WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

VoiceLink: English–Hindi Translation and Emergency Assistance System

Nethra SS[1], Mehaa S[1], Nandhika Vanjinayaki B[1], Mohana Krishnan S[1], Dr. K. E. Kannammal[2]

[1]Department of Computer Science Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India
[2] Faculty Guide:

A B S T R A C T :

VoiceLink is a browser-based assistance application designed to support communication between English and Hindi speakers and to make selected emergency information easier to access. Based on the supplied project source, the application combines text translation, speech input through browser speech-recognition capabilities, a two-person conversation interface, saved translations, and a curated set of common phrases that can be accessed offline. It also provides region-based emergency-number information, quick access to selected numbers, and a location-selection flow that asks users to confirm suggested regions. The translation module checks a built-in phrase collection and saved translations before attempting online translation providers. The interface warns that machine translation can be inaccurate and directs users to contact local emergency services first in urgent situations. The authors report that Firebase authentication has been tested and is running in their current project deployment. The project focuses on combining everyday bilingual communication with quick access to emergency phrases and local contact information in a single responsive web interface.

Keywords: VoiceLink, English–Hindi translation, speech recognition, emergency assistance, offline phrases, Firebase Authentication, web application

Introduction

Communication barriers can make routine interactions difficult when people do not share a common language. The problem can become more urgent when a person needs to describe a basic need, contact family, or request help. VoiceLink is a browser-based project intended to bring bilingual translation and selected emergency information into one interface. The implementation focuses on English and Hindi, with text entry, speech input, a conversation view, reusable phrases, and region-specific emergency numbers.

The application is designed as an assistance tool rather than a replacement for professional interpreters or emergency services. Its interface explicitly cautions that machine translation may be inaccurate, particularly for medical or urgent wording. For emergencies, users are advised to call the local emergency number first.

Objectives

The main objectives of VoiceLink are to: (i) provide English-to-Hindi and Hindi-to-English translation; (ii) support speech input where the browser provides speech-recognition functionality; (iii) provide a conversation interface for two people; (iv) keep selected phrases and saved translations available on the device; (v) present emergency numbers for a user-selected region; and (vi) support account creation and sign-in through the configured authentication provider.

1. System Overview

VoiceLink is implemented using HTML, CSS, and JavaScript and runs in a web browser. The main page organizes the application into translation, conversation, emergency-number, and phrase views. JavaScript modules separate the user interface and application state from speech input, translation, authentication, and emergency-information handling. Phrase and emergency-number data are maintained in dedicated data files.

1.1. Translation workflow

The translation module follows a layered lookup approach. First, it checks a built-in phrase collection for the selected language direction. Next, it checks saved translations for the same language pair. If neither produces a result, it attempts live translation through online providers. The code identifies the Google Translate web endpoint as unofficial and undocumented and uses MyMemory as a secondary provider. Requests have a timeout, and the application is designed to report unavailable translation rather than presenting unchanged source text as a successful translation.

The application supports translation in both directions between English and Hindi. The interface includes a language-swap control, automatic translation as text is entered, a character limit, and controls to clear or save translations. The project also includes transliteration support for selected Hindi outputs when a provider supplies a suitable Latin-script pronunciation.

1.2. Speech input and conversation

The speech module uses browser speech-recognition functionality when available. Users can speak into the input area instead of typing, while the conversation view allows each participant to enter or speak a message in their own language and view the translated message for the other participant. Availability and accuracy of speech recognition depend on the browser, device, microphone permission, and network conditions.

1.3. Offline phrases and saved translations

A built-in phrase collection includes practical statements such as requesting help, asking for directions, communicating basic needs, and contacting family. These fixed phrases do not require a live translation request. Saved translations are retained on the current device, as described in the interface. Live translation, by contrast, requires network access and depends on the external provider being available.

1.4. Emergency assistance

The emergency view provides region selection, a quick-access bar, emergency-number listings, and phrases that may help a user communicate a need. The source data records the origin of emergency-number entries and a verification date. Region suggestions may use browser geolocation or the device time zone, but the application asks the user to confirm a suggestion rather than silently treating it as correct. If a location cannot be matched to a region with verified numbers, the interface directs the user to consult official local sources instead of guessing.

The supplied emergency data includes India's 112 emergency number and additional region-specific entries. Emergency numbers and local service availability can change; users should follow official guidance and not rely on this application as the sole source in a crisis.

Authentication and Data Handling

The authentication module is structured to support Firebase Authentication and includes flows for email-and-password sign-in, account creation, Google sign-in, password reset, and sign-out when the corresponding configuration and provider options are enabled. The project also contains a local demo authentication mode. The authors confirm that Firebase authentication has been tested successfully and is running in the current deployment. The deployed application uses environment-specific Firebase project settings, and authentication behavior depends on valid configuration and enabled providers.

The application also uses browser storage for locally saved phrases or translations and for selected region preferences. Users should understand that browser-local data is associated with the device/browser and may not automatically synchronize across devices. The translation module sends text to external online translation services when live translation is used; sensitive information should therefore not be entered unless the user is comfortable with that data flow.

Implementation

The project separates responsibilities into modules: the core module manages shared utilities and application behavior; the speech module handles voice input; the translator module performs phrase lookup, saved-result lookup, and online requests; authentication modules manage account screens and provider interaction; and the emergency module renders regional contact information and phrase assistance. Styling is separated into CSS files, while phrase and emergency datasets are maintained independently from the main HTML document.

Table 1 summarizes the principal features identified in the supplied source files.

Feature	Implementation in VoiceLink	Dependency / note
Translation	English ⇄ Hindi; phrase lookup, saved translations, online providers	Online providers require network access
Speech and conversation	Browser speech input and two-person translated conversation	Browser support and microphone permission
Offline support	Built-in emergency phrases and device-local saved translations	Live translation is unavailable offline
Emergency assistance	Region selector, quick-access numbers, emergency phrases	Confirm region and follow official guidance

Testing and Results

The authors report that Firebase authentication was tested and is running successfully in the current deployment. The project files also show explicit handling for authentication errors such as invalid credentials, duplicate email addresses, weak passwords, network failures, blocked pop-ups, and unauthorized domains. These are implemented error-handling paths; they should not be interpreted as proof that every possible case has been independently tested.

The source code supports a feature-based evaluation of the application's design: bidirectional translation, built-in phrase lookup, saved translations, speech input, two-person conversation, region selection, emergency-number display, and authentication flows. No formal performance measurements, controlled user study, accuracy benchmark, or quantified usability results were supplied with the project ZIP. Accordingly, this paper does not claim measured translation accuracy, response-time improvements, or statistically validated user outcomes.

Limitations

Live translation depends on external services and an internet connection. The Google Translate endpoint used in the code is explicitly marked by the project as unofficial and may change or become unavailable. Speech recognition depends on browser support and permissions. Machine translation can produce errors, and emergency numbers may differ by location or change over time. Firebase behavior in a deployed build depends on a valid project configuration and enabled authentication providers. These limitations should be considered when using or evaluating the application.

Conclusion and Future Work

VoiceLink brings English–Hindi translation, speech-assisted input, a two-person conversation interface, reusable offline phrases, saved translations, and region-based emergency information into a single browser application. Its modular structure separates translation, authentication, speech, and emergency assistance, making the project easier to maintain and extend. The authors report successful Firebase authentication testing in the running deployment. Future work could include systematic usability testing, translation-quality evaluation, accessibility testing across browsers, periodic review of official emergency-number sources, and more robust translation-provider integration.

Acknowledgements

The authors thank Dr. K. E. Kannammal, Faculty Guide, and Sri Shakthi Institute of Engineering and Technology, Coimbatore, for their guidance and academic support.

REFERENCES :

[1] Mozilla Developer Network (MDN). "Web Speech API." https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API (accessed 28 October 2026).

[2] Firebase. "Authenticate with Firebase using Password-Based Accounts on Websites." https://firebase.google.com/docs/auth/web/password-auth (accessed 28 October 2026).

[3] Government of India, Emergency Response Support System. "About ERSS / 112." https://112.gov.in/about (verification date recorded in the supplied project data: 21 October 2026).

[4] Federal Communications Commission. "The Nation's 911 System." https://www.fcc.gov/node/25869 (verification date recorded in the supplied project data: 21 October 2026).

[5] Government of the United Kingdom. "999 and 112: The UK's National Emergency Numbers." https://www.gov.uk/guidance/999-and-112-the-uks-national-emergency-numbers (verification date recorded in the supplied project data: 21 October 2026).