BTP - 2

# Speech Transcription Verification

**Under the Guidance of Prof. Anil Kumar Vuppala**
**Mentor - Sai Aakarsh**

Project 1 : Team 2
Srujana Vanka - 2020102005
Shreeya Singh - 2020102011

# Objective

Develop an efficient Speech Transcription Validation System for Indian languages, incorporating automated audio segmentation, precise transcription generation, and a user-friendly interface with advanced features, including user authentication, transcription review, navigation, editing, synchronized audio playback, virtual keyboard support, and local storage.

# Introduction

The conversion of speech into text plays a pivotal role in various applications, including automatic captioning, audio search, and speech recognition. However, transcribing lengthy audio files is often time-consuming and prone to errors. The accuracy of these transcriptions holds paramount importance for application effectiveness.

Moreover, it's worth noting that, in the context of Indian languages, the availability of accurate ASR (Automatic Speech Recognition) models remains limited. This emphasizes the necessity for an efficient speech transcription verification system that ensures precision while reducing transcriber effort.

# Project Description

- Creating an **intuitive interface** for users to interact with the application, making the transcription verification process seamless and accessible.

- Enhancing Accessibility and Efficiency by ensuring **process resumption** to **save time** and effort for users.

- **Review and Editing Tools:** Users can review, edit, and play back synchronized audio, aided by virtual keyboards for accuracy.

- **Flexible Progress Management:** Users can save or discard transcripts, move to the next segment, or log out, with progress preservation.

- **Administrator Access:** Administrators can access login data for system transparency and enhanced accountability.

# Front-End

# Login Page

## User Authentication:

The login page is pivotal for transparency and data integrity.

- It collects transcribers’ names and email addresses

## File Selection:

Users utilize the login page to select the specific audio file they intend to verify, offering flexibility and customization.

## Language Selection:

The login page also enables users to designate their preferred language. This is used to generate the required virtual keyboard for Indian languages.

## Data Integrity:

Robust validation checks are incorporated to guarantee data integrity.

- These checks include verifying the completeness of details and validating the accuracy of entered email addresses.

# Interface

- Scrollable Navigation Panel
- Editable Text Area and Audioplayer
- Virtual Keyboard
- Save, Discard, Next buttons
- Progress Display
- Email Login data to Admin
- Logout Button

# Features and Functionalities

## Transcript Display

: Users are presented with the transcripts corresponding to the audio chunks they are verifying and can edit the existing transcripts in real-time, ensuring precision and accuracy in the verification process

## Audio Playback

Users can listen to the audio segments corresponding to the displayed transcripts. Furthermore, they have the ability to pause the playback, and can even control the playback speed for a customized verification experience

## Navigation Panel

Users can navigate to any transcript they wish to verify, and efficiently access specific transcripts using the search functionality. To enhance user productivity, the Navigation Panel exclusively displays remaining transcripts, automatically removing those that have been successfully verified.

## Progress Display

A quick snapshot of the verification progress displaying essential counts, such as:

- number of remaining transcripts
- saved transcripts
- discarded transcripts

## Virtual Keyboard

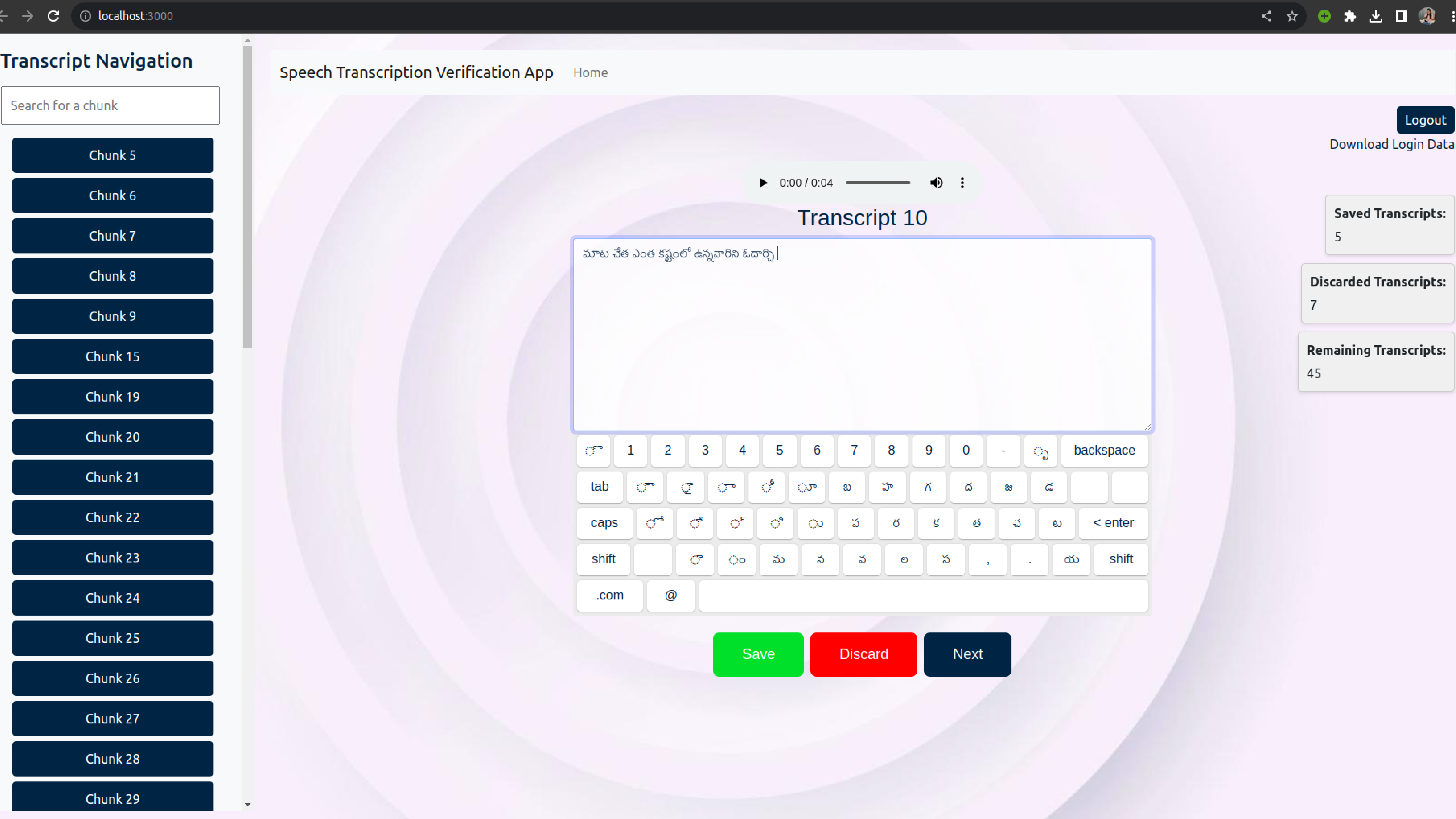
Offers virtual keyboard support, particularly for Indian languages, facilitating ease of input and verification.

## Mail Login Data

The system includes a feature to send login data to the admin. This feature compiles logs of transcribers’ login data and send the data through an email to the admin.

## Process Resumption

Whether logging back in or reloading the application, the system maintains each transcriber’s progress, eliminating the risk of duplicate verifications and ensuring that each transcript is reviewed only once



## Transcript Navigation

Speech Transcription Verification App

## Transcript 10

మాట చేత ఎంత కష్టంలో ఉన్నవారిని ఓదార్చి

**Saved Transcripts:** 5

**Discarded Transcripts:** 7

**Remaining Transcripts:** 45

# Back-End

# Backend Functionalities

- Node.js and Express.js Backend:

 - Foundation of the server built on Node.js and Express.js framework for robust system operations.

- Python Scripts for Processing:

 - Utilize child process module to seamlessly execute Python scripts for tasks like audio segmentation and transcription.

- Transcript Verification Preparation:

- Execute Python script to format transcripts and audio chunks, aligning data for frontend verification. The prepared folder is moved to the "original data" directory.

- Request Handling:

 - Backend listens on port 5000, leveraging body-parser middleware for efficient handling of incoming post requests, including saving or discarding transcripts and audio chunks.

# Technologies and Dependencies

**Programming Languages:** Python for backend development, JavaScript for frontend.

**Front-end:** HTML, CSS, and JavaScript for the user interface, enhanced by React for dynamic content rendering.

**Back-End:** Python for audio splitting and generating transcripts, Node.js for server-side logic and API development.

**Caching and Data Storage:** LocalStorage API is employed for client-side caching to store transient data, ensuring data continuity.

**Version Control:** Git was used for version control and collaborative development.

**Docker:** To containerize the web application for efficient deployment.

**Other Libraries and Frameworks:** Axios and Fetch API for save/discard, Pydub and WebRTC Voice Activity Detection (VAD) for audio chunking, react-router-dom for client-side routing and navigation, react-simple-keyboard for Telugu keyboard

# Scope of extension

- User-Friendly UI/UX
- Multi-Language Support
- Feedback Mechanism
- Integration with Storage Solutions (like AWS S3 or Google Cloud Storage)

# THANK YOU