

Real-Time Chat App with Language Translation

Requirements & Scope Document

Project Overview

The platform enables instant text communication between users online. Each message is translated into the recipient's preferred language via an external API. The system supports at least English, French, and any other language of choice at launch. (Optional: additional credit if built to easily add more languages).

Scope

In Scope

- Publicly accessible web chat interface
- One-to-one sessions for guest users (no account required)
- Persistent chat rooms for registered users
- Ephemeral guest sessions: messages deleted at session end
- Translation of incoming and outgoing messages
- User profile management (username, avatar, preferred language)
- Conversation history for registered users

Out of Scope

- Voice, video, or file sharing
 - Offline message queuing or delayed delivery
 - Group chat beyond two participants
 - Built-in speech recognition or synthesis
-

User Roles & Access

Registered Users

- Sign up with email/password
- View and retain full chat history
- Access multiple chat rooms or private conversations

Guest Users

- Enter a chat session without registering
- Automatically paired with another online guest or placed in a demo room
- Session ends when user closes browser or after inactivity
- All messages deleted immediately at session end

Administrators

- Activate/deactivate accounts
 - Moderate chats and view system logs
 - Configure supported languages and usage policies
-

Functional Requirements

Authentication & Onboarding

- Email/password signup and login for registered users
- Guest mode entry without credentials
- Optional single sign-on (Google, Microsoft, or email sign-in)

Real-Time Messaging

- WebSocket-based send/receive with <200 ms latency
- Typing indicators and presence status

Language Translation

- Integrate external translation API
- Translate messages on send and display both original and translated text

Data Retention

- Persist registered user chats in database
- Automatically purge guest session data at end of session

User Interface

- Responsive design for desktop and mobile browsers
 - Language selector and toggle between original/translated text
-

Non-Functional Requirements

- Scalability: Support 10+ concurrent sessions
- Security: HTTPS; hashed passwords; input validation

- Privacy: GDPR/PIPEDA compliance; clear guest privacy notice
 - Maintainability: Modular codebase; well-documented APIs
-

Technical Specifications

- Front-end: React or Vue.js; WebSocket library
 - Back-end: Node.js/Express or Python/FastAPI
 - Database: MongoDB or PostgreSQL
 - Translation API: Google Cloud Translate or Microsoft Translator
 - Deployment: Firebase, Glitch, or Replit (choose free, short-term tiers)
 - Version Control: GitHub repository with CI/CD pipeline
-

Deliverables

- Source code repository with README and setup guide
 - Final written paper (2-4 pages) covering design, implementation, testing, and future work
 - Video presentation (5–10 minutes) demonstrating features, architecture, and lessons learned
-

Evaluation Criteria

- Feature Completeness: All specified functions implemented
- Performance: Meets latency and concurrency targets
- Code Quality: Clean, modular, and well-documented
- UX Design: Intuitive, responsive interface
- Deliverable Quality: Clarity and depth of paper and video