Real-Time Chat Application

A Web-Based Instant Messaging System

Presented by Himanshu Shukla, Mohit Singh, and Pandey Vivek

B.Tech Computer Science Engineering

Introduction to Real-Time Communication

What is a Real-Time Chat Application?

A real-time chat application enables instant, bidirectional communication between users across the internet. Messages are transmitted and displayed immediately without page refreshes, creating a seamless conversational experience.

Why Real-Time Communication Matters

- Enables immediate response and collaboration
- Enhances user engagement and satisfaction
- Powers modern business communication
- Creates connected digital communities

Industry Leaders: WhatsApp, Facebook Messenger, Slack, Discord, and Microsoft Teams have revolutionised how billions communicate daily.

Problem Statement

Message Latency

Traditional HTTP-based systems suffer from significant delays in message transmission, frustrating users expecting instant responses.

Lack of Instant Communication

Without persistent connections, users experience disconnected conversations and must manually refresh to receive new messages.

Real-Time Interaction Gap

Modern users demand immediate, synchronous communication that mirrors face-to-face conversations—existing solutions often fall short.

Project Objectives

01

Fast Real-Time Chat Interface

Develop a responsive, intuitive interface that delivers messages instantly with minimal latency, ensuring smooth user experience across devices.

02

Multi-User Connectivity

Enable multiple users to join chat rooms simultaneously, supporting dynamic conversations with seamless connection management.

03

Instant Message Delivery

Implement WebSocket technology to push messages to all connected users immediately without polling or page refreshes.

04

Active User List Maintenance

Maintain and display real-time status of online users, with automatic updates when users join or leave chat rooms.

Technology Stack

Frontend Technologies

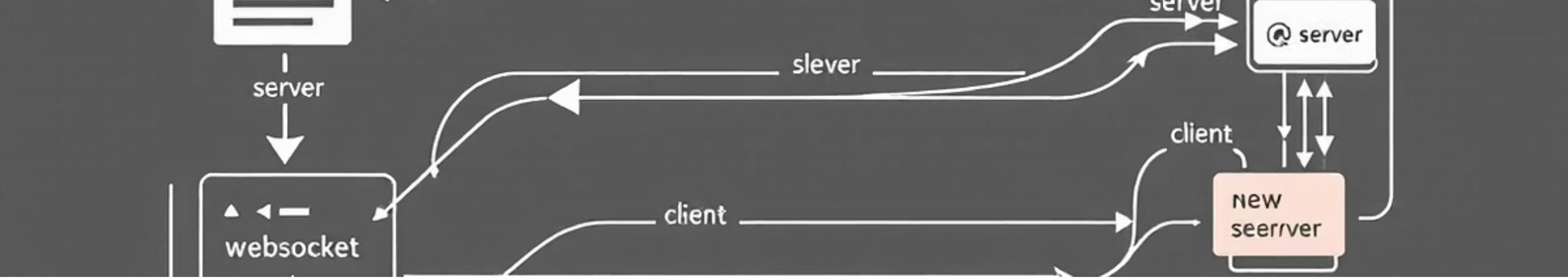
- **HTML5:** Semantic structure
- **CSS3:** Modern styling and animations
- **JavaScript:** Interactive functionality

Backend Technologies

- **Node.js:** Runtime environment
- **Express.js:** Web application framework
- **Socket.IO:** Real-time bidirectional engine

Development Tools

- **Git:** Version control
- **VS Code:** Integrated development environment



System Architecture

Client Browser

User interface running JavaScript that initiates WebSocket connections and handles message rendering.

Node.js Server

Express server managing HTTP requests, serving static files, and coordinating Socket.IO events.

Socket.IO Engine

Real-time communication layer enabling instant bidirectional data exchange via WebSocket protocol.

The architecture utilises persistent WebSocket connections for low-latency communication, falling back to polling when necessary. This ensures reliable real-time messaging across all network conditions.

Key Features

User Login & Room Selection

Users enter their name and choose a chat room, creating personalised, organised conversation spaces.

Real-Time Messaging

Messages broadcast instantly to all room participants with sub-second delivery times using WebSocket technology.

Join & Leave Notifications

Automatic system messages inform users when participants join or exit, maintaining conversation context.

Online Users List

Dynamic sidebar displays currently active participants in the room, updating in real-time.

Chat History (Optional)

Persistent message storage allows users to view previous conversations and maintain continuity.

How It Works: Message Flow

User Joins Room

User enters username and selects chat room from the login interface.

Socket Connection Established

Client initiates WebSocket connection to server via Socket.IO, creating persistent communication channel.

Message Sent to Server

User types message and submits; client emits event with message data to Node.js server.

Server Broadcasts Message

Server receives message and broadcasts it to all connected clients in the same room.

Instant Display

All users receive message simultaneously and see it appear in their chat interface without delay.

Thank You

Questions & Discussion

We appreciate your attention and welcome any questions about our Real-Time Chat Application project.

Project Team: Himanshu Shukla • Mohit Singh • Pandey Vivek

B.Tech Computer Science Engineering