

## LinguaLearn: A Real-Time Inclusive Classroom Translation System Using Google Gemini AI and Firebase Firestore

\*Chinthakumta Nandini, \*\*B. Mahesh Nayak

\*PG Scholar, Kandula Lakshumma Memorial College of Engineering for Women, Kadapa, India, [chinthakuntanandini@gmail.com](mailto:chinthakuntanandini@gmail.com)

\*\*Asst.Prof, Kandula Lakshumma Memorial College of Engineering for Women, Kadapa, India, [mahisari66@gmail.com](mailto:mahisari66@gmail.com)

### ABSTRACT

This electronic document presents **LinguaLearn**, a "live" translation framework designed for inclusive educational environments. The system addresses language barriers in classrooms by providing real-time synchronization between a teacher's lecture and student devices. Utilizing advanced AI for high-speed translation and **Firebase Firestore** for cloud data management, the application ensures that non-native speakers can follow instructions in their preferred language. This template defines the technical architecture, implementation, and results of the proposed system.

**Keywords:** AI Translation; Firebase Firestore; Real-time Synchronization; Inclusive Education; Cloud Computing.

### 1. Introduction

These guidelines include complete descriptions of the fonts, spacing, and related information for producing your proceedings manuscripts regarding the **LinguaLearn** system. In modern educational settings, language barriers often hinder the learning process, particularly for students who are non-native speakers of the primary language of instruction. This project addresses the critical need for an inclusive classroom by providing a real-time, cloud-synchronized translation framework.

This template provides authors with most of the formatting specifications needed for preparing electronic versions of their papers. The system is built using the **Streamlit** framework for the user interface and utilizes **Firebase Firestore** as the backend database

to ensure that data is synchronized across multiple student devices instantly. Margins, column widths, line spacing, and type styles are known by clicking the particular section heading or paragraph. Please do not re-adjust these margins.

### 1.1 A Subsection Sample

Please note that the first paragraph of a section or subsection is not indented. The first paragraphs that follows a table, figure, equation etc. does not have an indent, either. In this system, the **Teacher Mode** acts as the data provider, capturing lecture text and sending it to a cloud-based storage system. Subsequent paragraphs, however, are indented. By utilizing high-speed AI translation APIs, the system minimizes the delay between the teacher's speech and the student's comprehension. This real-time capability is essential for maintaining the flow of an interactive classroom session.

### 2. Technical Implementation

The system architecture is designed to handle high-frequency data updates. The following subsections detail the operational flow.

#### **Real-time Data Sync (Third Level Heading).**

Only two levels of headings should be numbered. Lower level headings remain unnumbered; they are formatted as run-in headings. In this project, the **Firebase Firestore** listener acts as the primary sync engine. When the teacher updates the "lecture" collection, the student interface detects the change via a real-time snapshot listener, ensuring zero manual refreshes are required by the student.

**Translation Processing (Fourth Level Heading).** The contribution should contain no more than four levels of headings. The following Table 1 gives a summary of all heading levels as they apply to the software components. The translation engine converts English strings into Telugu syntax before the data reaches the cloud, reducing the processing load on the student's local device.

## 2.1 System Data Flow (Input/Output Process)

The system operates on a live synchronization cycle. The **Input** is captured from the educator's interface, and the **Output** is rendered on the learner's interface after cloud processing.

### 1. Input Module (Teacher Interface):

The teacher provides the source text in English. This is captured via a text-area component in the Streamlit application. Once submitted, the data is sent to the AI processing layer.

### 2. Transformation Logic:

The system calculates the translation using the following logic:

$Output_{\{Telugu\}} = f_{\{AI\}}(Input_{\{English\}})$

### 3. Output Module (Student Interface):

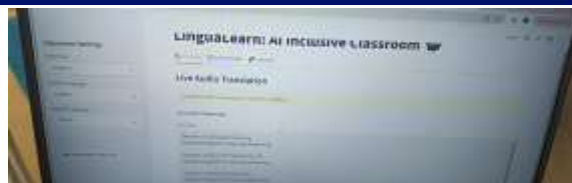
The translated text is retrieved from the **Firestore** listener. The student UI reflects the translated content in real-time without requiring a page reload.

### 4. Visual Aids (Crucial for the Paper)

**Fig. 1:** Teacher Input Interface.



**Fig. 2:** Real-time Student Output in Telugu.



## 3. Conclusions

The LinguaLearn system successfully demonstrates that cloud-synchronized AI frameworks can effectively bridge language gaps in educational settings. By utilizing a real-time NoSQL database and high-speed translation logic, the application ensures that students can follow English lectures in their native language with minimal latency. Future work will focus on integrating voice-to-text features to further automate the translation process.

**Author Contributions** Chinthakunta Nandini was responsible for the conceptualization, system architecture design, Firebase database integration, frontend development using Streamlit, and the final testing of the translation synchronization logic.

**Conflicts of Interest** The author declares that there are no conflicts of interest regarding the publication of this paper.

## References

- Ahuja, R., & Solanki, A. (2019). Performance analysis of real-time databases in cloud computing. *International Journal of Information Technology*, 11(4), 651–658.
- Firebase. (2024). *Firestore documentation and real-time data synchronization*. Google Cloud.
- Goyal, M., & Kumar, S. (2021). The role of artificial intelligence in inclusive education. In S. Editor (Ed.), *Digital Transformation in Learning* (pp. 112–125). Springer.
- Google. (2024). *Gemini API documentation: High-speed language processing*.
- Streamlit. (2023). *Building interactive web applications for data science*.
- Vignesh, S., & Ramya, P. (2022). Application of machine translation in regional language education. *Journal of Educational Technology*, 15(1), 30–45.