Project Proposal (Mapped from Specification)

Sr.no	Roll.no	Names
1.	23L-0907	Hanzala Ahsan
2.	23L-0718	Muhammad Daniyal
3.	23L-2509	Syed Ali Naqvi
4.	23L-0643	Muhammad Auon Naseer
5.	23L-2528	Moiz

“Echo”

Local Encrypted Chat App
A secure, end-to-end encrypted chat application designed to work over local networks, supporting text, images, and audio with AI features.

Scope

This project falls under secure communication and collaboration tools. Unlike existing solutions (WhatsApp, Signal, Telegram) that depend on internet servers and cloud infrastructure, our system is LAN-only, ensuring that messages and media never leave the local network.

The scope includes:

- End-to-end encrypted communication: Secure transfer of text, images, and audio where only the intended client can decrypt.
- Offline-first messaging: Storage and later delivery of undelivered encrypted messages.
- Ai powered chat assistance with following features:
    - **Summarization** → Users can request concise overviews of long chats/conversations.
    - **Smart Replies** → Quick, context-aware reply suggestions for faster interaction.

    - **Translation** → Multilingual support enabling real-time message translation across users.
    - **Toxicity Detection** → Automatic identification and filtering of offensive/harmful text before delivery.
- Local Deployment: Fully containerized (Docker) setup ensuring ease of deployment, testing, and scaling within local networks.
- AI assistance will operate strictly on an **opt-in basis**. Only users who allow AI features will have their chat data processed by the models, ensuring maximum security and user control.
- Group Chat support: In addition to one-on-one chats, the system allows **secure encrypted group messaging** within the LAN. Groups benefit from the same end-to-end encryption, offline delivery, and AI-powered enhancements (e.g., summarization of group discussions, toxicity filtering).

Does not cover (at this stage):

- Cloud scalability.
- Mobile-native clients.
- Multi-device synchronization beyond LAN.

Future extensions could explore cloud deployment, mobile apps, and additional AI features (voice-to-text, text-to-speech).

Objectives

- Provide a **secure LAN-based chat app** with end-to-end encryption.
- Enable **offline message storage and later delivery**.
- Support **encrypted text, images, and audio** transfer.
- Ensure **server only stores ciphertext**; decryption remains client-side.
- Enhance user experience with **AI features** like summarization, smart replies, translation, and toxicity detection.
- Package the system into **Docker containers** for easy deployment and testing.
- Ensure AI features are **user-consent driven**, processing only data from participants who have opted in.
- Enable **encrypted group chats** that support collaborative communication while maintaining privacy and usability.

Problem Statement & Description

Most modern messaging applications require internet connectivity and store user data on external servers, which creates **privacy concerns**. In sensitive environments (e.g., offices, schools, local communities), users need a **private and LAN-only chat system** that guarantees no plaintext messages are ever accessible to the server.

This project addresses the problem by providing a **LAN-based, end-to-end encrypted chat system** where:

- Users' private messages and media remain inaccessible to the server.
- AI enhancements ensure **privacy-preserving intelligence**.
- A modular architecture allows future scaling to cloud or mobile without redesigning the core.

In short, it **reduces reliance on third-party servers**, ensures **secure communication in local settings**, and introduces **modern AI-powered convenience features** in a private environment.

AI Features

The project leverages state-of-the-art natural language processing (NLP) models and libraries to enhance the communication experience. The following features are included in the current scope:

- **Summarization** → Implemented using **Hugging Face BART** to generate concise summaries of long conversations.
- **Smart Reply** → Powered by **Hugging Face Flan-T5**, enabling context-aware quick replies.
- **Translation** → Facilitated by **Hugging Face MarianMT**, supporting multilingual chat functionality.
- **Toxicity Detection** → Handled through the **Detoxify** model to identify and filter offensive or harmful messages.
- **Deadline/Task Extraction** → Implemented using **Duckling** (for date/time recognition) in combination with **spaCy** (for entity extraction).

Future AI Enhancements (Not in initial release)

- **Voice-to-Text** → Planned integration with **Whisper** for converting speech to text.
- **Text-to-Speech** → Possible implementation via **gTTS** or **pyttsx3** for accessibility features. *(Note: pyttsx3 uses the system's engine rather than AI models.)*

Languages and Tools

- **Backend:** Java 17+, Spring Boot 3.x, PostgreSQL, Flyway, Maven
- **Frontend:** React (Vite), Tailwind CSS, Axios, WebSocket API
- **Encryption:** WebCrypto API (frontend), BouncyCastle/Java Crypto (backend)
- **AI:** Hugging Face models (BART, Flan-T5, MarianMT), Detoxify, spaCy, Duckling; optional Whisper + gTTS/pyttsx3
- **Containerization:** Docker & Docker Compose

- **Testing:** JUnit, Mockito, Jest, React Testing Library
- **Version Control & CI/CD:** GitHub, GitHub Actions