# FULL STACK PROJECT

## (2021-22)

Real-Time Chat System

SYNOPSIS



# Institute of Engineering & Technology

## Team Members

## Aryan Garg (181500135)

## Aryan Saxena (181500136)

## Ishika Dubey (181500283)

**Ashita Vijay Seth (181500145)**

**Supervised By:**

## Mr. Pankaj Kapoor

**Department of Computer Engineering & Applications**

**Title**

The main purpose of this project is to provide a real time service to the user to chat with each other.

## Introduction &Problem Statement

Real-time chat system using socket.io and Node Js. Our system stores chats in the database. We have used Mongo DB with Mongoose. For Front End we are going to use React, HTML, CSS, Bootstrap and JavaScript, while for backend we will use NodeJS, Mongoose and Socket.io web socket Library.

Our chat system will work in real-time. So we don't need to refresh the page to get new data.(SPA)

We have used Socket.io for our Web Sockets connection.

## Objective

The Objective of this project is to develop a front-end part of a chat website. It will allow the user to chat with each others in real time. Our main objective is developing a simple chat application; therefore for simplicity we have avoided the authentication part. The messages must be updated without refreshing the page.

## Methodology

To build a real time chat application, there should be a real-time system for sending-receiving data. This can be attained using web-sockets. Instead of using the web-sockets API directly we will be using socket.io. We will also be using node.js so as to simplify building the web-app.

To develop a real chat application, we have to put in place a real-time system of sending/receiving data. It will be impossible to do it with a relational database and Ajax calls. So we will be using WebSocket and the library socket.io.

## Implementation Hardware/Software Interface

This section lists the minimum hardware and software requirements needed to run the system efficiently

## Hardware Interface

* Pentium Processor
* Hard disk: 1 TB
  + 4 GB RAM

## Software Interface:

* + - Operating System: Windows 10.
    - Web Browser: Mozilla FF 31 or Google Chrome.
    - Tools Used: Html, CSS, JavaScript, Bootswatch, NodeJs,ExpressJs,MongoDB.
    - User Interface Design: Visual Studio (Version 1.48).

## Motivation

Messaging apps are surging in popularity nowadays, there are literally thousands of chat based apps and they are also very popular among the users as they provide a personal touch while connecting the world together. This is the most basic chat app we can create. It is small and light weight but effective. You can add your functionalities in the app with a better UI and you are good to go.

## Online Git Repository

## <https://github.com/gargaryan/CHAT-MATE/upload/main>

## Conclusion

So, in this web application we have built a real time chat system in which user has enter his/her name first in order to access the webpage. One can chat with the other for however long as he want. It is a real time web application so user does not have to bother about refreshing the page again and again.

## References

* + - [https://www.w3schools.com](https://www.w3schools.com/)
    - [https://getbootstrap.com](https://getbootstrap.com/)
    - [https://www.javascript.com](https://www.javascript.com/)
    - [https://www.beta-labs.in](https://www.beta-labs.in/)
    - <https://developer.mozilla.org/en-US>