IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI-POWERED REAL-TIME VIDEO CAPTION RECOMMENDATION SYSTEM

Prof. Sindhu K, Fouzal M, Vanitha A, B M Sushma, Varsha C

Assistant Professor, Student, Student, Student, Student
Department Of Artificial Intelligence and Machine Learning
Vijaya Vittala Institute of Technology, Bengaluru, India

Abstract: This project introduces an AI-powered real-time video captioning system that enhances accessibility and engagement by leveraging Automatic Speech Recognition (ASR) and Natural Language Processing (NLP) to improve caption accuracy, synchronization, and multi-language support. This system will benefit content creators, educators, and enterprises.

Keywords: ASR, NLP, Hugging Face Transformers, FastAPI, FFmpeg, PostgreSQL, Streamlit.

I. INTRODUCTION

Digital media has transformed communication and learning, making video accessibility crucial. Captions improve comprehension, but manual efforts are time-consuming and costly. AI advancements in ASR and NLP allow automation, yet existing systems struggle with accuracy, synchronization, and multilingual support. This project aims to address these challenges with an AI-powered solution.

II. OBJECTIVE

1. Enhance Accuracy: Improve real-time speech-to-text conversion with AI-driven ASR-NLP models.
2. Improve Readability: Use NLP techniques for better grammar and semantic coherence.
3. Ensure Synchronization: Precisely align captions with video content.
4. Support Multi-language Accessibility: Real-time caption generation in multiple languages.
5. Seamless Integration: Develop an easy-to-integrable system for various video streaming platforms.

III. LITERATURE SURVEY

Recent advancements in Artificial Intelligence, particularly in Automatic Speech Recognition (ASR) and Natural Language Processing (NLP), have significantly enhanced the capabilities of automated captioning systems. However, challenges remain in achieving high accuracy, real-time synchronization, and support for multiple languages.

[1] Sharma et al. (2025) integrated deep learning-based ASR models with NLP post-processing to enhance caption precision. Their work emphasized grammar correction and semantic alignment but did not address multilingual or platform-independent deployment challenges.

[2] Karad et al. (2024) proposed an AI-powered captioning system utilizing advanced ASR-NLP models to improve the accuracy of generated captions. Their approach significantly improved over traditional models but lacked real-time integration and multilingual scalability.

[3] Patel et al. (2024) introduced a Transformer-based ASR model that emphasized grammar and semantic structure improvements. Their results were promising for enhancing readability and coherence, but lacked integration with video streaming platforms for real-time application.

[4] Tanaka et al. (2023) developed a multilingual ASR-NLP captioning system capable of processing speech in various languages. While their system showed promise in linguistic flexibility, it struggled with synchronization and real-time processing under low-latency constraints.

[5] These existing systems highlight the growing interest in AI-driven captioning solutions but also underscore persistent limitations such as a lack of synchronization, inadequate multilingual support, and complexity in platform integration. This motivates the development of a comprehensive system that leverages state-of-the-art models like OpenAI Whisper and Hugging Face Transformers while ensuring real-time, synchronized, and accessible captioning.

IV. PROPOSED SYSTEM

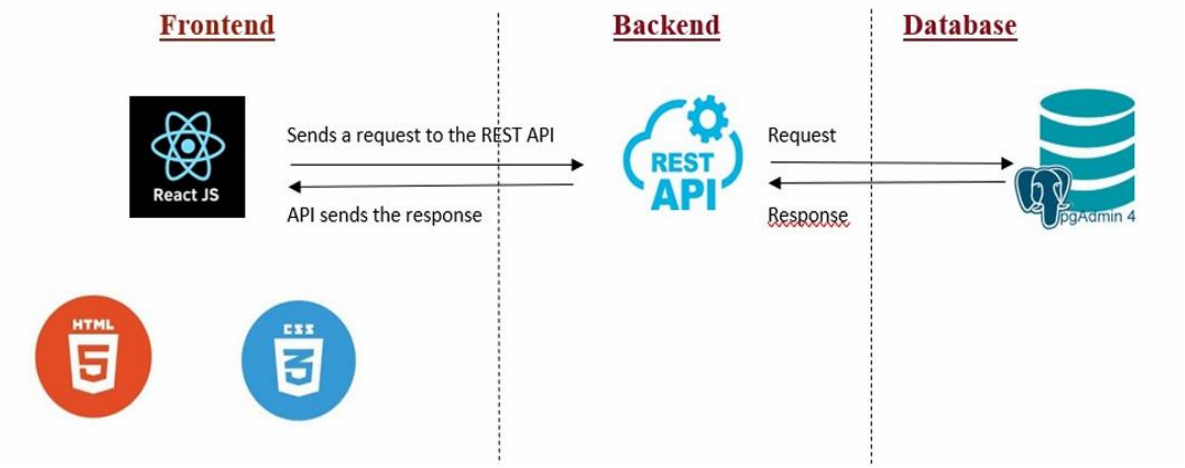

Fig.1: Proposed System

This system provides real-time caption generation using AI-based speech recognition and natural language processing. It integrates a user-friendly UI with back-end APIs, state-of-the-art ASR and NLP models, and multilingual support. The architecture is modular, scalable, and optimized for accessibility and automation.

A. User Login

The process begins with a secure login interface developed using Streamlit, where users can enter their credentials. FastAPI handles the authentication by validating these credentials against entries stored in a PostgreSQL database. On successful login, a session is initiated, and the user is directed to the dashboard for further interaction.

B. Uploading the Video

Once logged in, the user is provided with an intuitive dashboard to upload videos. This is handled by a form-based UI in Streamlit. The uploaded file is processed on the backend using FastAPI, which calls FFmpeg to extract audio from the video. Metadata like filename, duration, and format is stored in the PostgreSQL database.

C. Audio Processing and ASR Analysis

The extracted audio is sent to the Automatic Speech Recognition (ASR) engine powered by OpenAI's Whisper model. It converts spoken language into raw transcripts with timestamped segments. The model supports various languages and performs reliably even in noisy environments.

D. Natural Language Processing (NLP) Enhancement

To improve the raw transcript, NLP models from Hugging Face are used. This step ensures better readability by adding punctuation, correcting grammar, and improving sentence structure while preserving the original meaning of the speech.

E. Caption Synchronization
The cleaned transcript is aligned with the timestamped audio using segmentation logic. This results in captions formatted as .srt or .vtt files, which are compatible with most video players and platforms. Each caption chunk is carefully mapped to maintain readability and timing accuracy.

F. Metadata Storage in PostgreSQL
All essential data—including video info, transcripts, captions, languages, and timestamps—is stored in PostgreSQL. This makes it easy to retrieve or edit previous work, ensuring efficient session management and data reusability.

G. Caption Display and User Interaction
The final captions are displayed alongside the video on the Streamlit interface. Users can preview or approve the captions before export. This interactive layer ensures quality control and gives users flexibility over the final output.
H. Output and Export Options

After approval, the user can download the captions in formats like .txt. Additionally, the system allows embedding the captions directly into the video using FFmpeg and provides sharing or uploading options for different platforms.

V. SPECIFICATION REQUIREMENTS

A. Hardware Requirements
CPU: Intel i5, providing sufficient processing power for real-time video and audio tasks.
GPU: NVIDIA RTX 3060 or above, essential for accelerating AI model inference.
RAM: Minimum 16GB to ensure smooth multitasking and handling of large video/audio processing operations.
Storage: 512GB SSD, offering faster read/write speeds and efficient handling of media files and databases.
Camera: Real-time captioning tasks.
Internet: A high-speed internet connection is required for accessing cloud-based AI models and smooth streaming.
OS: Compatible with Windows 10 or Windows 11.

B. Programming and AI Stack
Programming Language: Python is the core language used, supporting a wide range of AI and video processing libraries.
Libraries/Frameworks: OpenCV is used for video frame handling and Whisper-based models.
AI Models: NLP enhancements are done using Hugging Face Transformers, ensuring quality in speech-to-text output.

C. Backend Development
Framework: FastAPI is used for building fast, scalable backend APIs that handle video uploads, audio extraction, and ASR model interaction.
Model Inference & Processing: All audio-to-text and NLP enhancement logic is encapsulated in FastAPI services, ensuring modular deployment and scalability.

D. Frontend Development
Framework: Streamlit is used for the user interface, allowing rapid prototyping and interactive UI for video upload, preview, caption editing, and download.
Integration: FastAPI endpoints are consumed from the front end using Python's requests or HTTP library for seamless communication.

E. Database and Storage
Database: PostgreSQL is the primary database used for storing user credentials, video metadata, transcripts, and caption history.

File Handling: Local storage or optional cloud services can be integrated for managing video and caption files for scalability.

VI. FLOW OF SYSTEM

The system operates through the following flow:
1. User Login: The User logs in via Streamlit UI; FastAPI authenticates credentials using PostgreSQL.
2. Video Upload: The User uploads a video, and FastAPI handles upload and uses FFmpeg to extract audio.
3. Audio Transcription: The audio is passed to OpenAI Whisper (ASR) to generate a raw, timestamped transcript.
4. Transcript Enhancement: NLP models clean the transcript by adding punctuation, grammar, and structure.
5. Caption Generation: Enhanced transcript is aligned with timestamps and converted to .srt/.vtt captions.
6. Data Storage: Video metadata, transcript, and captions are stored in PostgreSQL for future use.
7. Caption Display & Edit: Captions are previewed alongside the video on Streamlit UI.
8. Export: Captions can be downloaded or embedded into video using FFmpeg; video can be shared or published.

VII. RESULTS

1. Login Interface – Video Caption Generator

The login interface allows users to easily access the system by selecting their profile.

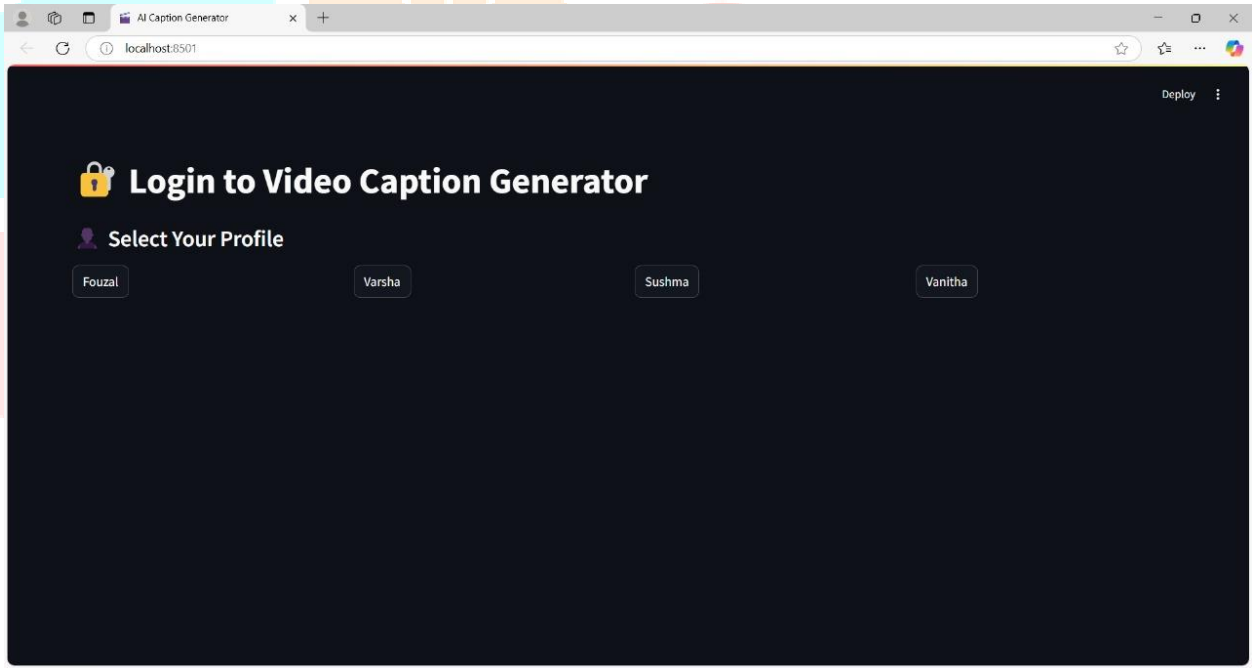

Fig.2: Login Interface

2. Secure Credential Login Interface

After selecting a profile, the user is prompted to log in using their email and password. This authentication step adds a layer of security and ensures that each session is user-specific, enabling personalized access to video processing and caption management features.

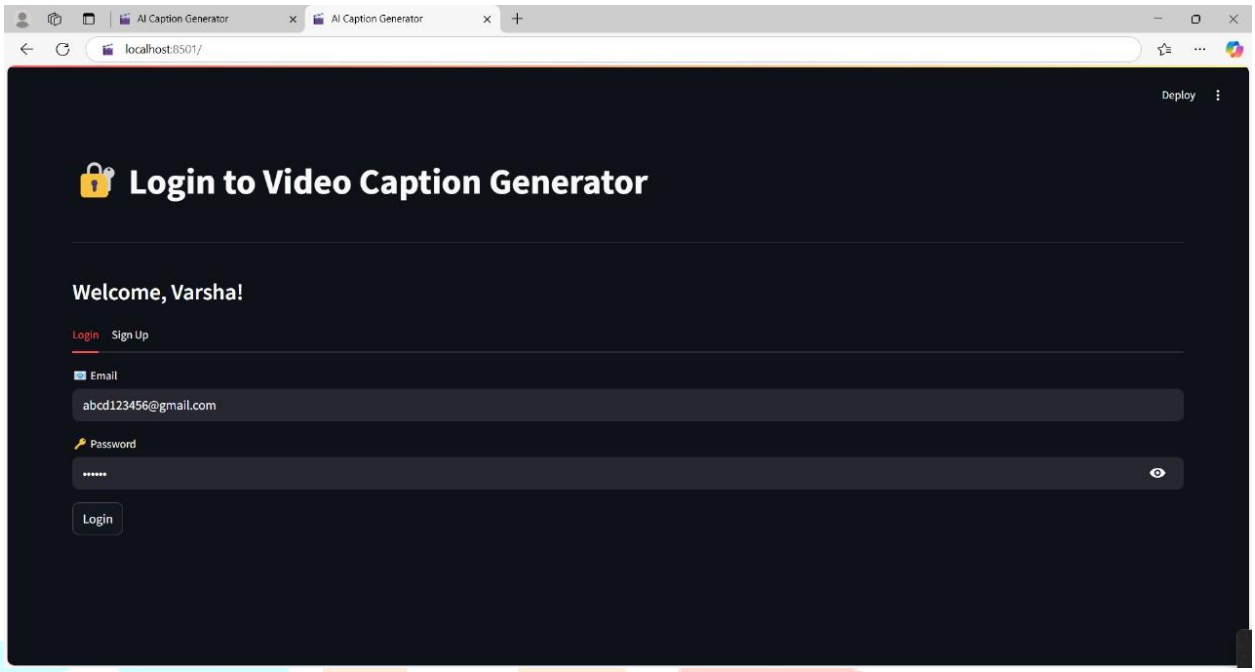

Fig.3: Secure Login

3. Video Caption Translation Interface

This interface allows users to upload a video, select a transcription language, and translate captions into languages like Kannada. It supports optional transcript uploads and processes videos up to 1GB in size.

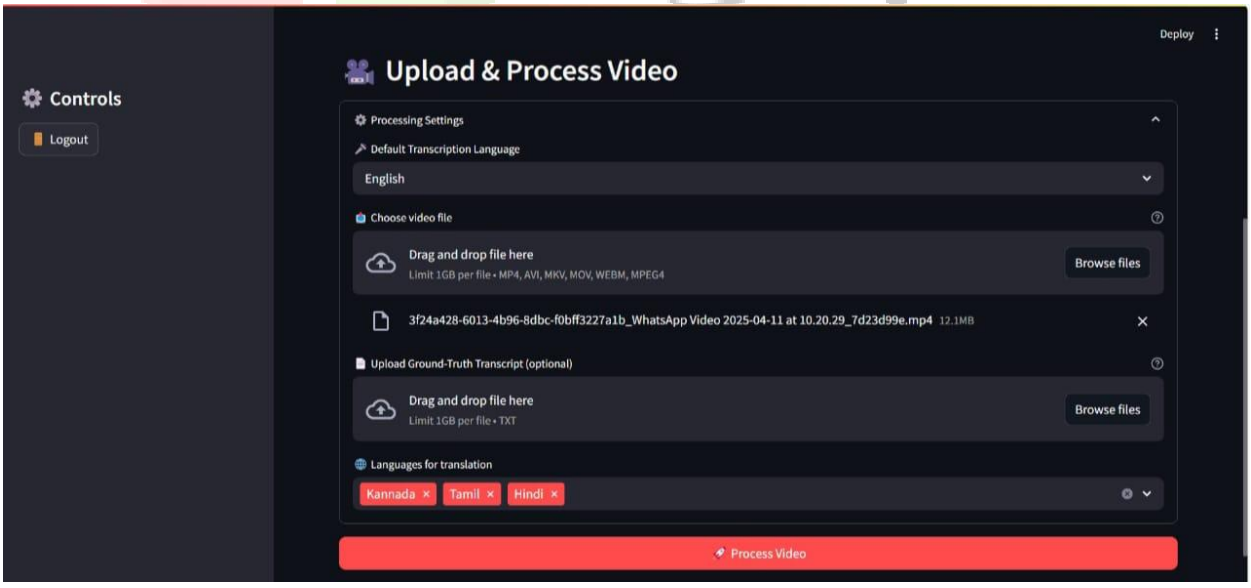

Fig.4: Multilingual Captions

4. Video Playback with Developer Menu Options

This screen shows a video preview within the AI Caption Generator, while the developer options menu is open in the top right corner. The menu allows actions such as rerunning the app, accessing settings, printing, recording a screencast, and clearing the cache. This interface supports both testing and development of the video captioning workflow.

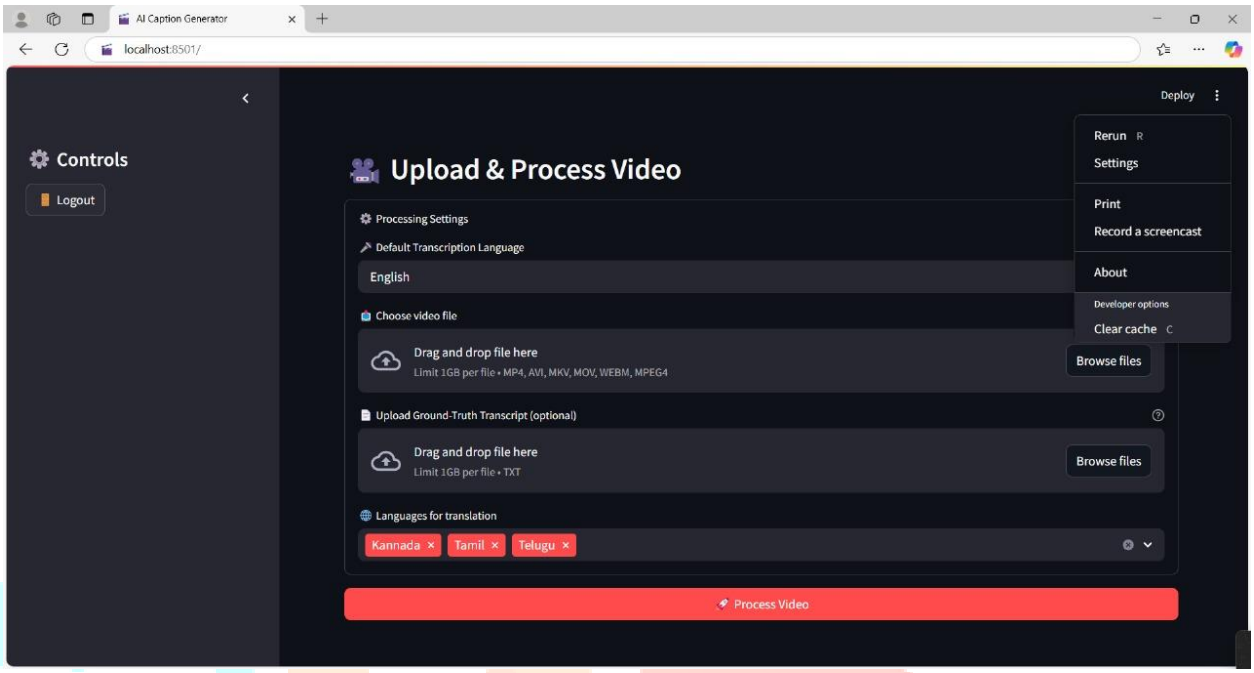

Fig.5: Advanced Video Player

5. Video Captioning Output with Transcript and Translation

This results page displays the output of the video captioning system. It shows the processed video with captions, the original English transcript, and a Kannada translation. A confidence score of 94.84% indicates the accuracy of the transcription. Users can download the captioned video, transcript, and translation directly from the interface.

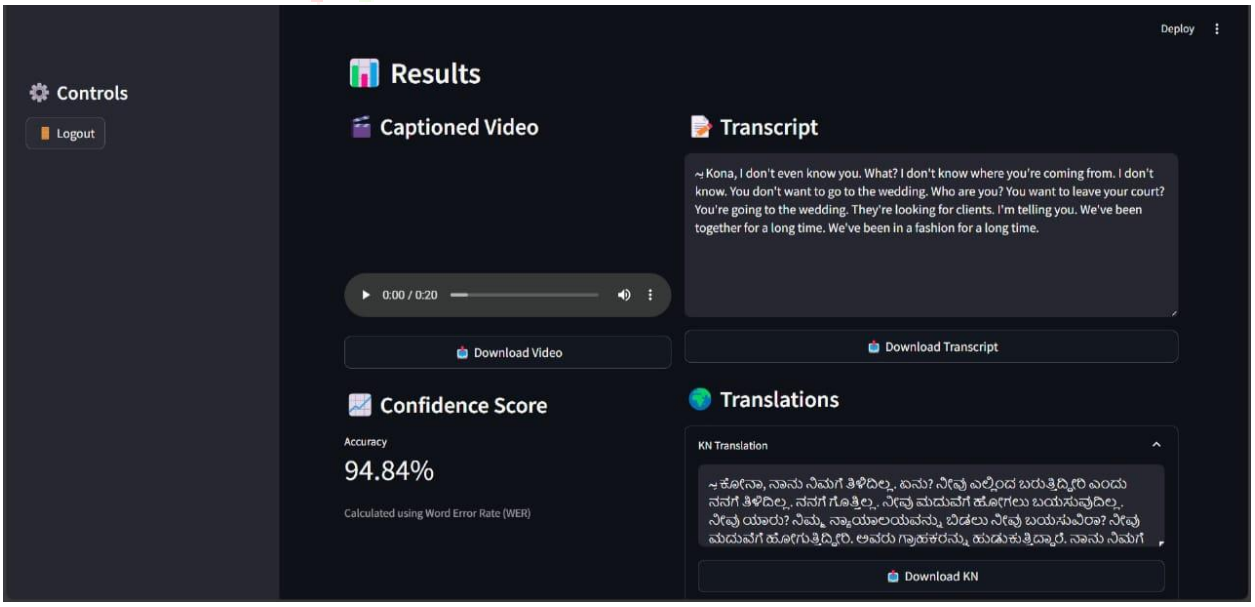

Fig.6: Transcript and translate

VIII. CONCLUSION AND FUTURE SCOPE

The proposed caption generation system improves accessibility by providing real-time captions for videos, aiding individuals with hearing impairments. It automates transcription, reducing manual effort and enhancing accuracy through advanced ASR and NLP technologies. Supporting multiple languages, it promotes inclusivity and easily integrates with various platforms, making it a versatile tool for content creators and businesses.

FUTURE SCOPE:

1. Real-time Translation: Extend the system to provide real-time translation of captions into different languages to reach a global audience.
2. Emotion and Sentiment Detection: Integrate sentiment analysis to reflect the speaker's tone or emotion, improving context in captions.
3. Voice Differentiation: Enhance the ASR to identify and label multiple speakers in the video.
4. Offline Support: Develop an offline version of the tool for use in areas with limited internet connectivity.
5. Mobile Platform Integration: Optimize and deploy the system for mobile platforms to expand accessibility and use cases.
6. AI Model Optimization: Continuously train and fine-tune models to handle domain-specific jargon (e.g., medical, legal) and improve performance under various audio conditions.

IX. REFERENCES

[1] Karad, S., Mehta, A., & Ramesh, V. (2024). AI-Powered Captioning using Advanced ASR-NLP Integration. International Journal of Artificial Intelligence Research, 12(1), 45–53.

[2] Sharma, R., & Kulkarni, P. (2025). Deep Learning-Based Speech Recognition and NLP Post-Processing for Video Captioning. IEEE Transactions on Multimedia, 27(4), 321–330.

[3] Tanaka, H., Yamada, S., & Suzuki, K. (2023). Multilingual Real-Time Captioning System Using Transformer-Based ASR and NLP. ACM Transactions on Accessible Computing, 15(2), 77–90.

[4] Patel, N., & Roy, S. (2024). Improving Semantic Coherence in Speech-to-Text Systems with Transformer Architectures. Journal of Computational Linguistics, 50(1), 23–36.

[5] OpenAI. (2022). Whisper: Open-Source Automatic Speech Recognition System

[6] Hugging Face. (2023). Transformers: State-of-the-Art Natural Language Processing for Pytorch and TensorFlow.

[7] FFmpeg Developers. (2023). FFmpeg: A Complete, Cross-Platform Solution to Record, Convert, and Stream Audio and Video.

[8] PostgreSQL Global Development Group. (2023). PostgreSQL Documentation.

[9] FastAPI Documentation. (2023). FastAPI – Modern, Fast (High-performance) Web Framework for Building APIs with Python.

[10] Streamlit Inc. (2023). Streamlit – The Fastest Way to Build and Share Data Apps.