# Complete WebSocket (Socket.IO) Implementation Notes

## 1. Introduction

This document summarizes all key concepts, use-cases, and flow discussions related to integrating WebSockets (using Socket.IO) in a MERN stack project. The focus is on real-time data flow between frontend (React) and backend (Node.js/Express), including online status, notifications, and task updates.

## 2. Core Concepts

• socket.emit: Used to send data from client to server or server to client.

• socket.on: Used to listen for events on the client or server.

• io.emit: Used on the server to broadcast to all connected clients.

• socket.broadcast.emit: Sends data to all clients except the sender.

• socket.on('disconnect'): Triggered when user disconnects.

## 3. Real-Time Join Flow

✔️ User accepts project invite and gets redirected to frontend.

✔️ Frontend emits: socket.emit('new\_member\_joined', { username, projectId })

✔️ Server receives and stores in memory: onlineUsers[projectId]

✔️ Server emits to others: io.emit('show\_notification', { message })

✔️ Frontend listens via socket.on('show\_notification') and shows popup/alert.

## 4. Real-Time Use Cases

• New member joined → show live notification

• Live online/offline status → using isActive flag

• Typing indicator → socket.emit('typing'), socket.on('typing')

• Task activity updates → socket.emit('task\_update') + UI broadcast

• Private or project chat using WebSocket

• Project-wide presence tracking using socket.id

## 5. Online/Offline Status

• On user join → add socket.id and username to memory (isActive: true)

• On disconnect → update isActive to false and broadcast update

• Frontend shows green/red dot based on isActive flag

## 6. Example Events

• socket.emit('join\_project', { username: 'Ravi', isActive: true })

• socket.on('new\_member\_joined', callback) → add to memory

• io.emit('show\_notification', { message: 'Ravi joined the project 🎉' })

• socket.on('disconnect', () => { update status to offline })

## 7. Summary

WebSockets allow real-time, two-way communication between the client and server. In a project management app, this enables live notifications, status updates, task tracking, and better collaboration.