**Project Concept: A Real-time Chat Application**

**Overview:**

Let's create a real-time chat application that can handle concurrent users, ensuring seamless communication. We'll leverage the power of programming language C#, .Net Desktop App, ASP.Net Core MVC Rest API and WebSocket technology to achieve this.

**Technology Stack:**

Frontend: C# Win Forms

Backend: ASP.Net Core MVC Rest APIs and WebSocket

**Architecture:**

Frontend

User Interface: Develop a user-friendly interface with features a provision to connect to the chat window by giving the user’s name and email id and once in the chat room entering chat messages. User should be able to see his and others messages simultaneously.

Real-time

Updates: Utilize WebSocket technology to establish real-time communication between the client and server.

Backend

API Server: Create a REST API to handle user registration

Real-time

Server: Use WebSocket to manage real-time communication, broadcasting messages to connected clients, and handling user connections and disconnections.

**Benefits of this Approach:**

Real-time Communication: WebSocket enables real-time messaging and updates.

By combining these technologies, we can build a reliable, and feature-rich real-time chat application that can accommodate a number of users and deliver a seamless user experience.