GlobeChat: Realtime Multilingual Messaging System

A real-time multilingual chat system supporting instant translation across users.

Name: Bhushan Datre
Department: Computer Science and Engineering
Academic Year: 2025

Abstract

GlobeChat is a real-time messaging system designed to eliminate language barriers in communication. It automatically translates messages into each user's preferred language, enabling seamless conversation between users who speak different languages. The platform uses Firebase for authentication and database management, Socket.IO for real-time communication, and the Google Cloud Translation API for multilingual support.

1 Introduction

Language differences frequently hinder effective communication in digital platforms. GlobeChat addresses this challenge by integrating real-time message translation directly into the chat workflow. Users communicate in their native languages, and messages are automatically translated on delivery. This significantly improves accessibility and enables meaningful cross-lingual communication.

2 System Architecture

The system follows a client-server real-time architecture.

- **Frontend:** Implemented using HTML, CSS (TailwindCSS), and JavaScript. The UI handles chat rendering, authentication interfaces, and user interactions.
- **Backend:** Built with Node.js and Express.js. It manages WebSocket connections, translation requests, and message routing.
- **Real-Time Messaging:** Socket.IO maintains persistent bidirectional communication between users and the server.
- **Database:** Firebase Firestore stores chat histories, user information, and metadata.
- **Authentication:** Firebase Authentication manages secure user login and registration.
- **Translation Service:** Google Cloud Translation API translates text messages into user-selected languages.

3 Workflow

1. A user signs in or registers using Firebase Authentication.

2. The user searches for another registered user and initiates a chat.
3. Messages are sent from the client to the server using Socket.IO.
4. The server stores the original message in Firestore.
5. For each receiving user, the server retrieves their preferred language and requests translation using Google Translate API.
6. Translated messages are broadcast back to all connected users in real-time.

4 Key Modules

4.1 User Authentication Module

Handles secure user sign-up, login, and session management using Firebase Authentication.

4.2 Chat Management Module

Responsible for creating chat documents in Firestore, tracking participants, and retrieving chat histories.

4.3 Messaging Module

Maintains real-time synchronization of chat messages across clients. Uses Socket.IO to deliver messages instantly.

4.4 Translation Module

Uses the Google Cloud Translation API to translate each message dynamically into the recipient's preferred language.

5 Results

The system successfully allows users speaking different languages to converse naturally. All messages are translated and displayed in real-time. The interface is responsive and simple to use, and the database structure allows chats to be retrieved efficiently.

6 Conclusion

GlobeChat demonstrates how multilingual real-time communication can be integrated into modern messaging applications. By combining translation services with a WebSocket-based messaging system, GlobeChat promotes inclusivity and global connectivity. Future enhancements include support for group chats, encryption, and mobile app deployment.

7 Screenshots of GlobeChat

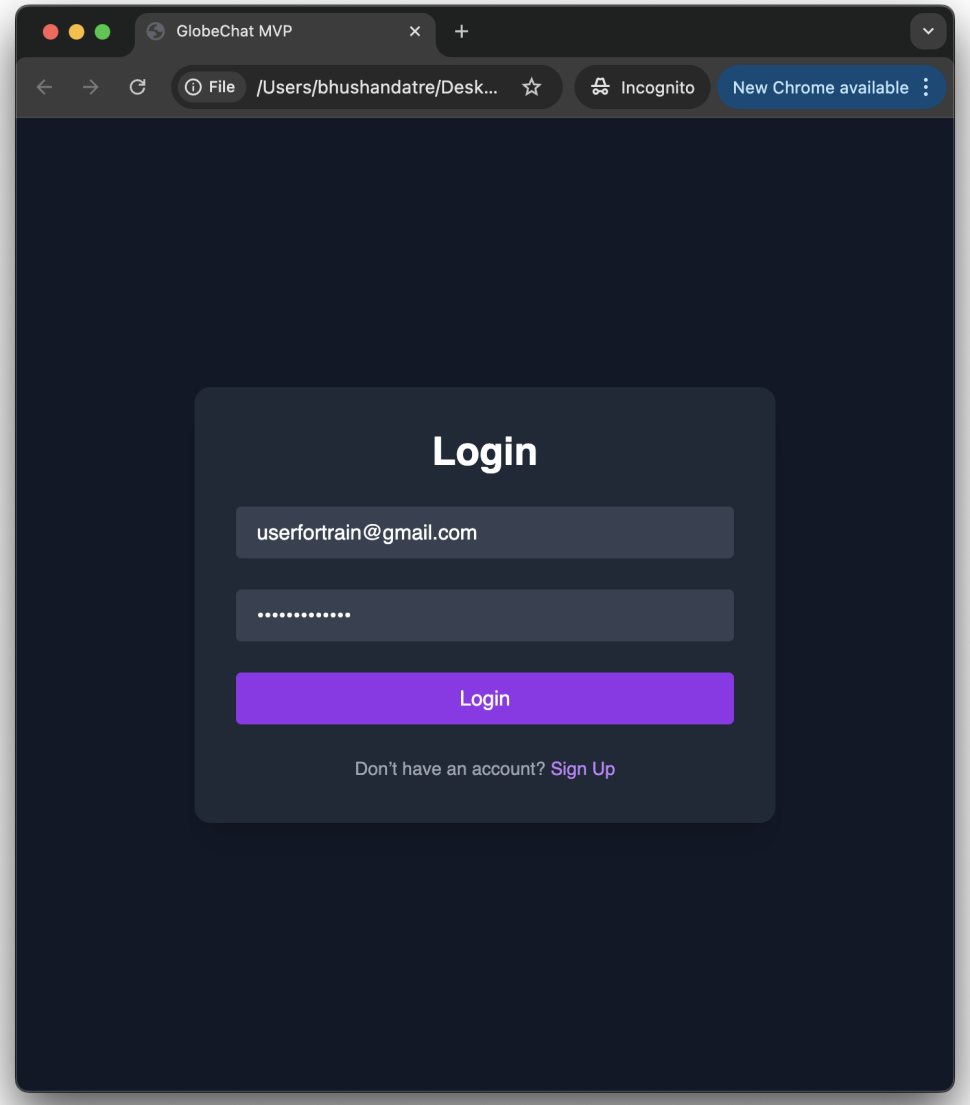

Figure 1: Login Screen of GlobeChat

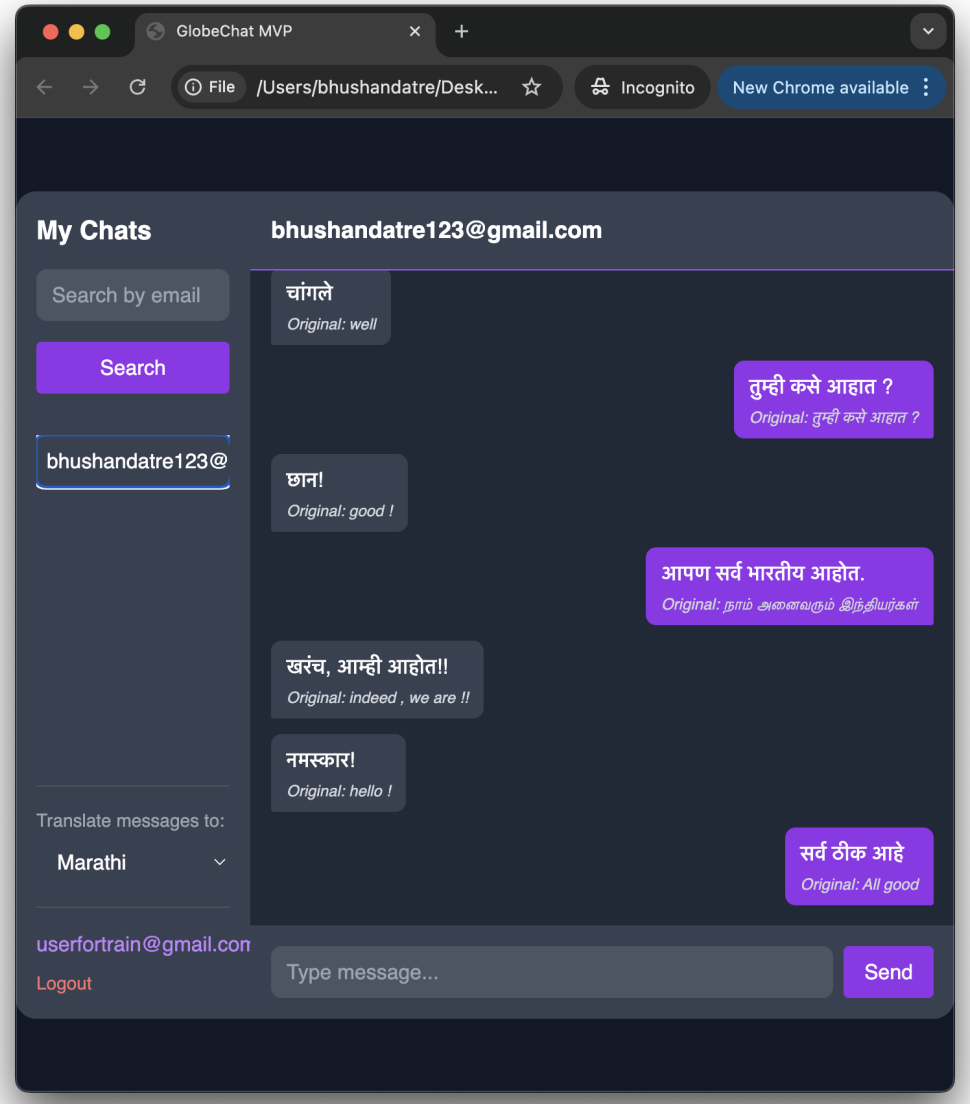

Figure 2: User A Screen of GlobeChat

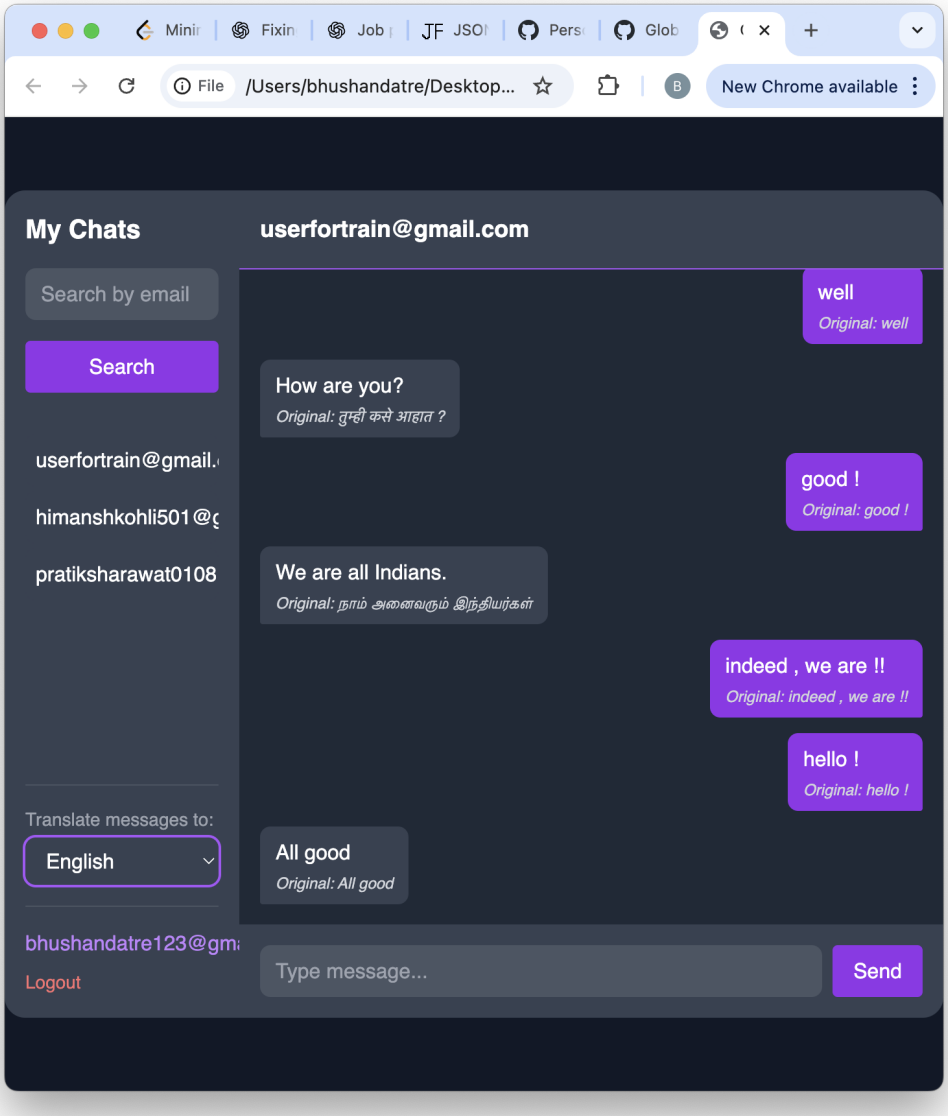

Figure 3: User B Screen of GlobeChat

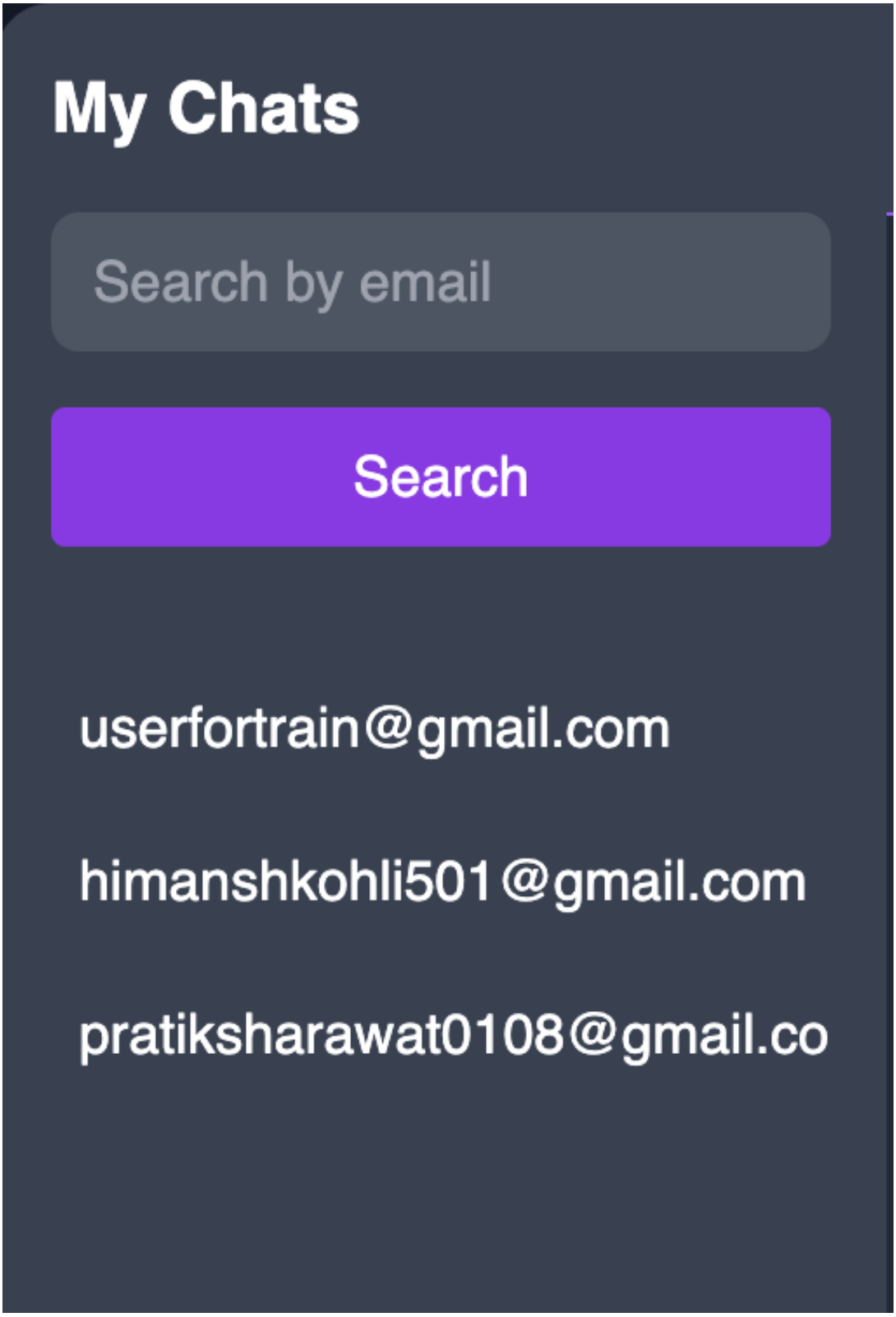

Figure 4: Searching , active chats interface of GlobeChat

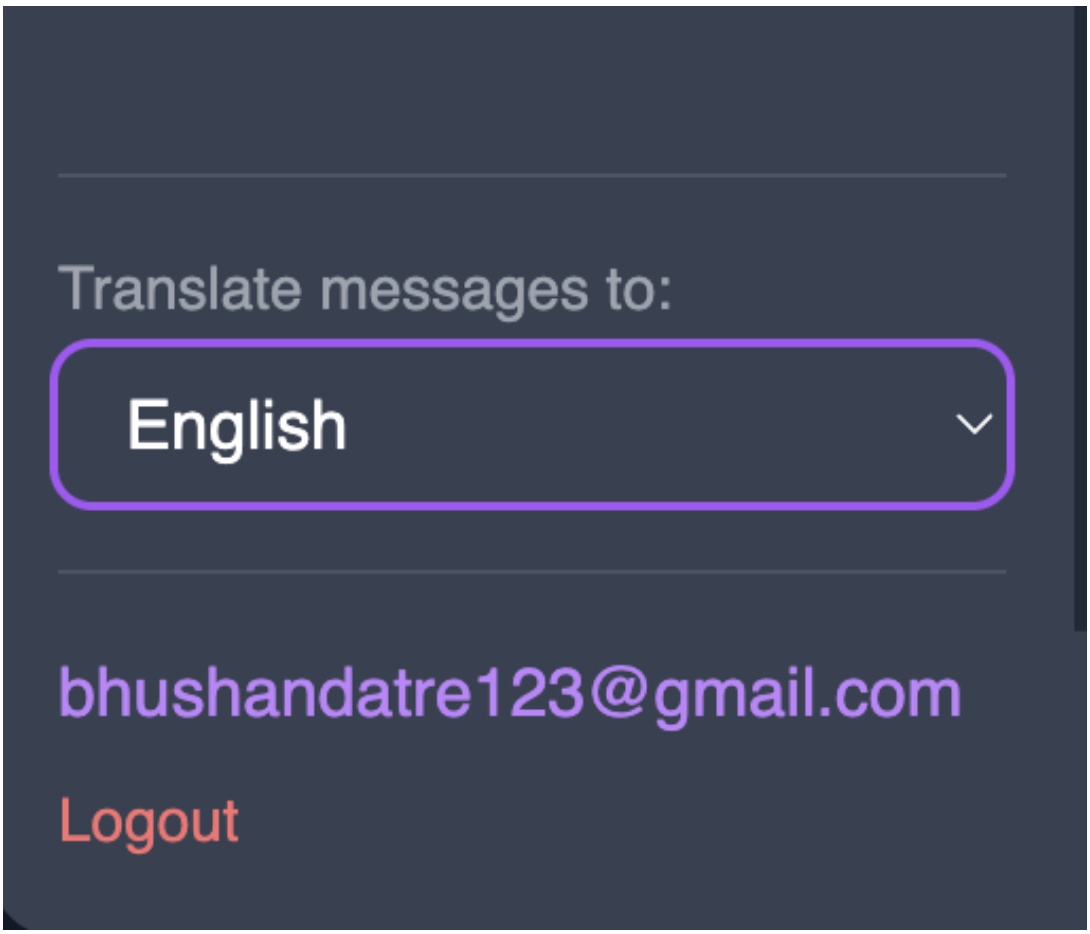

Figure 5: language selection interface of GlobeChat