Project Proposal Team 12

Title: Troupe Chat

Project Overview:

Troupe Chat is a real-time chat application designed to help students and self-taught professionals connect through vibrant, interest-based communities. Whether you're passionate about anime, programming, literature, fitness, or entrepreneurship, Troupe Chat lets you join or create your own “troupe” a focused group where members can chat, share ideas, and build meaningful connections. The app blends instant messaging with community discovery, making it easy to find your people and stay engaged.

The core problem Troupe Chat solves is the lack of tailored, real-time spaces for learners and creators to connect around shared interests. Traditional platforms are either too broad or too static, making it hard to build genuine relationships or collaborate effectively. Troupe Chat fills that gap by offering customizable chat communities where users can engage in focused conversations, organize meetups, and collaborate on projects all in a space that reflects their passions.

What makes Troupe Chat valuable is its simplicity, flexibility, and user empowerment. Communities are created by users, for users with tools to manage roles, moderate discussions, and shape the culture of each troupe. It’s more than just messaging, it’s a platform for building networks, launching ideas, and growing together through shared interests. Whether you're a student looking for study partners or a self-taught developer seeking collaborators, Troupe Chat is your space to connect, contribute, and belong.

Project Scope:

What’s in?	What’s out?
User account creation and authentication	Video/voice calling features
Real-time messaging between users	Large file sharing (beyond text and small attachments)
Interest-Based Communities ("Troupes")	Push Notifications
Community Discovery	Advanced Moderation Tools
User Profiles	
Responsive UI	

App Features – Troupe Chat:

1. Users can create an account and log in

• Users register with email and password.

• Authentication ensures that only authorized users access the app.

• Requires: Database (users table) + authentication system

2. Users can send and receive real-time messages

• Users can chat 1-on-1 or within project groups.

• Messages appear instantly without page reload.

• Requires: Database (messages table) + real-time service

3. Users can create and join interest-based communities (“Troupes”)

• Communities are organized by subjects or courses.

• Users can become members to participate in discussions.

• Requires: Database (communities table + community membership table).

4. Users can discover and browse communities

• Search or explore communities based on their interests or subjects.

• See active communities and join them.

• Requires: Database queries for community listings + search functionality.

5. Users can manage their profiles

• Edit display name, profile picture, and personal info.

• View other users’ profiles within communities.

• Requires: Database (user profiles table), optional file storage for profile images.

6. **Users can create project-specific groups within communities**

• Form smaller groups for assignments or collaborative work.

• Groups have their own message threads separate from the main community.

• Requires: Database (groups table + group membership table).

7. **Responsive UI for all devices**

• Access Troupe Chat on desktops, tablets, and mobile devices.

• Layout adjusts automatically for different screen sizes.

• Requires: Frontend design (CSS/Blazor components)

Stretch Challenges:

1. **Users can create and manage Troupe**

• Users can create new communities and manage membership

• Admins of a Troupe can edit details or delete the Troupe.

• Requires: Database user authentication, role management (admin vs member).

2. **User can receive push notifications**

• Notify users of new messages, invites to communities, or updates within Troupes.

• Notifications can be in-app or browser-based.

3. **Progressive Web App (PWA) support**

• Users can install Troupe Chat on their device like a native app.

• Works offline or with intermittent connectivity (view cached messages or Troupes).

• Requires: Service workers, caching strategies.

Technical Considerations:

Data Storage: User accounts, messages, and chat rooms will be stored in a relational database (Supabase).

User Accounts: Yes, with authentication and password hashing using Supabase Auth.

External Services: Supabase as the backend and cloud storage.

Device Compatibility: Works on desktops, tablets, and mobile browsers via responsive design and as strecth challenge an PWA.

Basic Security: HTTPS, password hashing, session management, and role-based access.

Frameworks/Tech: .NET with Blazor, Render/Azure(as hosting).

Project Links:

GitHub: https://github.com/ElRodFe/CSE325-Team12-Project

Trello: https://trello.com/b/uYkML6lu/cse325-team12project