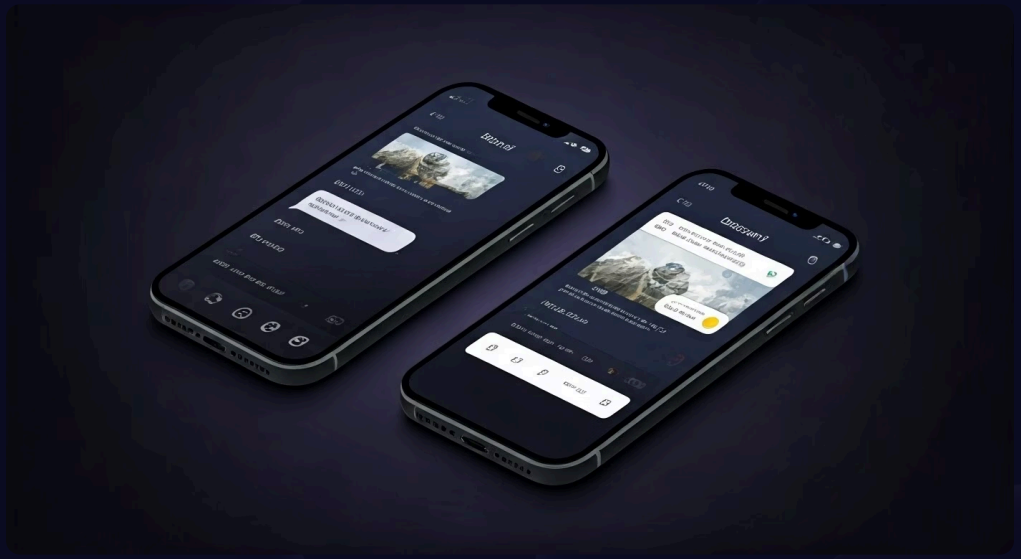
ChatApp – Mobile Messaging

ChatApp is a seamless cross-platform mobile application designed using React Native, Expo, and Firebase. It delivers a smooth, real-time messaging experience with support for sharing images and location data across both iPhone and Android devices. Perfectly crafted with a clean, modern UI to keep you connected anytime, anywhere.



Developed by: Olha Tsurenko

GitHub Repository: https://github.com/o-vilna/ChatApp

Project Overview

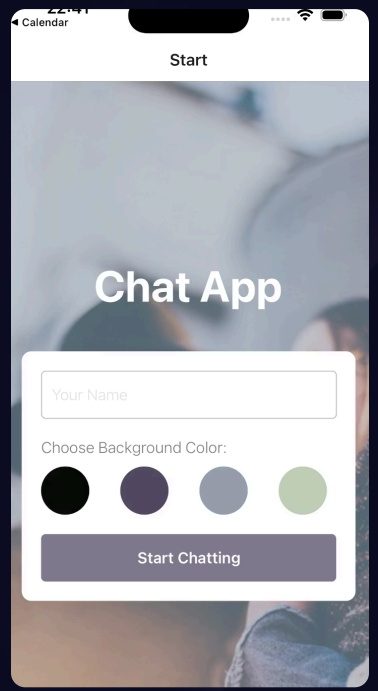
Objective:
Develop a cross-platform mobile chat application that enables users to send text messages, images, and share their geolocation seamlessly.

Tech Stack:

- **React Native** with **Expo** for app development
- **Google Firebase** (Authentication, Firestore, Storage) for backend services
- **Gifted Chat library** for chat UI components
- **AsyncStorage** for local data persistence

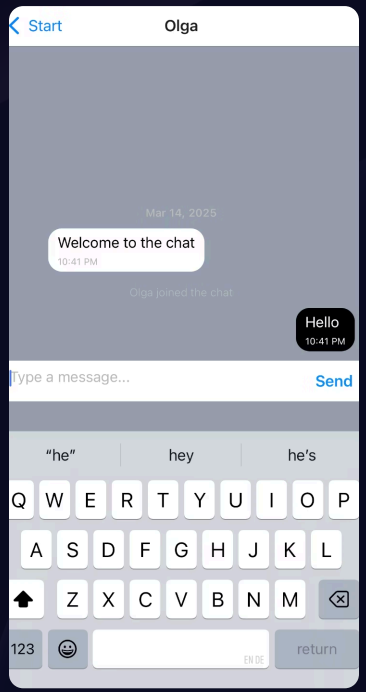
Target Devices:
iOS and Android platforms

Main Screen



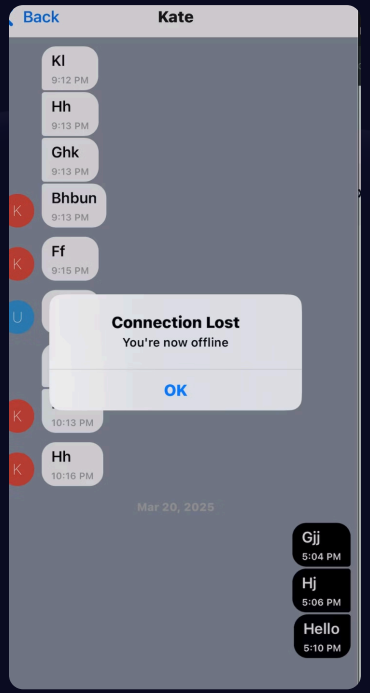
Intuitive and clean interface designed for effortless navigation and messaging.

Real-Time Messaging with Firestore



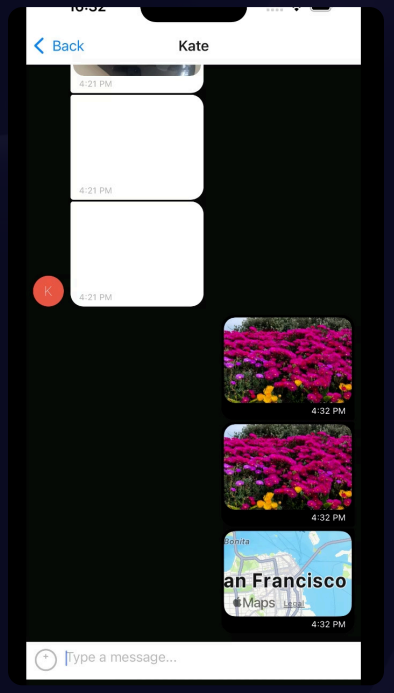
Instant message synchronization powered by Cloud Firestore for seamless conversations.

Offline Access



Access and read your message history even without an internet connection.

Image & Location Sharing



Share images and your location easily within chats.

The Problem

Modern users demand communication tools that are:

- Instant
- Accessible
- Offline-capable
- Feature-rich, supporting text, media, and location sharing

Developers face significant challenges building high-performance apps for both iOS and Android without managing separate codebases.

The Solution

- Native mobile app that works cross-platform (iOS/Android)
- Supports media and location sharing
- Stores messages both online and offline
- Uses an accessible and user-friendly design

User Stories

As a new user, I want to join a chat room easily.

As a user, I want to send text, images, and share my location.

As a user, I want to show where I am via map view.

I want to view messages offline.

I want to use the app with a screen reader.

Process

1 Research & Planning

Analyzed mobile UX patterns and screen reader accessibility guidelines.

Explored the Gifted Chat library to ensure alignment with project requirements.

2 Design & Prototyping

Adhered precisely to design specifications including fonts, spacing, and color schemes.

Developed a welcome screen featuring username input and a color picker.

3 Development

Implemented Firebase Authentication, Firestore database, and Cloud Storage.

Integrated expo-image-picker, expo-location, and AsyncStorage for enhanced functionality.

4 Testing & Iteration

Conducted testing on physical devices using Expo Go.

Resolved keyboard interaction issues and optimized the app layout for better usability.

Key Features

1 **Customizable Welcome Screen**

Allows users to enter their name and choose a color for personalization

2 **Convenient Chat**

Supports sending text, images, and location data

3 **Firestore Sync**

Provides instant updates with offline capability

4 **Camera & Image Upload**

Allows taking photos and uploading from the gallery

5 **Location Sharing**

Share your current location via map links

6 **Accessibility**

Fully optimized for users with special needs

Challenges & Decisions

Challenges:

- Handling async image uploads + chat message sync
- Platform-specific behavior (keyboard overlap, permissions)
- Offline functionality with real-time updates

Key Decisions:

- Stored images in Firebase Storage, not Firestore
- Cached messages locally for offline use
- Used **Gifted Chat** to streamline UI logic

Results & Takeaways

Successfully delivered a fully functional, cross-platform chat application.

Key learnings:

- Hands-on experience integrating React Native with Firebase
- Critical importance of accessibility and offline functionality
- Effective simplification of complex features like image uploads and maps

Next steps:

- Implement emoji reactions and push notification support