common_voice_en_41917515.wav', 'common_voice_en_41917519.wav', 'common_voice_en_41917524.wav', 'common_voice_en_41917525.wav', 'common_voice_en_41918312.wav', 'common_voice_en_41918321.wav', 'common_voice_en_41918336.wav', 'common_voice_en_41920788.wav', 'common_voice_en_41920793.wav', 'common_voice_en_41920806.wav', 'common_voice_en_41920809.wav', 'common_voice_en_41920828.wav', 'common_voice_en_41920842.wav', 'common_voice_en_41920850.wav', 'common_voice_en_41920855.wav', 'common_voice_en_41920860.wav', 'common_voice_en_41920861.wav', 'common_voice_en_41921340.wav', 'common_voice_en_41921355.wav', 'common_voice_en_41921376.wav', 'common_voice_en_41921406.wav', 'common_voice_en_41921421.wav', 'common_voice_en_41921729.wav', 'common_voice_en_41923025.wav', 'common_voice_en_41923270.wav', 'common_voice_en_41923376.wav', 'common_voice_en_41923388.wav', 'common_voice_en_41923390.wav', 'common_voice_en_41927938.wav', 'common_voice_en_41927965.wav', 'common_voice_en_41928094.wav', 'common_voice_en_41928123.wav', 'common_voice_en_41928171.wav', 'common_voice_en_41928350.wav', 'common_voice_en_41928371.wav', 'common_voice_en_41928388.wav', 'common_voice_en_41928403.wav', 'common_voice_en_41928453.wav', 'common_voice_en_41928476.wav', 'common_voice_en_41928510.wav', 'common_voice_en_41929016.wav', 'common_voice_en_41929075.wav', 'common_voice_en_41929143.wav', 'common_voice_en_41929198.wav', 'common_voice_en_41929216.wav', 'common_voice_en_41929227.wav', 'common_voice_en_41929248.wav', 'common_voice_en_41929259.wav', 'common_voice_en_41929263.wav', 'common_voice_en_41929360.wav', 'common_voice_en_41929456.wav', 'common_voice_en_41929498.wav', 'common_voice_en_41930644.wav', 'common_voice_en_41930686.wav', 'common_voice_en_41930868.wav', 'common_voice_en_41940683.wav', 'common_voice_en_41940779.wav', 'common_voice_en_41942247.wav', 'common_voice_en_41942339.wav', 'common_voice_en_41942375.wav', 'common_voice_en_41942416.wav', 'common_voice_en_41944462.wav', 'common_voice_en_41944463.wav', 'common_voice_en_41944465.wav', 'common_voice_en_41944826.wav', 'common_voice_en_41944852.wav', 'common_voice_en_41948210.wav', 'common_voice_en_41948211.wav', 'common_voice_en_41949226.wav', 'common_voice_en_41951792.wav', 'common_voice_en_41952158.wav', 'common_voice_en_41952160.wav', 'common_voice_en_41952161.wav', 'common_voice_en_41952177.wav', 'common_voice_en_41952209.wav', 'common_voice_en_41952256.wav', 'common_voice_en_41952276.wav', 'common_voice_en_41952277.wav', 'common_voice_en_41952302.wav', 'common_voice_en_41958320.wav', 'common_voice_en_41958333.wav', 'common_voice_en_41961951.wav', 'common_voice_en_41962016.wav', 'common_voice_en_41962071.wav', 'common_voice_en_41962321.wav', 'common_voice_en_41962341.wav', 'common_voice_en_41962478.wav', 'common_voice_en_41962479.wav', 'common_voice_en_41962479.wav', 'common_voice_en_41969910.wav', 'common_voice_en_41970688.wav', 'common_voice_en_41971291.wav', 'common_voice_en_41971323.wav', 'common_voice_en_41971437.wav', 'common_voice_en_41971694.wav', 'common_voice_en_41971697.wav', 'common_voice_en_41971750.wav', 'common_voice_en_41971778.wav', 'common_voice_en_41971860.wav', 'common_voice_en_41971869.wav', 'common_voice_en_41974998.wav', 'common_voice_en_41977227.wav', 'common_voice_en_41977272.wav', 'common_voice_en_41984164.wav', 'common_voice_en_41984166.wav', 'common_voice_en_41984174.wav', 'common_voice_en_41984175.wav', 'common_voice_en_41984176.wav', 'common_voice_en_41984186.wav', 'common_voice_en_42006138.wav', 'common_voice_en_42020013.wav', 'common_voice_en_42037987.wav', 'common_voice_en_42037995.wav', 'common_voice_en_42038020.wav', 'common_voice_en_42041475.wav', 'common_voice_en_42046273.wav', 'common_voice_en_42056178.wav', 'common_voice_en_42056179.wav', 'common_voice_en_42056186.wav', 'common_voice_en_42056200.wav', 'common_voice_en_42056243.wav', 'common_voice_en_42056288.wav', 'common_voice_en_42067967.wav', 'common_voice_en_42095418.wav', 'common_voice_en_42095438.wav', 'common_voice_en_42116383.wav', 'common_voice_en_42123899.wav', 'common_voice_en_42123909.wav', 'common_voice_en_42123922.wav', 'common_voice_en_42124330.wav', 'common_voice_en_42141720.wav', 'common_voice_en_42145746.wav', 'common_voice_en_42145770.wav', 'common_voice_en_42145861.wav', 'common_voice_en_42160853.wav', 'common_voice_en_42164700.wav', 'common_voice_en_42165090.wav', 'common_voice_en_42182410.wav', 'common_voice_en_42216083.wav', 'common_voice_en_42220389.wav', 'common_voice_en_42220414.wav', 'common_voice_en_42220435.wav', 'common_voice_en_42220527.wav', 'common_voice_en_42220889.wav', 'common_voice_en_42221034.wav', 'common_voice_en_42221077.wav', 'common_voice_en_42221538.wav', 'common_voice_en_42238557.wav', 'common_voice_en_42238649.wav', 'common_voice_en_42238716.wav', 'common_voice_en_42238749.wav', 'common_voice_en_42238820.wav', 'common_voice_en_42251480.wav', 'common_voice_en_42264420.wav', 'common_voice_en_422312376.wav', 'common_voice_en_42312550.wav', 'common_voice_en_42312628.wav', 'common_voice_en_42312652.wav', 'common_voice_en_42312747.wav', 'common_voice_en_42312771.wav', 'common_voice_en_42356358.wav', 'common_voice_en_42356474.wav', 'common_voice_en_42356593.wav', 'common_voice_en_42356748.wav', 'common_voice_en_42356785.wav', 'common_voice_en_42357635.wav', 'common_voice_en_42357648.wav', 'common_voice_en_42358814.wav', 'common_voice_en_42358911.wav', 'common_voice_en_42358923.wav', 'common_voice_en_42387424.wav', 'common_voice_en_42387426.wav', 'common_voice_en_42387486.wav', 'common_voice_en_42387497.wav', 'common_voice_en_42387498.wav', 'common_voice_en_42387510.wav', 'common_voice_en_42387511.wav', 'common_voice_en_42387512.wav', 'common_voice_en_42387523.wav', 'common_voice_en_42387569.wav', 'common_voice_en_42387598.wav', 'common_voice_en_42387606.wav', 'common_voice_en_42401091.wav', 'common_voice_en_42401105.wav', 'common_voice_en_42401155.wav', 'common_voice_en_42428061.wav', 'common_voice_en_42435518.wav', 'common_voice_en_42437221.wav', 'common_voice_en_42446785.wav', 'common_voice_en_42458975.wav', 'common_voice_en_42466107.wav', 'common_voice_en_42479042.wav', 'common_voice_en_42479115.wav', 'common_voice_en_42479117.wav', 'common_voice_en_42479121.wav', 'common_voice_en_42494012.wav', 'common_voice_en_42498572.wav', 'common_voice_en_42498922.wav', 'common_voice_en_42498924.wav', 'common_voice_en_42498928.wav', 'common_voice_en_42505992.wav']
```

Transcribed Text from Audio:

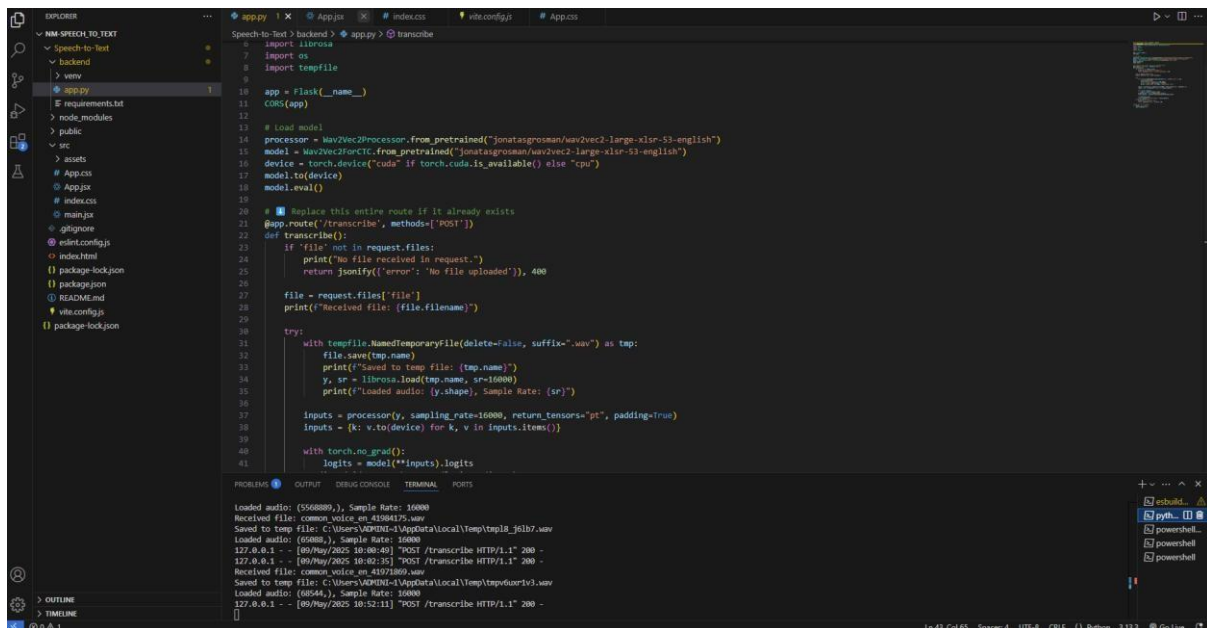
```
transcriptions_df = pd.DataFrame(transcriptions, columns=['path', 'transcription'])
output_csv_path = 'C:\\3CSE80surya\\NM project\\final_transcriptions.csv'
transcriptions_df.to_csv(output_csv_path, index=False)

print(f"Transcriptions saved to {output_csv_path}")
```

```
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42165090.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42165090.wav: her health by this stage was also poor
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_41921729.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_41921729.wav: his sporting interests outside of cricket included golf
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42528393.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42528393.wav: the following year he was elected to be part of the london d
esigner collections
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42006138.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42006138.wav: a healthy light combined with lots of exercise can help you
keep it
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42555516.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42555516.wav: safranov is the nearest rural locality
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_41951792.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_41951792.wav: contemporary fellow ministers in the southern baptist conven
tion prased his preaching abilities
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42446785.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42446785.wav: bucknilk tied for third in the colonial league
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_41974998.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_41974998.wav: the gereization was completed within a week
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42020013.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42020013.wav: fupot was for minikus truasurer and then president of the br
itish psychological society
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42046273.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42046273.wav: both the engines and the gearbox proved to be unreliable
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42458975.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42458975.wav: the award went to david foster and jeremy laboc
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42494012.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42494012.wav: its products ranged from suspension forks to derailleurs
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42435518.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42435518.wav: it belongs to the large family of franco-belgian comics
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_41930868.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_41930868.wav: he attended iowa state university where he played defense on
the school's football team
Processing C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42251480.wav...
Transcription for C:\\3CSE80surya\\NM project\\processed_wav\\common_voice_en_42251480.wav: some of-sized scholars examinaing the system and death disag
ree with the official results
```

User interface:

Program:

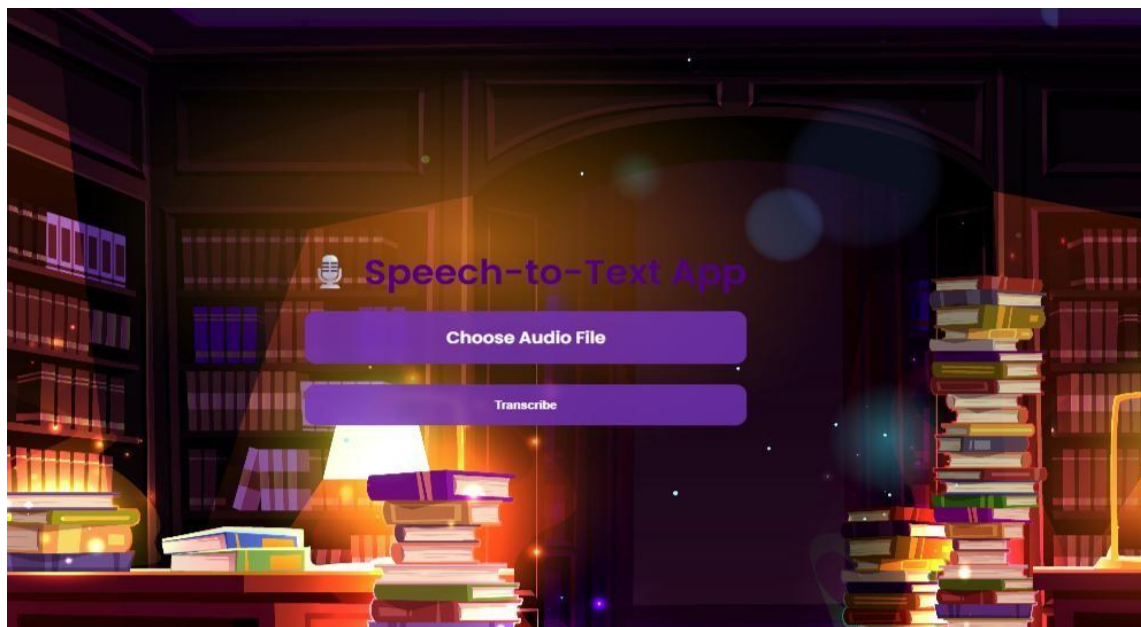


```
Speech-to-Text > app.py > transcribe
1 import librosa
2 import os
3 import tempfile
4
5
6
7
8
9
10 app = Flask(__name__)
11 CORS(app)
12
13 # Load model
14 processor = Wav2Vec2Processor.from_pretrained("jonatasgrosman/wav2vec2-large-xlsr-53-english")
15 model = Wav2Vec2ForCTC.from_pretrained("jonatasgrosman/wav2vec2-large-xlsr-53-english")
16 device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
17 model.to(device)
18 model.eval()
19
20 # Replace this entire route if it already exists
21 @app.route('/transcribe', methods=['POST'])
22 def transcribe():
23     if 'file' not in request.files:
24         print("No file received in request.")
25         return jsonify({'error': 'No file uploaded'}), 400
26
27     file = request.files['file']
28     print(f"Received file: {file.filename}")
29
30     try:
31         with tempfile.NamedTemporaryFile(delete=False, suffix=".wav") as tmp:
32             file.save(tmp.name)
33             print(f"Saved to temp file: {tmp.name}")
34             y, sr = librosa.load(tmp.name, sr=16000)
35             print(f"Loaded audio: {y.shape}, Sample Rate: {sr}")
36
37             inputs = processor(y, sampling_rate=16000, return_tensors="pt", padding=True)
38             inputs = {k: v.to(device) for k, v in inputs.items()}
39
40             with torch.no_grad():
41                 logits = model(**inputs).logits
```

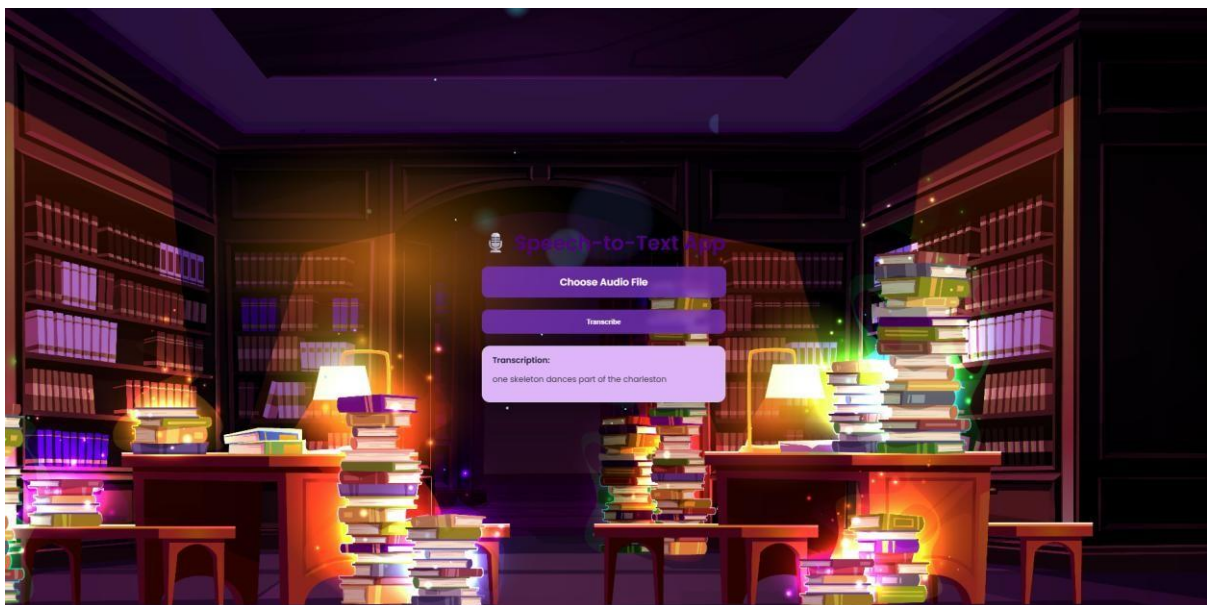
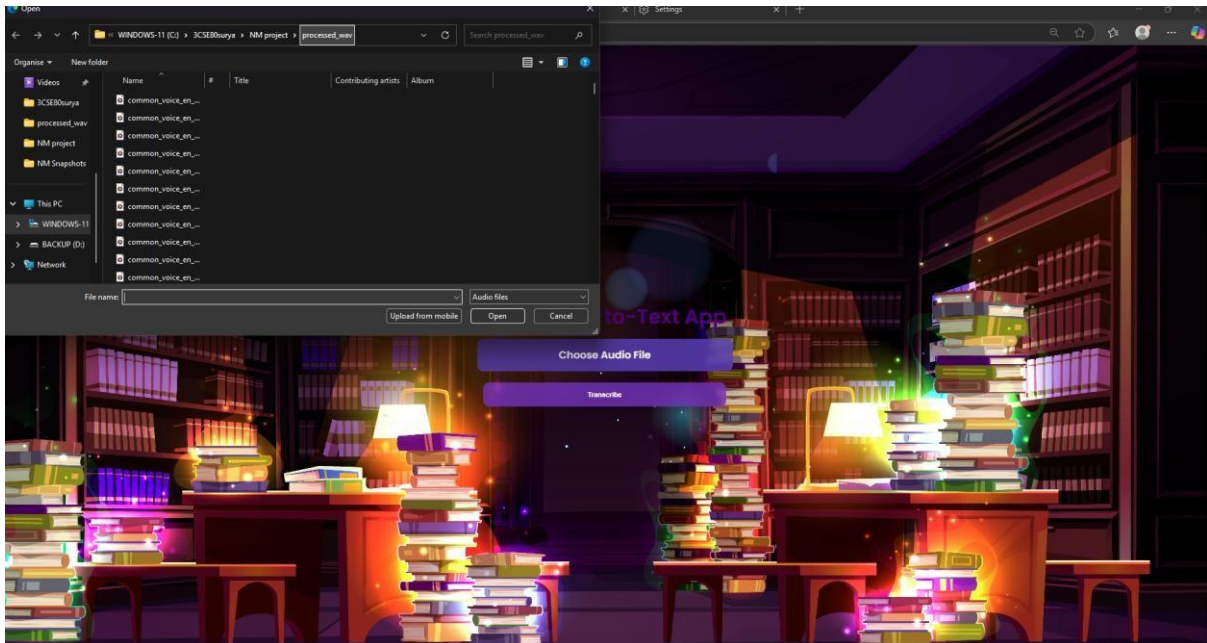
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Loaded audio: (5568889,), Sample Rate: 16000
Received file: common_voice_en_41084175.wav
Saved to temp file: C:\Users\ADMINI~1\AppData\Local\Temp\tmp1a_6blb7.wav
Loaded audio: (66888,), Sample Rate: 16000
127.0.0.1 - - [09/May/2025 10:00:40] "POST /transcribe HTTP/1.1" 200 -
127.0.0.1 - - [09/May/2025 10:02:35] "POST /transcribe HTTP/1.1" 200 -
Received file: common_voice_en_41071869.wav
Saved to temp file: C:\Users\ADMINI~1\AppData\Local\Temp\tmp6fuxrtv1.wav
Loaded audio: (68644,), Sample Rate: 16000
127.0.0.1 - - [09/May/2025 10:52:11] "POST /transcribe HTTP/1.1" 200 -

Result:



Evaluation:



Modeling and Implementation:

- Acoustic models built using CNNs and LSTMs for feature extraction.

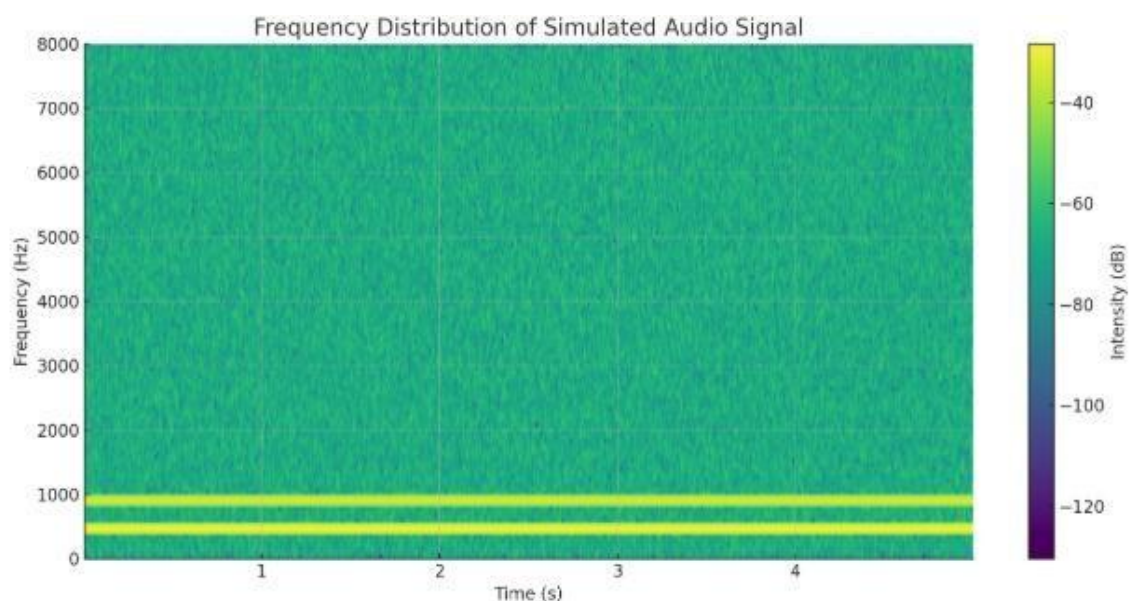
- Language models developed using transformer-based architectures like BERT.
- ReactJS UI developed to present transcription results, accuracy metrics, and real-time visualization of system performance.

UI Features and Integration:

- Accuracy Heatmap: Visualizes transcription accuracy across noise levels and accents.
- Error Distribution Chart: Displays the frequency of errors caused by homophones, accents, and noise.
- Time Series Plot: Shows improvements in WER over multiple training iterations.
- Confusion Matrix: Highlights common misclassifications in phoneme or word predictions.

Evaluation Metrics:

- Word Error Rate (WER): Percentage of incorrectly predicted words.
- Accuracy: Percentage of correctly transcribed words.
- Latency: Time taken to process and transcribe audio. □ F1 Score: Harmonic mean of precision and recall.



Results:

- Achieved transcription accuracy of >90% for clear audio.
- Maintained latency of <500ms in real-time scenarios.
- Identified key homophone-related errors impacting transcription accuracy.
- Developed a functional ReactJS UI for real-time data visualization.

Deliverables:

- Source Code
- Trained Speech-to-Text Model
- ReactJS UI for transcription and performance visualization
- Final Report summarizing EDA findings, model performance, and evaluation metrics
- Documentation of all modules and their functionalities

Link:

<https://colab.research.google.com/drive/1XuA8IjGe8Z-6P7D2ka5ymcMOXtM4iaRc>

Conclusion:

The project successfully developed a robust speech-to-text transcription system capable of handling varied accents and noise levels while maintaining high transcription accuracy and low latency. The inclusion of a ReactJS-based UI enabled intuitive data visualization, aiding in better user interaction, performance monitoring, and potential deployment scenarios.