

# Implementation of Real-Time Text Messaging using socket.io

Abhishek Kumar, Ankita, Ankit Kumar

Computer Science Department, Presidency University, Bangalore, India

---

## Abstract

We show how real-time text may be smoothly incorporated into instant messaging systems using new standardized methods. Text that is communicated in real-time while it is being written or generated is known as real-time text. The recipient may read the sender's text right away, without having to wait. Despite the benefits of instant messaging (IM), such as its simplicity and speed, IM cannot be compared to face-to-face interaction. In terms of temporal organization and information dimension levels, the existing message-by-message interface falls short of genuine dialogue. Our research looked towards using a real-time text-display interface to close this gap. To evaluate and assess the duration of quiet (brief pause while no one types), comparative user research was done using three types of IM interfaces.

*Keywords:* Web-based Applications, JavaScript, Web Framework, Socket.io, ExpressJs, Mongoose, MongoDB, NodeJs, BcryptJs, JWT, Redux, ReactJs

---

## Introduction

As time passes, information and communication technology continues to advance at a breakneck pace, playing an increasingly essential role in society. We can enable individuals to get information and communicate with one another from anywhere, at any time, and at a quicker rate than before thanks to information and communication technology. If we look at the current situation after the technology has arrived, most people no longer use newspapers to get information, and even fewer people use news channels to get information these days. Instead, most people use smart technology to receive any type of information via smartphones or other devices, and if we talk about chat applications, most people use chat applications to receive information. Socket.IO is a real-time JavaScript library. We favour it since the applications are essentially platform-agnostic, meaning they run well in all browsers. Another reason is that Node.js comes with a fantastic, lightweight JavaScript library for doing back-end tasks. Socket.IO bills itself as the "fastest and most dependable real-time engine" on the market. It is quite easy to set up. Let's begin by including Socket.IO into your project. There is just one way to install it right now – via npm (node package manager). To do so, open a terminal window and type the command below (you need to have Node.js installed on your machine)

## Frameworks and Technologies Used

In this section, we will refer to the technologies used for this project. These are the modern technologies commonly used around the globe to develop high-level web applications. Below are the technologies that power this project.

- **Socket.IO:**

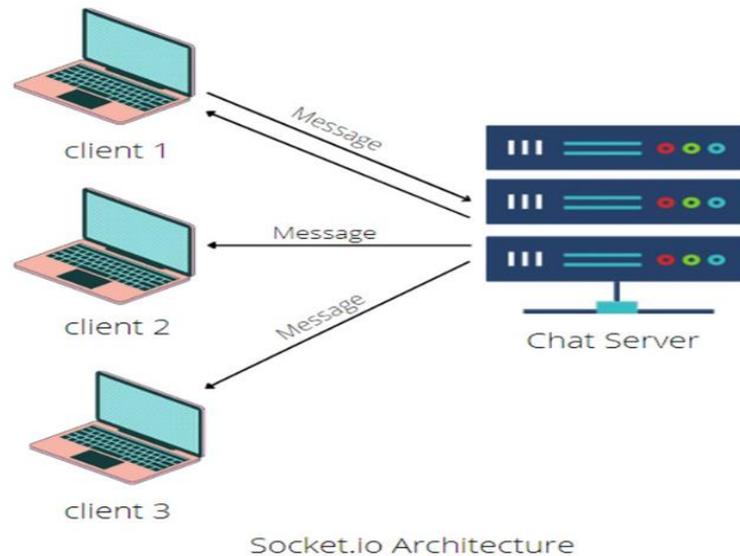
Socket.IO is a real-time web application library based on an event-driven JavaScript language. It is divided into two parts: a browser-based client library and a Node.js server library. The APIs are substantially the same in both components. While offering the same interface as Socket.IO, it primarily employs the Web Socket protocol with polling as a backup option. Although it may be used as a simple wrapper for Web Sockets, it has a lot more functionality, such as broadcasting to numerous sockets, storing data for each client, and asynchronous I/O. The Node Package Manager is required to install it.

### What can socket.io do?

Client-server communication is possible with Socket.IO. When a client has Socket.IO installed in their browser and a server has the Socket.IO package installed, bi-directional communication is enabled. Data may be delivered in a variety of formats, but JSON is the most straightforward.

### What web socket.io is not?

Socket.IO is NOT a WebSocket implementation. WebSockets aren't supported by Socket.IO. When feasible, Socket.IO utilizes WebSocket as a transport, but it also adds extra metadata to each message. That's why a WebSocket client won't be able to connect to a Socket.IO server, and vice versa.



- **NodeJS :**

Node.js (or Node) is a framework or a library, as well as is a runtime environment<sup>9</sup>. This framework can be used to develop JavaScript programs that run outside the browser. The Node.js framework includes an API to work with the filesystem, access databases, listen to HTTP requests, and more.

- **ReactJS :**

Facebook's open-source toolkit for creating user interfaces is called React<sup>8</sup>. It's non-intrusive and works with any library or framework. React generates its own virtual DOM object, reducing access to the browser's DOM and improving speed. React introduces the JSX format, a JavaScript syntactic extension that resembles XML, for content rendering. It's best to use JSX, although it's not required..

- **Redux :**

For JS Apps, a Predictable State Container. It aids in the development of applications that act consistently, runs in a variety of settings (client, server, and native), and are simple to test.

The Redux DevTools makes it simple to see when, when, why, and how the state of an application changed. The design of Redux allows you to track changes, employ "time-travel debugging," and even send full error reports to a server. Redux can be used with any UI layer and has a wide ecosystem of extensions to meet your specific requirements.

- **Material UI**

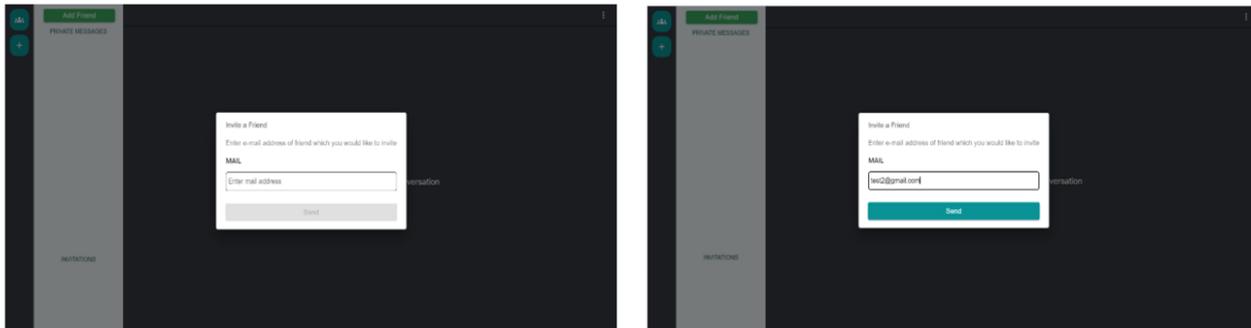
Material Design, a new suite of user interface components from Google, may become a viable alternative to Bootstrap. Material Design is a cross-device design that includes a collection of nice-looking UI components<sup>7</sup>. Material Design is a cohesive framework for creating digital experiences that integrates philosophy, resources, and tools.

### SYSTEM DESIGN AND IMPLEMENTATION

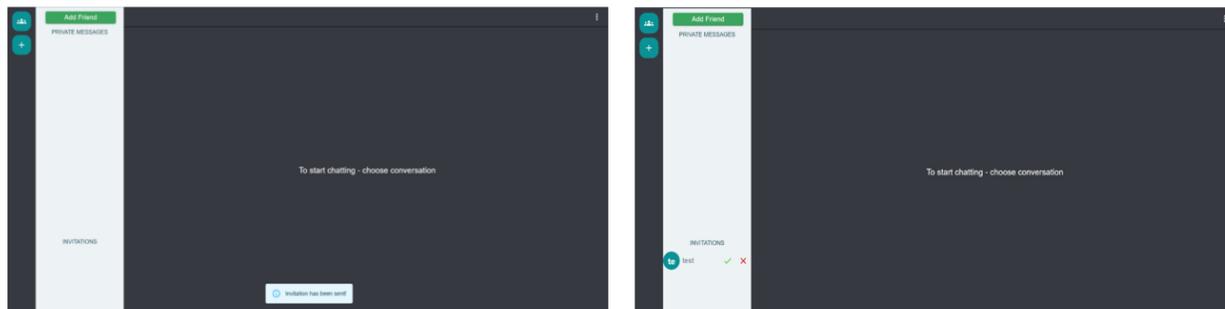
After logging in, the user may search for other users who have already registered on that platform and connect with them by sending an invitation request, which the invited user can accept or deny. When a user accepts the invitation, they can work together in various rooms. Users can connect with one

another through real-time text messaging after the cooperation is complete.

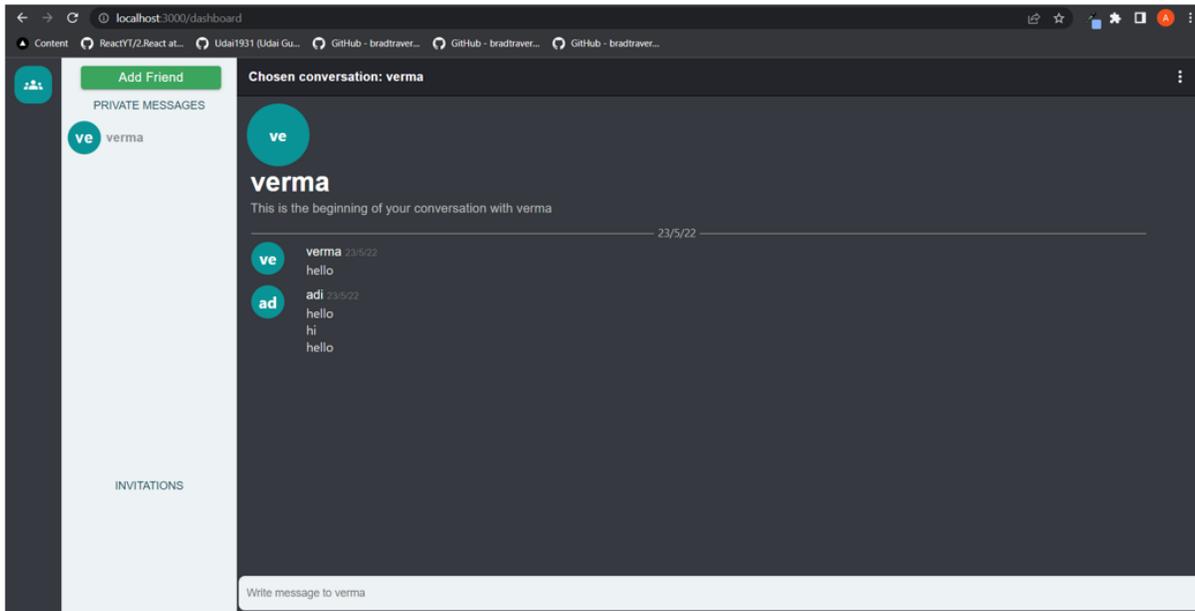
#### FRIEND INVITATION



#### ACCEPTING FRIEND INVITATION



## Chat Section:



## CONCLUSION

Making a "Real-Time Web Application," as opposed to a website about a product or a portfolio, has proven to be rather exciting and novel. As a result, I can say that socket.io made creating and developing our own chat application a breeze. I can honestly say that this endeavor taught me a lot more than I expected. In today's world, chat applications are in high demand and needed everywhere, and they are saving us a lot of time and money. As I previously stated, with the help of a chat application, we can send the same message to a large number of users at once, and we don't have to call everyone for every query or problem. Finally, I'd like to point out that developing a chat application with Node.js and Socket.io is ideal for two reasons: the first is that it saves time by allowing you to text in a matter of seconds, and the second is that you will learn a lot while developing it, ensuring that you have a firm grasp on the technology.

## ACKNOWLEDGMENT

This work was supported by the Department of Computer Science, School of Engineering, Presidency University, Bangalore under the guidance of Mr. Nasurudeen Ahamed Assistant professor CSE department, School of Engineering, Presidency University.

## REFERENCES

- Zinah Nayyef Sarah Amer and Hussain "Peer to Peer Multimedia Real-Time Communication System based on WebRTC Technology" International Journal for the History of Engineering & Technology vol. 2 no. 9 pp. 125-130 2019
  - Akshay Kashyap, Graduate Student Member, IEEE, and Benny Bing, Senior Member, IEEE "Efficient HD Video Streaming Over the Internet"2019
  - Alam Rahmatulloh, Irfan Darmawan, Irfan Darmawan,IEEE "Performance Analysis of Data Transmission onWebSocket for Real-time Communication"2019
  - <https://www.mongodb.com/docs/>
  - <https://reactjs.org/docs/getting-started.html>
  - <https://webrtc.org/>
  - <https://socket.io/docs/v4/>
- <https://redux.js.org/>
- <https://medium.com/swlh/real-time-chat-application-using-socket-io-in-node-js-37806e98918c>