



**NAME OF GROUP MEMBERS:**

1. STUDENT NAME: **MEHDI RAZA** STUDENT ID: **62589**
2. STUDENT NAME: **BILAL KHAN** STUDENT ID: **62708**

**NAME OF UNIVERSITY:**

**KARACHI INSTITUTE OF ECONOMICS AND TECHNOLOGY**

**COURSE NAME:**

**NETWORK PROGRAMMING**

**DEPARTMENT:**

**BS COMPUTER SCIENCE**

**COURSE ID:**

**103395**

**SUPERVISOR:**

**MA'AM MISBAH ANWER**

**TITLE OF REPORT:**

**MULTI CLIENT CHAT APPLICATION USING NODE.js**

**ABSTRACT:**

Our project is an example of a chat server. It is made up of 2 applications: the client application, which runs on the web server-client application. To start chatting clients should get connected to a server where they can chat. In this project, we built a real-time chat application with emojis that sends and shows messages to a recipient instantly without any page refresh using JavaScript framework Node js and libraries like socket.io. In this we also show the login status of clients.

**Background art of the work/prior-history:**

In this project for front hand we use html & react, and for backhand we use node.js(express) for this we have to install npm it handle peer-to-peer communications in real time, In it we use modules of javascript and import libraries of react needed for this project. Users must be able to log in for the application. If the user skips this step, the application should not be open. If any of the accounts with Chat Application, those accounts must automatically be

added to the user's contact list. Send Message User should be able to send an instant message to any contact on his/her Chat Application contact list. The user should be notified when the message is successfully delivered to the recipient by displaying a message on his own chat screen. In this we maintain a list of online clients and his/her status.

## **DESCRIPTION OF THE PROJECT:**

### **INTRODUCTION:**

For a chat application we will make a server and client-side and for the front hand of our chat board is built using HTML or React js and backhand is by using Node js. In this app, multiple clients are connected and send/received messages. Another feature of this app is that you can also send emojis in chat.

### **OBJECTIVES AND AIMS:**

Broadcasting Chat Server Application is going to be a text communication software, it will be able to communicate between two browsers using point to point/ peer to peer communication. Companies would like to have a communication software wherein they can communicate instantly within their organization. The fact that the software uses an internal network setup within the organization makes it very secure from outside attacks.

### **TECHNIQUES USED:**

The user interacts with the tool using GUI. The GUI Operates in two forms, Contacts Forms and Chat forms. The contact forms contain a list of all friends and the Chat form will be used to chat with friends. We use Node js because the most important advantage of using Node is that we can use JavaScript as both a front-end and back-end language. JavaScript will be used primarily for client-side scripting, in which scripts are embedded in a webpage's HTML/React and run client-side by a JavaScript engine in the user's web browser.

### **RESULTS:**

We got our result successfully by making a Multi-Client Chat Application as we discussed above and in the proposal, we made a react and node js based application and we also tested it.

### **CONCLUSION:**

There is always room for improvement in any apps. Right now we are just dealing with text communication and emoji sharing. Several Android apps serve a similar purpose as this project, but these apps were rather difficult to use and provide confusing interfaces. A positive impression is essential in the human relationship as well as in human-computer interaction. This

project hopes to develop a chat service good. In the future, we may be extended to include features such as 1. Voice Message 2. Video Message 3. Audio Call 4. Video Call 5. Group Call and so on but, hope for the better project is done and try our best to fulfill all needs of this project which work in the real world also. This project is open for all as we upload it to GitHub and attach the link with it for convenience of the advisor.