

**GLA University, Mathura -2021**

**Mini Project**

**Final Report**



**TEAM DETAILS**

**Name :-**

- 1.SHUBHAM BHARDWAJ (191500797)
- 2.TANMMAY GAUR (191500844)
- 3.HARESH SINGH (191500311)
- 4.UMANG CHAUHAN (191500869)
- 5.SHUBHRANSH SINGH (191500807)

**PROJECT NAME: "I-CHAT"**

***Supervised By:-***

***Mandeep Singh***

***(Technical Trainer)***

## *Content*

- Certificate
- Synopsis
- Acknowledgement
- Abstract

### 1.Introduction

#### 1.1 Motivation

#### 1.2 Problem Statement

#### 1.3 Area of Computer Science

### 2. Software Requirement Analysis

#### 2.1 System Analysis

#### 2.2 Role of System Analyst

#### 2.3 Methodology

### 3. Implementation Details

### 4. Progress till Date & The Remaining Work

### 5. Some Screenshots

### 6. Future Scope

### 7. References

## *Certificate*

This is to certify that *Shubham Bhardwaj, Tanmay Gaur, Haresh Singh, Umang Chauhan*, and *Shubhransh Singh* students of B.Tech (CSE) 3rd year has successfully Completed the MINI PROJECT named *I-Chat* on Web Development under the Guidance of Mr. Mandeep Singh During 2021-22.

Signature:

Mr. Mandeep Singh

(Mentor)

## **Project Synopsis**

### **Project Information:**

Title Of Project/Training/Task	I-Chat
Role & Responsibility	<p>Tanmay &amp; Haresh: CSS work</p> <p>Shubhransh: HTML work</p> <p>Shubham &amp; Umang: Front-end &amp; Back-End developer and documentation work.</p>
Technical Details	<p>Hardware Requirements:</p> <ul style="list-style-type: none"><li>• 2 GB RAM</li><li>• 20 GB OF HARD DISK</li><li>• Processor i3 (7<sup>th</sup> Gen)</li><li>• Internet</li></ul> <p>Software Requirements:</p> <ul style="list-style-type: none"><li>• VS Code</li><li>• Socket.io (with back-end)</li></ul>

Implementation Detail	Fully Implemented
Application Software	Front-end <ul style="list-style-type: none"> <li>• HTML</li> <li>• CSS</li> <li>• JavaScript</li> </ul> Back-end: <ul style="list-style-type: none"> <li>• Node.js</li> <li>• Socket.io</li> </ul>

## **Summary of the Training Work:**

During the completion of this project I-Chat we learnt about HTML5, CSS, and some basic Javascript. Shubham Bhardwaj is the founder of the project and implemented major part of the backend and he is the designer of the project and completed some part of frontend along with documentation work required doing the project building period and also did many modules according to the design. Umang Chauhan worked on front-end's html along with Shubhransh, Tanmay, and Haresh. Our mentor Mr. Mandeep Singh guided us and helped a lot whenever required by us.

### *Acknowledgment*

We have done our best in this project. However, it would not have been possible without the kind support and help of our Mentor . I would like to extend my sincere thanks to him.

We are highly indebted to Mr. Mandeep Singh for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project. We would like to express my gratitude towards our parents & member of GLA University for their kind co-operation and encouragement which help us in completion of this project.

## **Abstract**

We are creating “A Web-browser based application” named “**I-chat**”. As we know, every Website has a front-end as well as a back-end. So, **Front-end** is a practice of converting data to a graphical interface, through the use of HTML, CSS, and JavaScript, so the users can view and interact with that data and,

**Back-end** development refers to server-side development. It focuses on databases, scripting, and architecture of websites.

Code written by back-end developers helps to communicate the database information to the browser.

So in this mini-project we are using:-

- CSS
- Javascript
- Html

As a front- end for our real time-chat web-page.

And for back-end:-

- Node Js

Another technology that will be used in this project is named as Socket.io(will be explained below).



## **Introduction**

### **Overview:**

- All the functional/non-functional requirements, corresponding DFD's, UML and Use Case Diagrams have been organized in this report.
- The complete description of the application followed by the functionalities has been listed initially. Later on, the Webpage has been described diagrammatically with the help of different designing tools like Data Flow Diagram, Use Case Diagram.

### **Motivation:**

In today's era, we all are aware of the need to communicate with each other. However we the people are so busy in our own lives, we forget and sometimes ignore that *the world beyond us exists*. One solution to this is **Virtual Communication**.

### **Problem Statement:**

Communicating virtually with people is when individuals interact without being face-to-face but through words and texts that are sent at a higher speed than letters. We can share information, comment, and ask questions, develop social etiquette.

Communicating virtually make us elevate from being digitally literate to digitally fluent, collaborate using appropriate virtual tools and system and they are cost-effective too.

So putting our efforts into this application, we are going to create I-chat.

In that, we will be using Socket.io and many more technologies.

### **Q. What Socket.io is?**

- Socket.io is the two-way connection between the server and the client.
- As we know Http, Https protocol is a one-way connection between server and client. So in this, we are going to use ws(web socket), wss(web socket secure) protocol to establish the connection between the server and client too.

As we know a web-page contains front-end and back-end.

Let's describe the technologies used to build our front-end.

### **Q. What CSS is?**

- CSS is used for styling the Webpages.
- CSS is of three types i.e. internal CSS, external CSS, inline CSS.
- It describes how Html elements are to be displayed on the screen.
- It involves many designing for the bare Mark-up to create it attractive.

## **Q. What is HTML?**

- The first version of Html was written by Tim Berners-Lee in 1993. Since then, there have been many different versions of Html. We are using Html 5 in our project.
- Html is a programming language used to describe the structure of information on a webpage.
- Front end developer uses Html as to make the structure of their ideas or we can say a raw website. In that, design our idea by using CSS, JS.

## **Q. What is JavaScript?**

- JavaScript was created by Brendan Eich in 1995.
- It is the programming, Scripting language that allows implementing complex features on web pages.
- JS helps the web developer to make a dynamic and interactive webpage by implementing a custom client-side script.

Now, the technology used in the back-end is:-

## **Q. What Node.Js is?**

- NodeJs was developed by many developers but the original author was Ryan Dahl in 2009.
- NodeJs is an open-source, cross-platform that executes JavaScript code outside a web browser.
- It also works as a backend. And we are going to use it as a backend.

## **Area of Computer Science**

This real-time chat web-browser is been developed for web-technologies on various devices like iPad, mobile phones, desktop, laptops, etc by many users. Multiple users can interact at the same time on this web-page and can chit-chat there.

Web-Technologies like:-

- Firefox
- Windows
- Mac-OS
- Chrome
- Opera

## **Hardware Requirements**

### **Hardware requirements**

In hardware requirement, we require all those components which will provide us the platform for the development of the project. The minimum hardware required for the development of this project is as follows-

**RAM - a minimum of 128 MB**

**Hard disk- minimum 4GB**

**Processor - core i3**

These all are the minimum hardware required for our project. We want to make our project to be used in any type of computer therefore, we have taken a minimum configuration to a large extent. 128 MB ram is used so that we can execute our project in the least possible ram. 500 GB hard disk is used because the project takes less space

## **Software Requirement**

The software can be defined as a program that runs on our computer. It acts as petrol in the vehicle. It provides the relationship between the human and computer. Various software is needed in this project for its development.

Which are as follows-

**Operating system- windows7**

**Others- visual studio**

### **Contribution Summary OUR TEAM:**

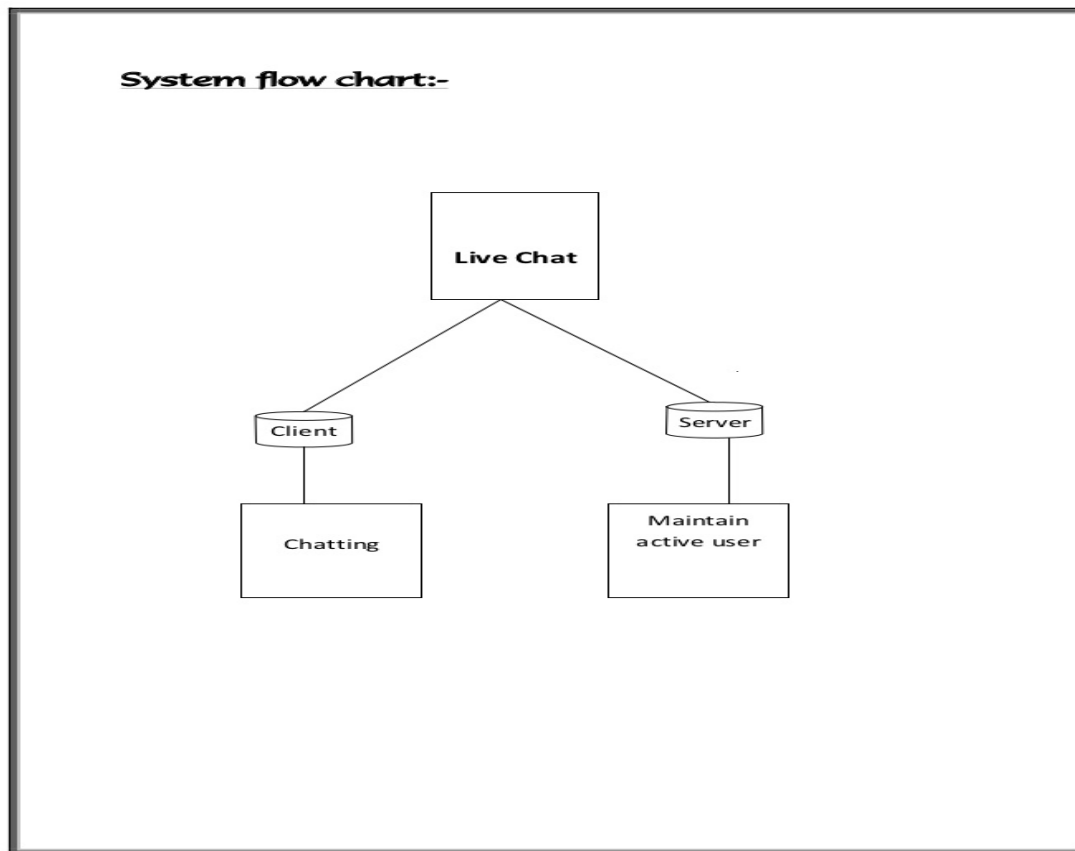
Founder , front-end and back-end :- Shubham & Umang

Documentation: Shubhransh, Haresh and Tanmay

## **Methodology**

Our methodology is designed to help you take maximum advantage of the internet technologies. It incorporate all aspects related to our website and allows us to ensure that the final product is of the highest standards. Below are the steps we will take to ensure that all your deliverables are completed in time, within budget also we will try to solve each and every problem efficiently

## Implementation Details



A live chat contains two parts:-

- Client
- Server

So we have divided our mini project into two parts as shown.

As per the implementation progress, we have completed our part 1 with full enthusiasm. And part 2 is in progress.

### **Part 1:- Client-side includes front-end.**

- Basic Mark-up (HTML)
- Designing HTML(CSS)
- Building the logic of webpage(JavaScript)

## **Part 2:- Server-side includes backend.**

- Connecting frontend with backend.
- Connection of socket.io from front-end.
- Connection of node.js from front-end.

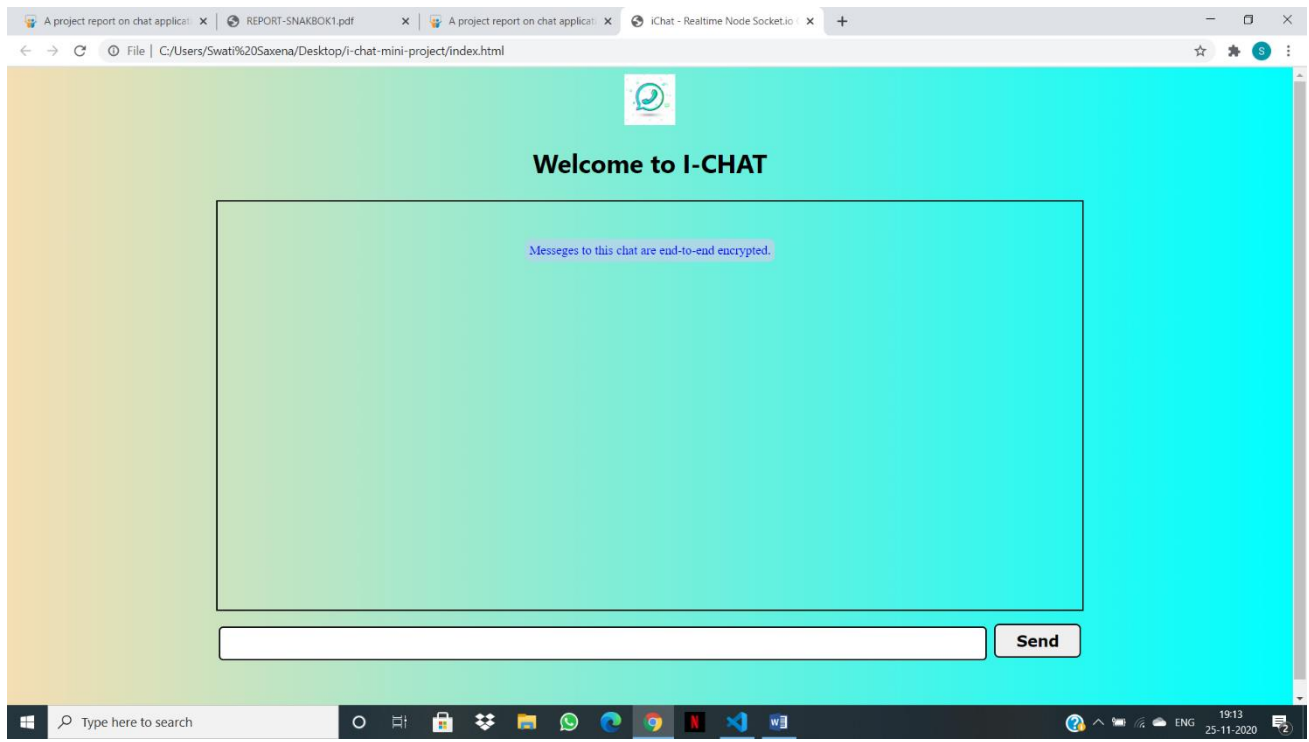
## **Part 3:- Final Submission**

- It includes part1+ part2.

Part1 , part2 and part3 Completed.

## **Appendix 1**

### **Source-Code of Front-End:**



## Front-End :

- Html as (Index.html)
- Css as (Style.css)
- javaScript as (Client.js, client side javaScript) and (Index.js for layout)

### Index.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-
width, initial-scale=1.0">
  <title>iChat - Realtime Node Socket.io Chat A
pp</title>
  <script defer src="http://localhost:4000/sock
et.io/socket.io.js"></script>
  <script defer src="js/client.js"></script>
  <link rel="stylesheet" href="css/style.css">
</head>

<body>
  <nav>

    <h1>Welcome to I-CHAT</h1>

  </nav>
```



```

<div class="container">
  <div id="block">
    <p id="encryp"> Messeges to this chat
are end-to-end encrypted.</p>
  </div>
</div>

  <div class="send">

    <form action="#" id="send-container">
      <input type="text" name="messageIn
p" id="messageInp">
      <button class="btn" type="submit">
Send</button>

    </form>
  </div>

</body>
</html>

```

**Style.css:**

```
body{
```

```
    height: 100vh;
    background-image: linear-
gradient(to right,wheat,aqua);
}
.logo{
    display: block;
    margin: auto;
    width: 60px;
    height: 60px;
}
h1{
    margin-top: 10 px;
    font-size: 30px;
    font-
family: 'Segoe UI', Tahoma, Geneva, Verdana, sans
-serif;
    text-align: center;
}

.container{
    max-width: 955px;
    border: 2px solid black    ;
    margin: auto;
    height: 55vh;
    padding: 33px;
    overflow-y: auto;
}
.message{
    background-color: rgb(233, 227, 227);
    width:auto;
    overflow: auto;
    padding: 10px;
```

```
margin: 17px 12px;
border: 3px solid black;
border-radius: 10px;
}
.left{
  float: left;
  clear: both;
}

.right{
  float: right;
  clear: both;
}
#send-container {
  display: block;
  margin: auto;
  width: 68%;
  max-width: 100%;
  text-align: center;
  margin-top: 15px;
}

#messageInp {
  width: 88%;
  border: 2px solid black;
  border-radius: 5px;
  height: 34px;
}

.btn {
  font-size: 18px;
  cursor: pointer;
```

```
border: 2px solid black;
border-radius: 6px;
height: 40px;
width: 10%;
font-weight: bold;
font-family: Verdana, Geneva, Tahoma, sans-serif;
margin-left: 5px;
}

.btn:hover {
  color: blue;
}

#block {
  display: flex;
  justify-content: center;
  align-items: center;
}

#encryp {
  text-align: center;
  color: blue;
  background-color: lightblue;
  border-radius: 7px;
  padding: 5px;
  margin: 10px;
  font-size: 15px;
}

@media(max-width:900px) and (min-width: 500px) {
  .btn {
```

```

        width: 20%;
    }
    .message {
        width: 60%;
    }
}

@media(max-width: 499px) {
    .btn {
        width: 28%;
        height: 30px;
    }
    .message {
        width: 75%;
        font-size: 16px;
    }
}

```

Index.js:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-
width, initial-scale=1.0">
    <title>iChat - Realtime Node Socket.io Chat A
pp</title>
    <script defer src="http://localhost:4000/sock
et.io/socket.io.js"></script>
    <script defer src="js/client.js"></script>

```

```
<link rel="stylesheet" href="css/style.css">
</head>

<body>
  <nav>

    <h1>Welcome to I-CHAT</h1>

  </nav>
  <div class="container">
    <div id="block">
      <p id="encryp"> Messages to this chat
are end-to-end encrypted.</p>
    </div>
  </div>

  <div class="send">

    <form action="#" id="send-container">
      <input type="text" name="messageIn
p" id="messageInp">
      <button class="btn" type="submit">
Send</button>

    </form>
  </div>

</body>
</html>
```

## Client.js:

```
const socket = io('http://localhost:4000');

const form = document.getElementById('send-
container');
const messageInput = document.getElementById('mes
sageInp')
const messageContainer = document.querySelector('.
container')

var audio_receive = new Audio('ting.mp3');

const append = (message, position) => {
  const messageElement = document.createElement
('div');
  messageElement.innerText = message;
  messageElement.classList.add('message');
  messageElement.classList.add(position);
  messageContainer.append(messageElement);
  if (position == 'left')
    audio_receive.play();
}

form.addEventListener('submit', (e) => {
  e.preventDefault();
  const message = messageInput.value;
  append(`You : ${message}`, 'right');
```

```
socket.emit('send', message);
messageInput.value = ''
})

const name = prompt("Enter Your Name To Join The Chat");
socket.emit('new-user-joined', name);

socket.on('user-joined', name => {
  append(`${name} has joined the chat...!`, 'right')
})

socket.on('receive', data => {
  append(`${data.name} :${data.message}`, 'left')
})

socket.on('left', name => {
  append(`${name} left the chat`, 'left')
})
```

### **Back-end:**

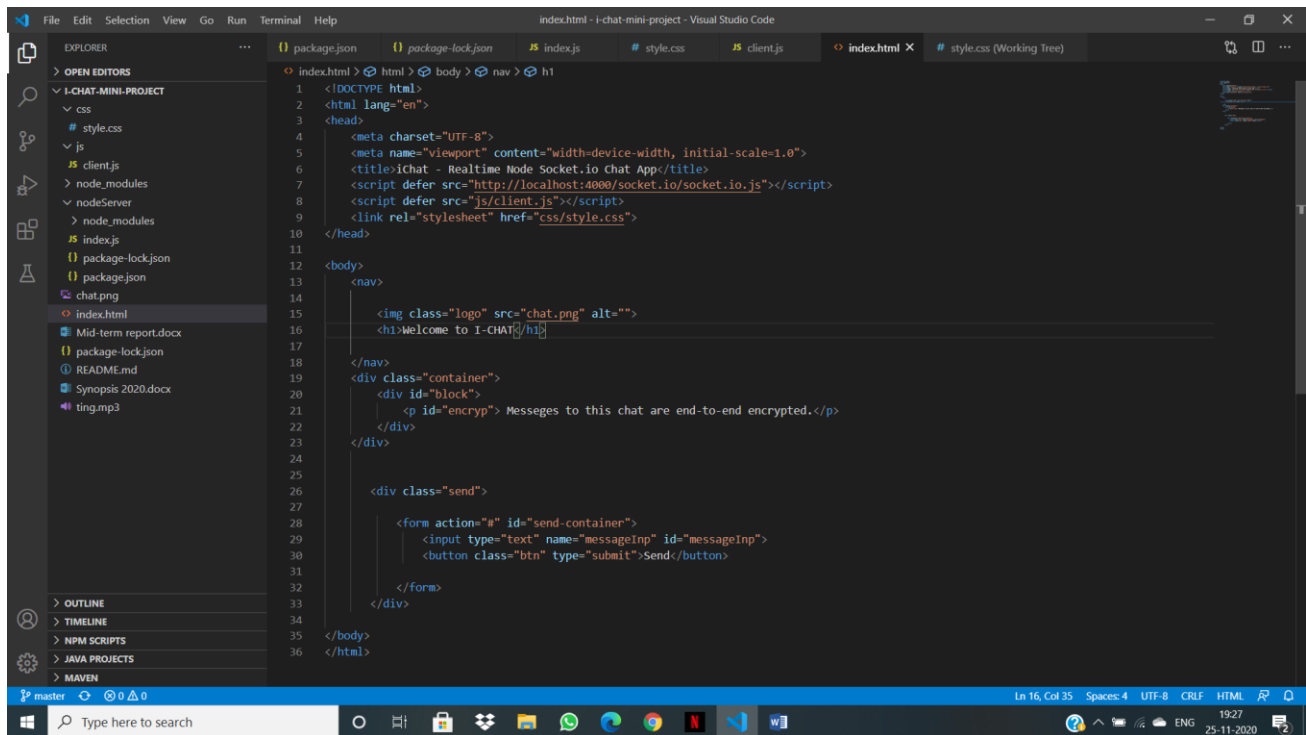
- node.js(code on terminal)
- Socket.io(code on terminal)



## Appendix 2

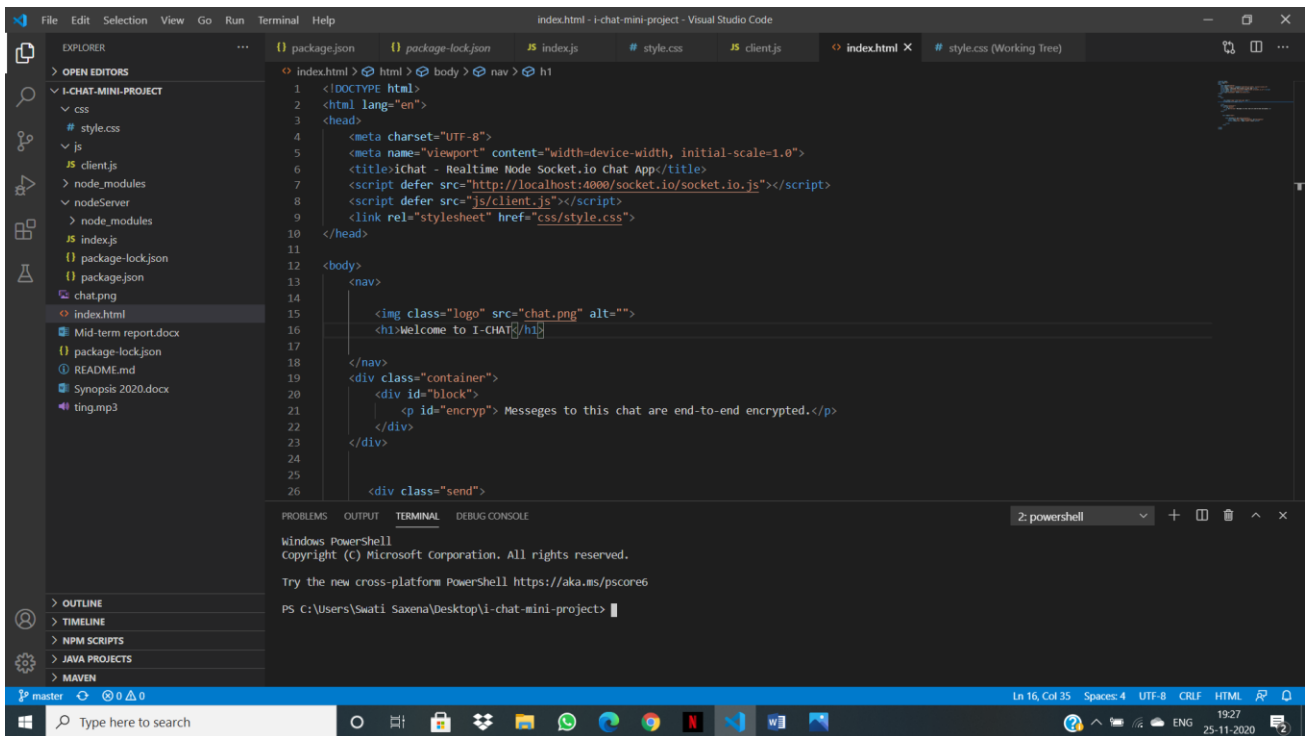
### Output:

Firstly the node server will be run on vs code's terminal.so that our project should run.

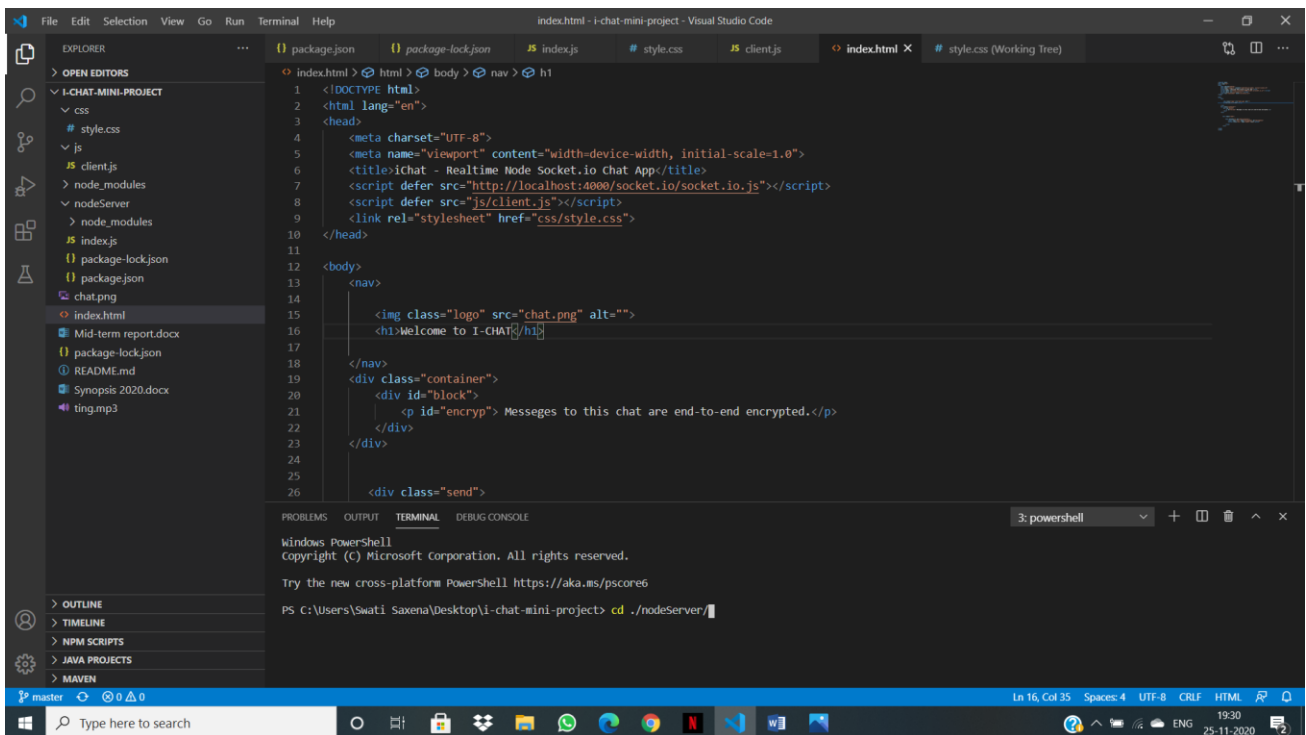


```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>iChat - Realtime Node Socket.io Chat App</title>
7   <script defer src="http://localhost:4000/socket.io/socket.io.js"></script>
8   <script defer src="js/client.js"></script>
9   <link rel="stylesheet" href="css/style.css">
10 </head>
11
12 <body>
13   <nav>
14
15     
16     <h1>Welcome to I-CHAT</h1>
17
18   </nav>
19   <div class="container">
20     <div id="block">
21       <p id="encrypt">Messages to this chat are end-to-end encrypted.</p>
22     </div>
23
24     <div class="send">
25
26       <form action="#" id="send-container">
27         <input type="text" name="messageInp" id="messageInp">
28         <button class="btn" type="submit">Send</button>
29       </form>
30     </div>
31
32   </div>
33 </body>
34 </html>
```

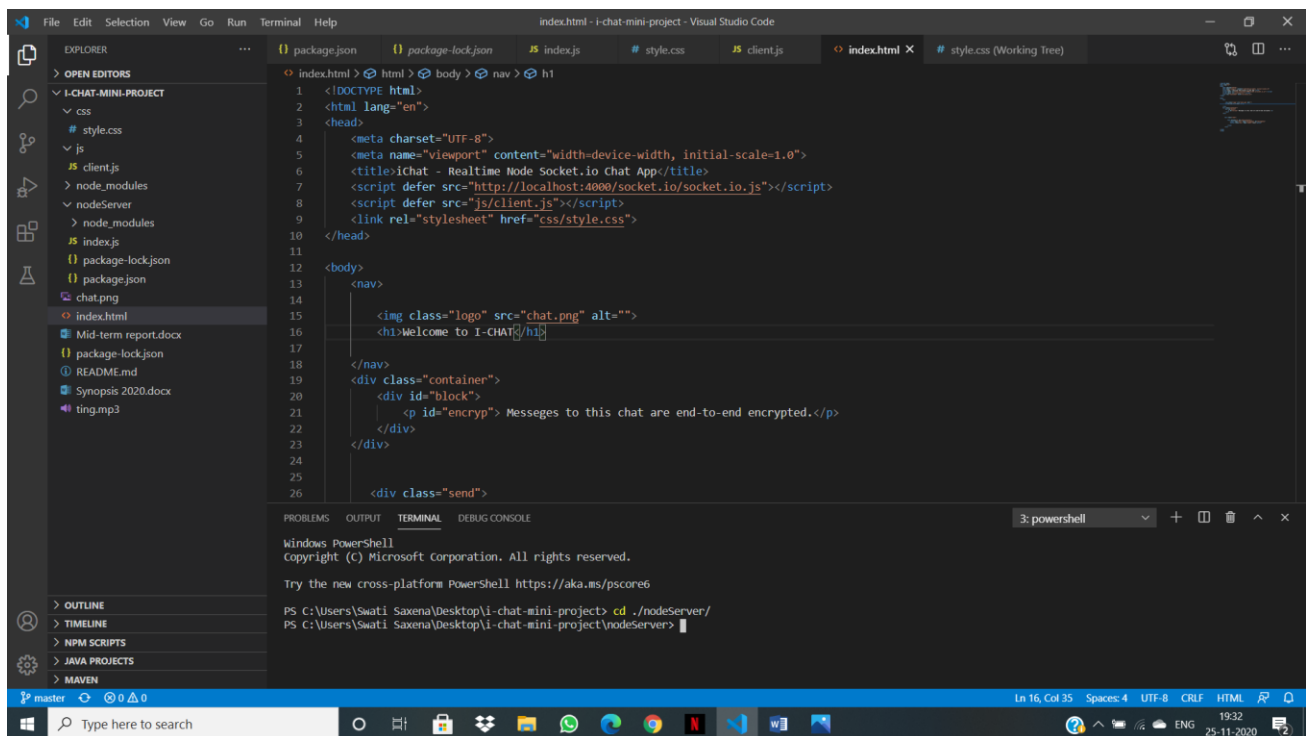
Click on terminal >> new terminal



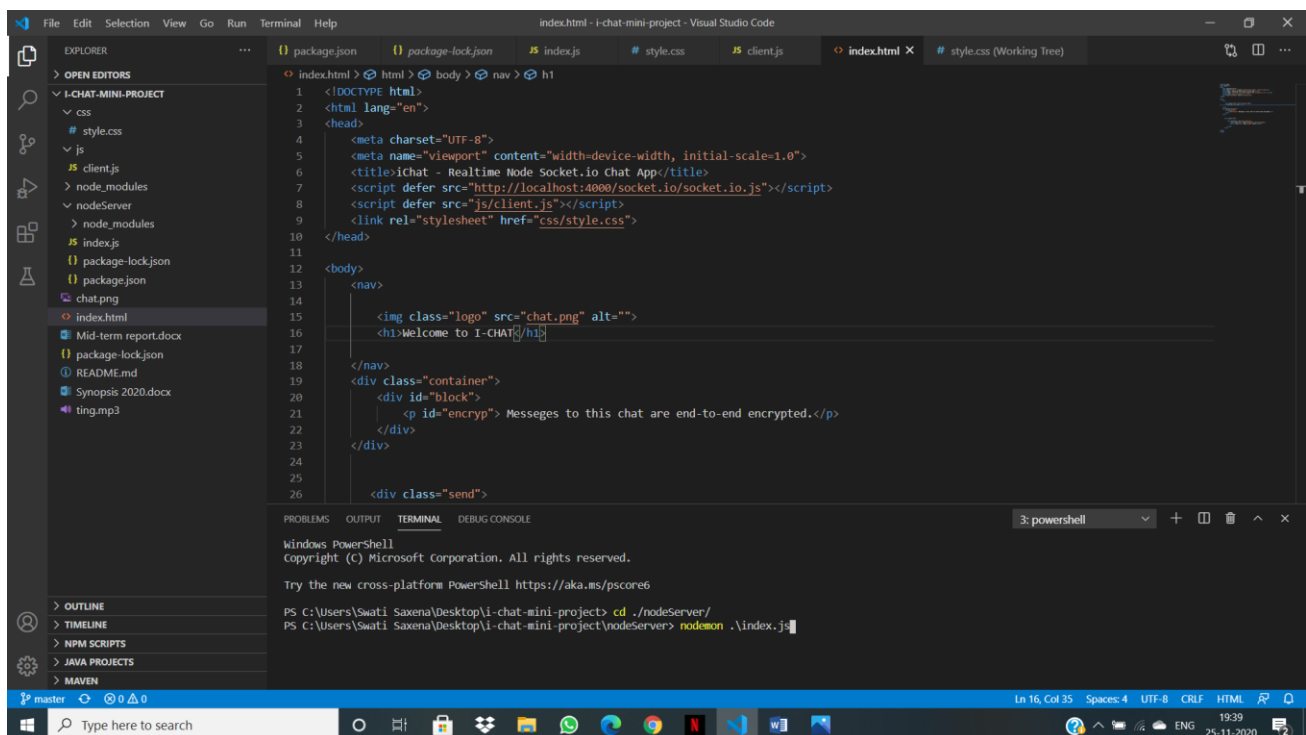
Then write the command `cd ./nodeServer/`



Then give the path where you want to connect this node server.



Write command nodemon ./index.js



So after writing the command nodemon ./index.js. The nodeServer gets connected to the Index.js.

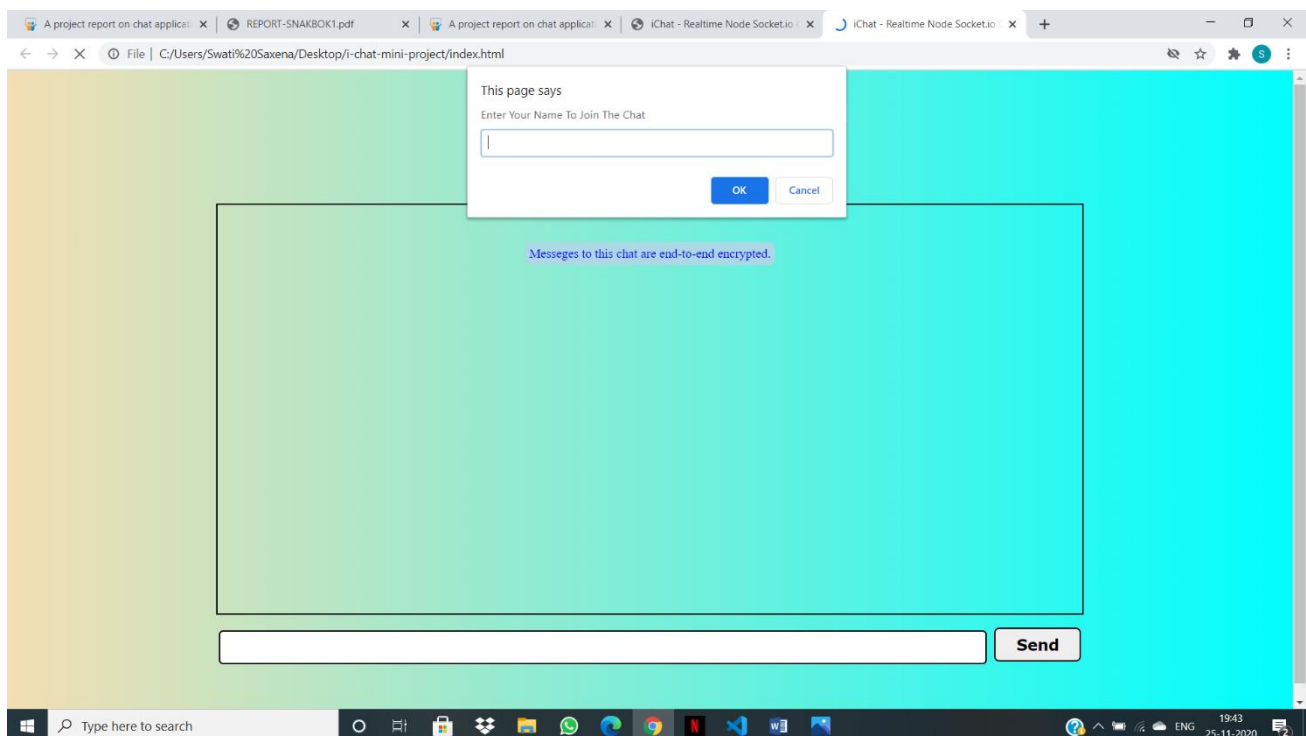
The screenshot shows the Visual Studio Code editor with the 'iChat-mini-project' open. The Explorer sidebar on the left lists files including 'css', 'js', 'node\_modules', 'package-lock.json', 'package.json', 'chat.png', 'index.html', 'Mid-term report.docx', 'package-lock.json', 'README.md', 'Synopsis 2020.docx', and 'ting.mp3'. The main editor displays the 'index.html' file with the following code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>iChat - Realtime Node Socket.io chat App</title>
7   <script defer src="http://localhost:4000/socket.io/socket.io.js"></script>
8   <script defer src="js/client.js"></script>
9   <link rel="stylesheet" href="css/style.css">
10 </head>
11 <body>
12   <nav>
13     
14     <h1>Welcome to I-CHAT</h1>
15   </nav>
16   <div class="container">
17     <div id="block">
18       <p id="encrypt">Messages to this chat are end-to-end encrypted.</p>
19     </div>
20   </div>
21   <div class="send">
22   </div>
23 </body>
24 </html>
```

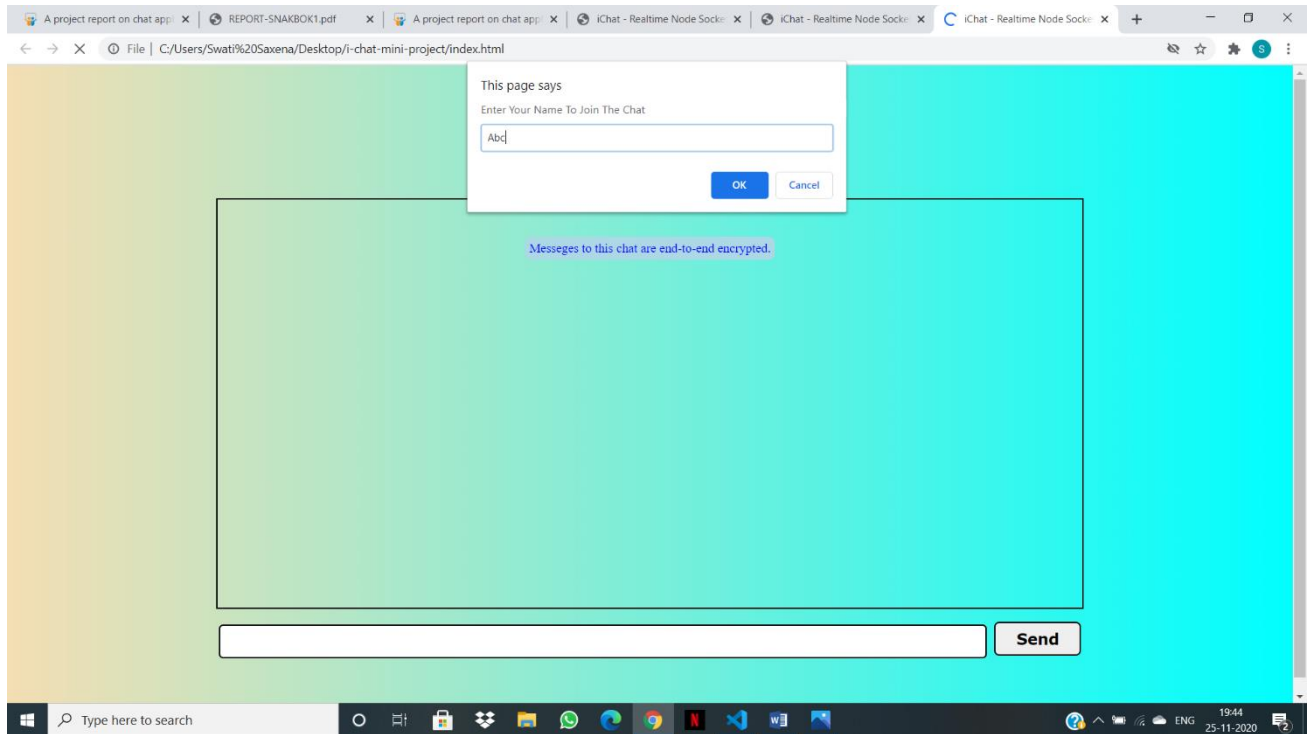
The bottom panel shows the terminal with the following output:

```
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\Swati Saxena\Desktop\i-chat-mini-project> cd .\nodeServer/
PS C:\Users\Swati Saxena\Desktop\i-chat-mini-project\nodeServer> node .\index.js
[nodemon] 2.0.6
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting node .\index.js
```

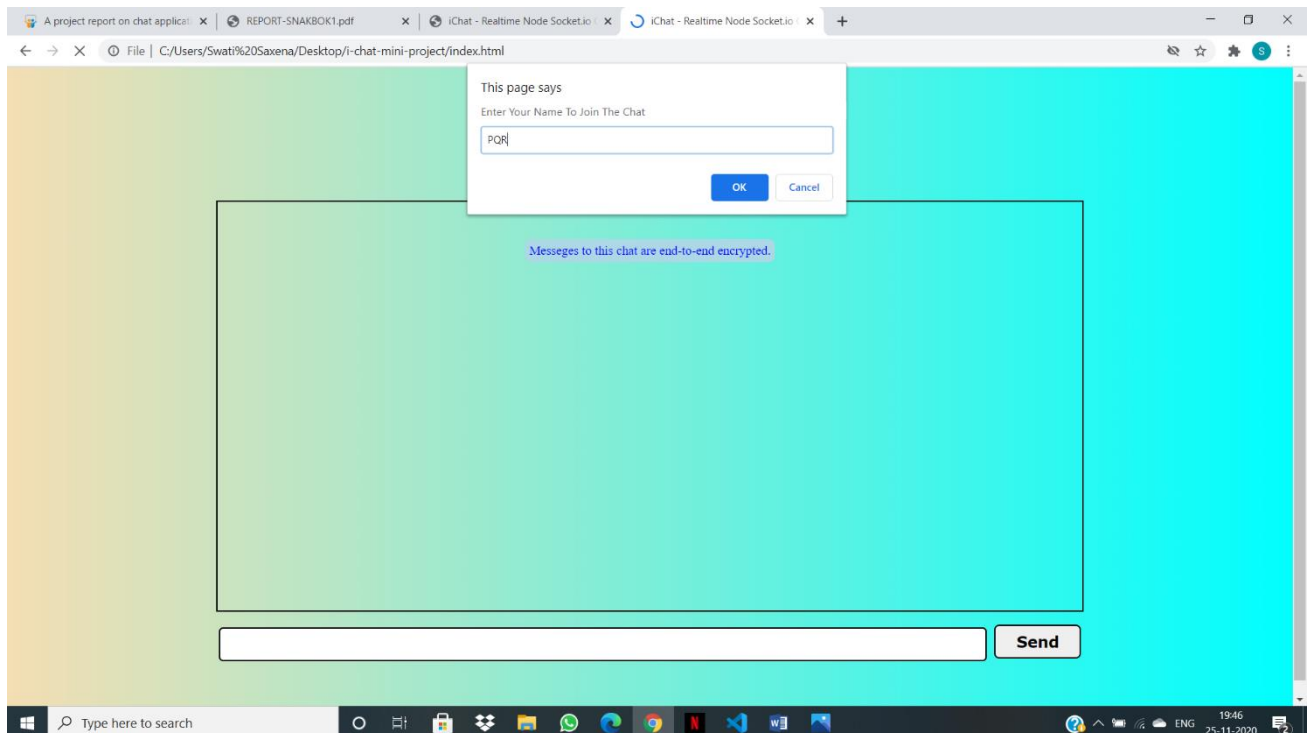
Then we will start the live Server it will look like this:



Here user will enter the name to join the chat.

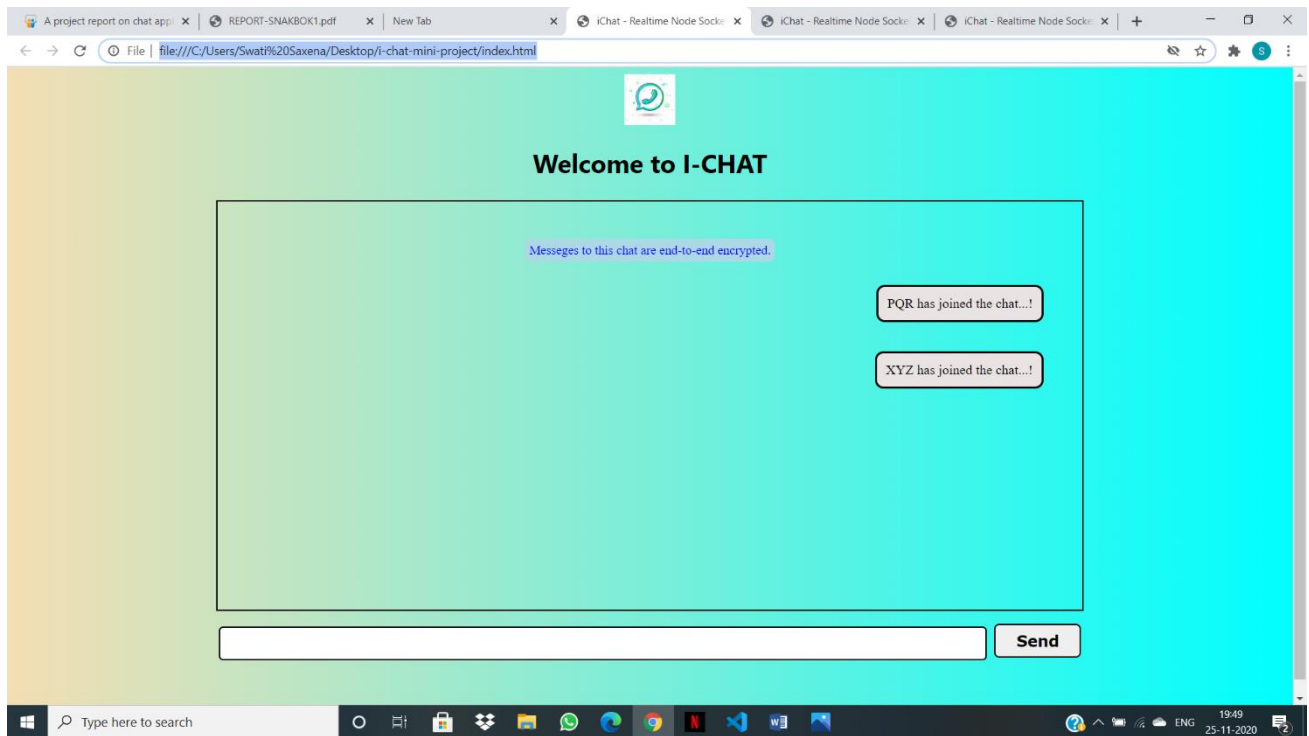


Then let the another user joins the chat.

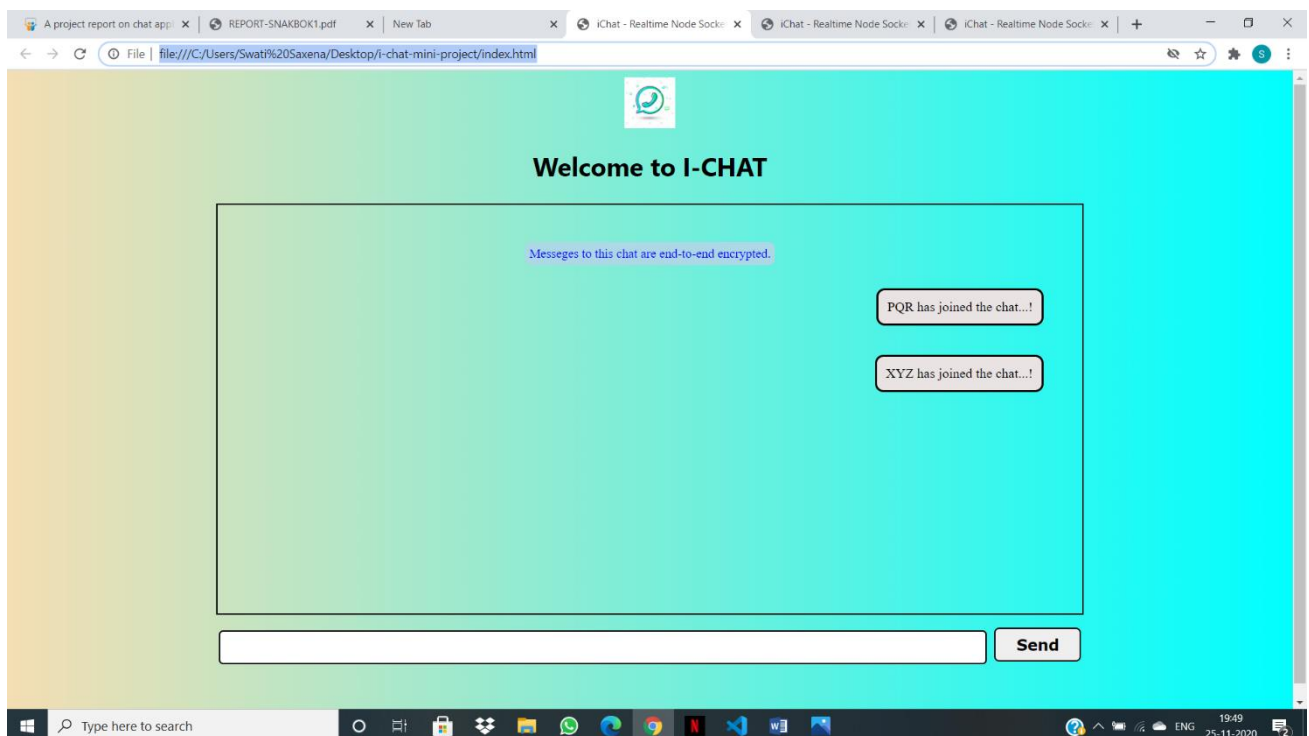


Then the notification to the users apart from the user joined will be send with the help of socket.io.

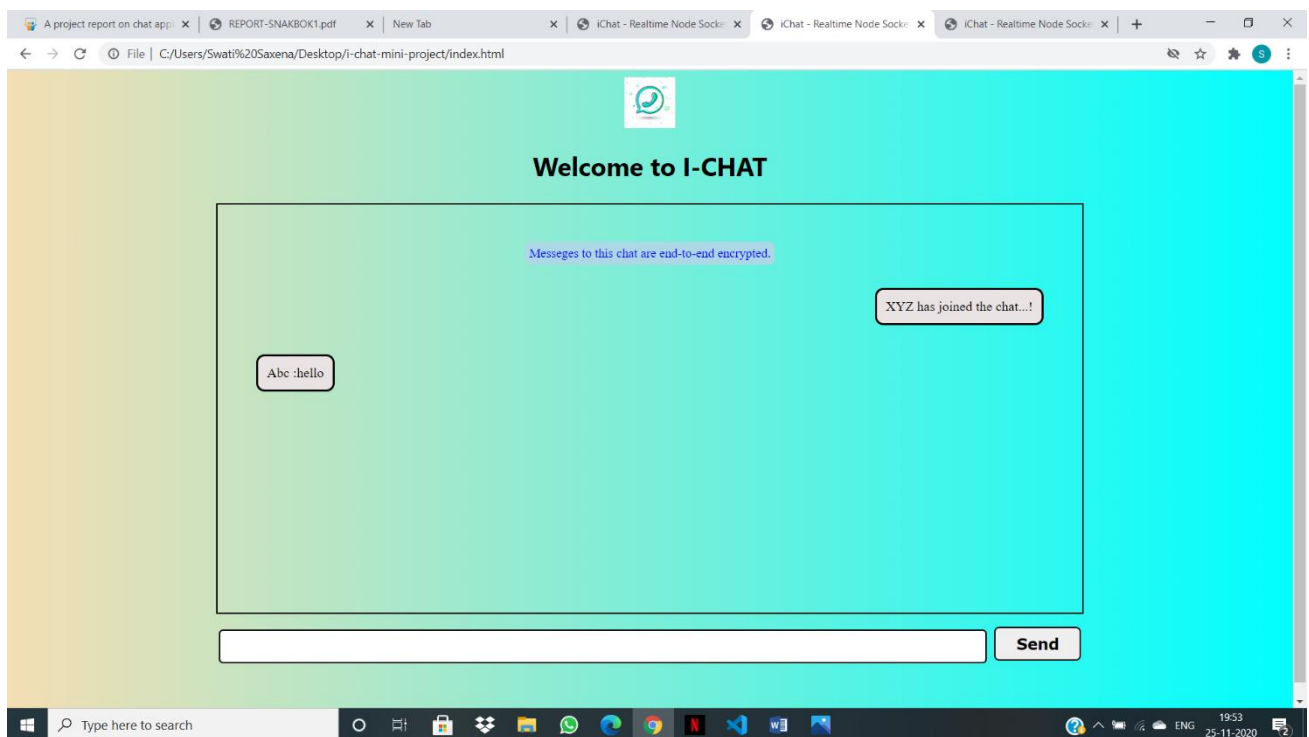
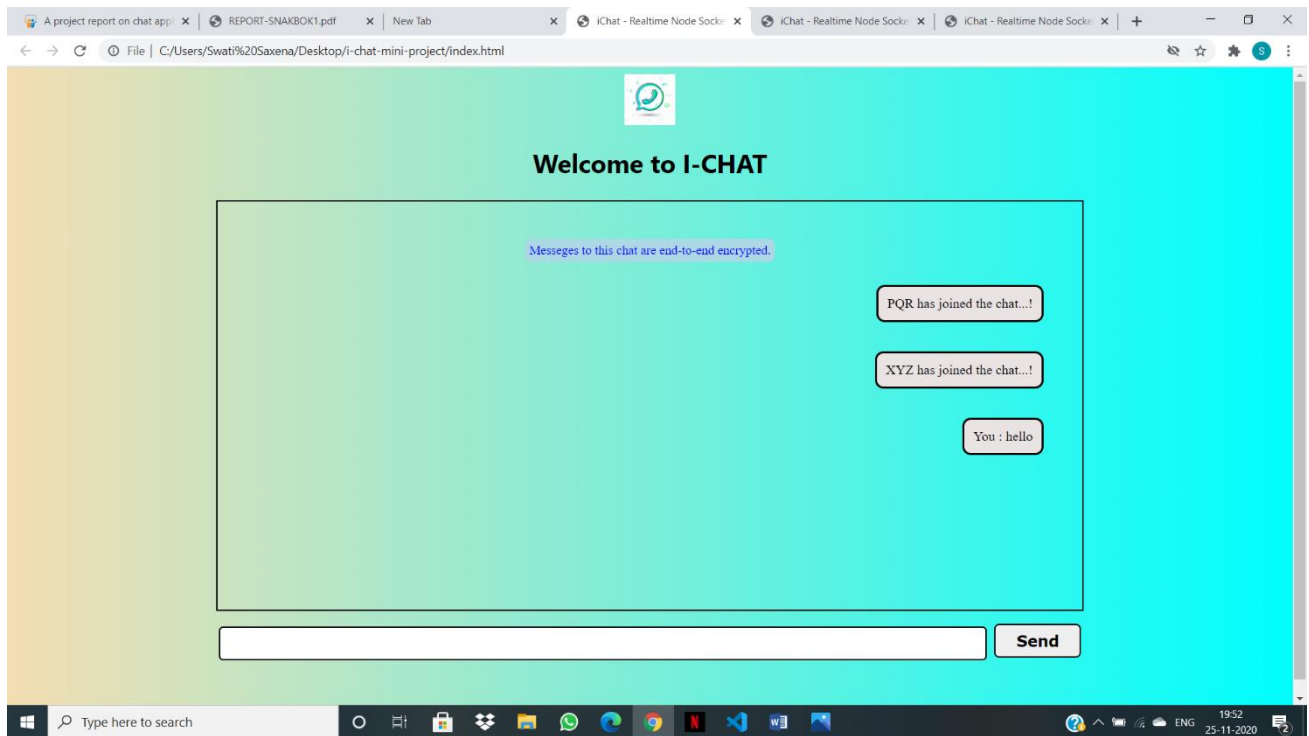
Notification be like XYZ has joined the chat...!



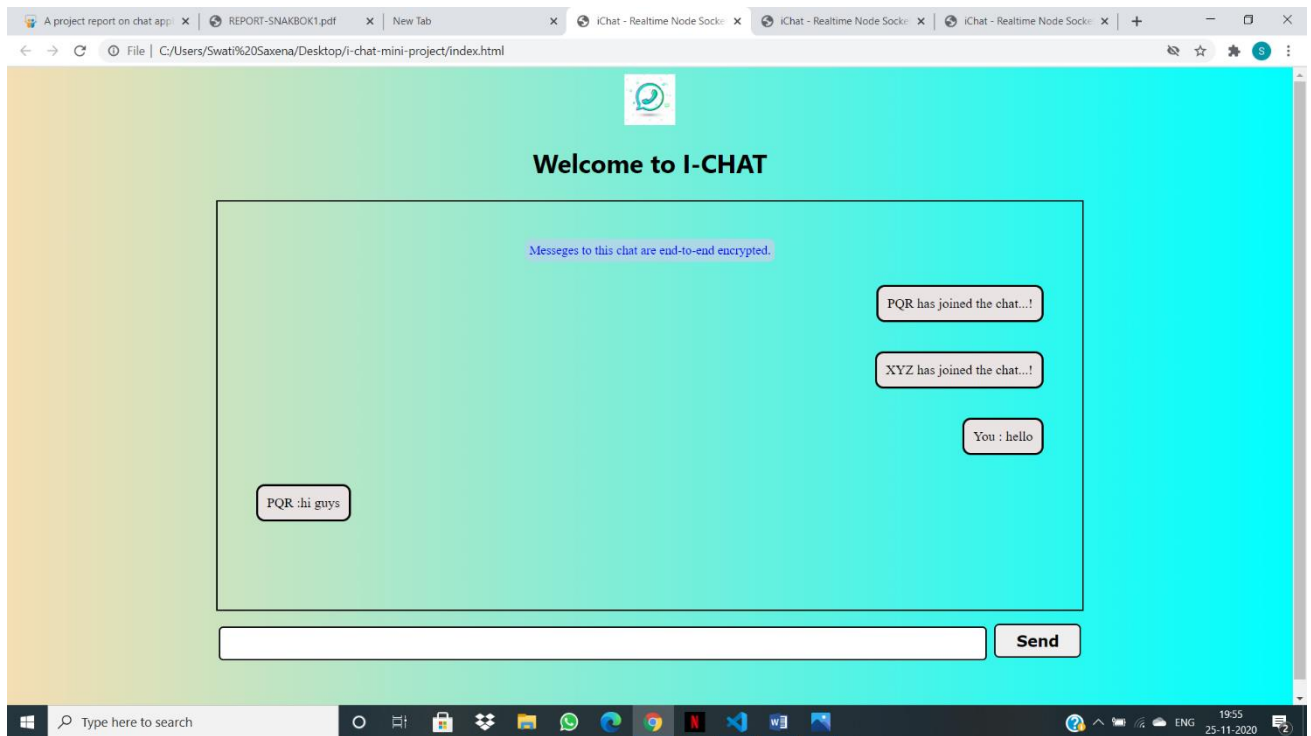
Likewise,



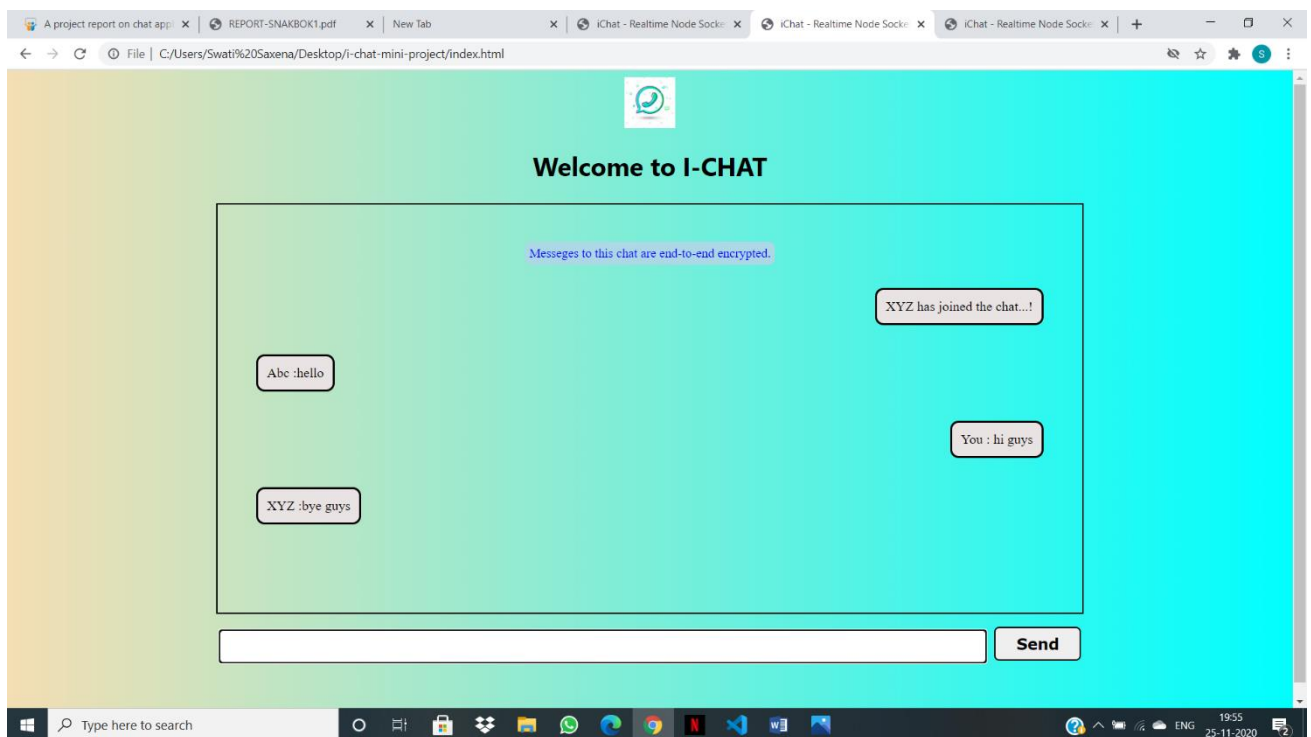
Then any user types the message then it would reflect to another windows too.



The message will receive with a notification sound. Offcourse that cant be shown.

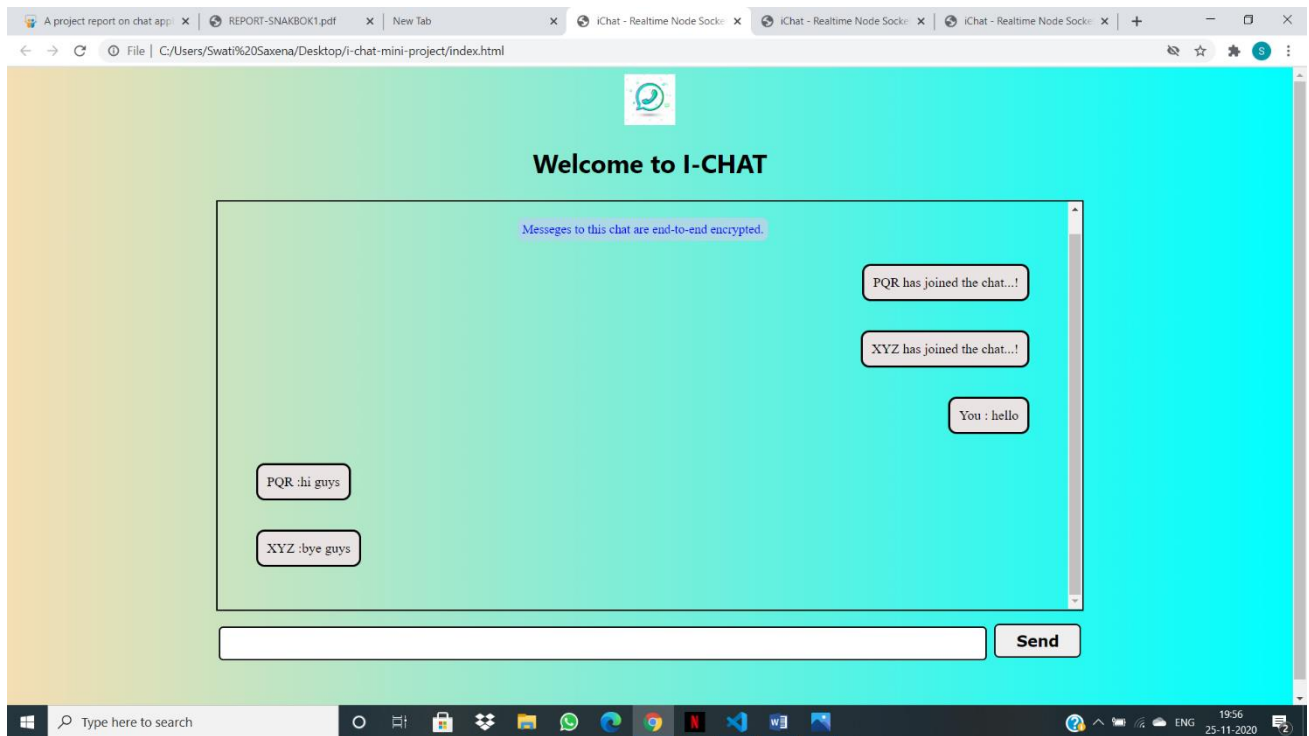


Likewise,

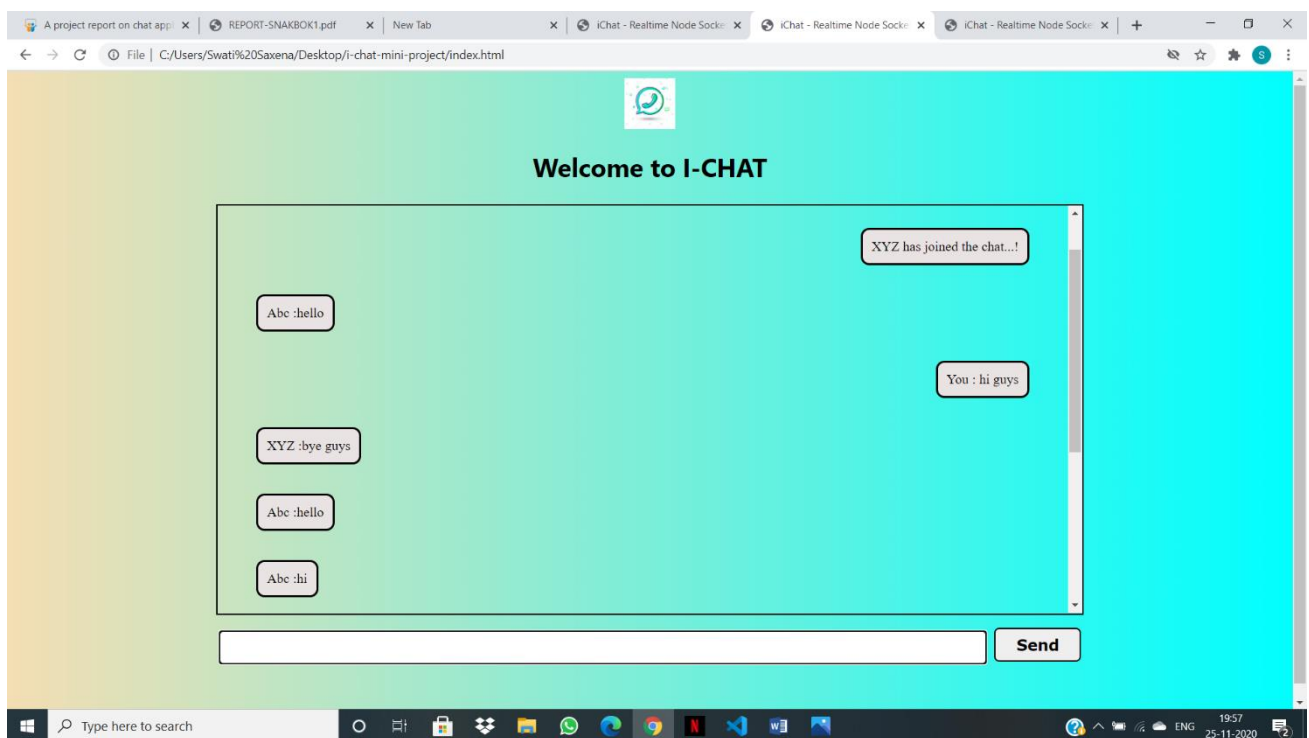


And another

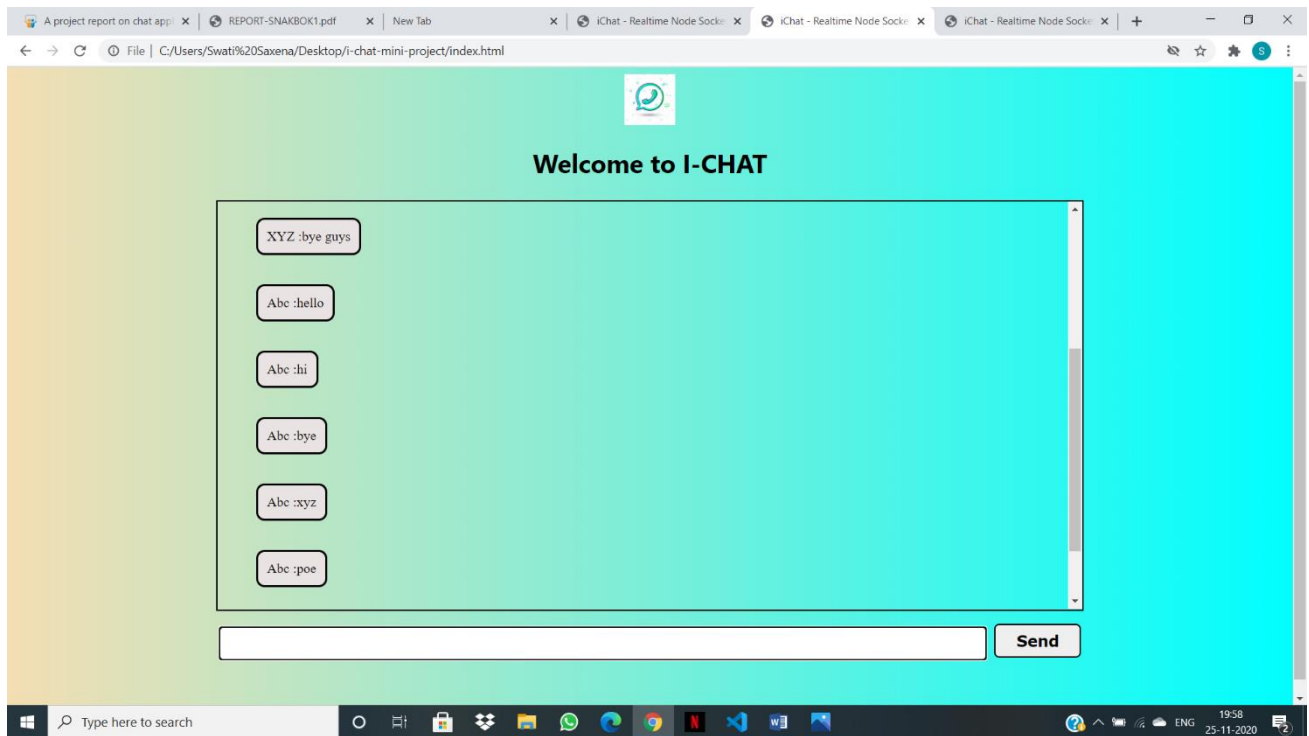




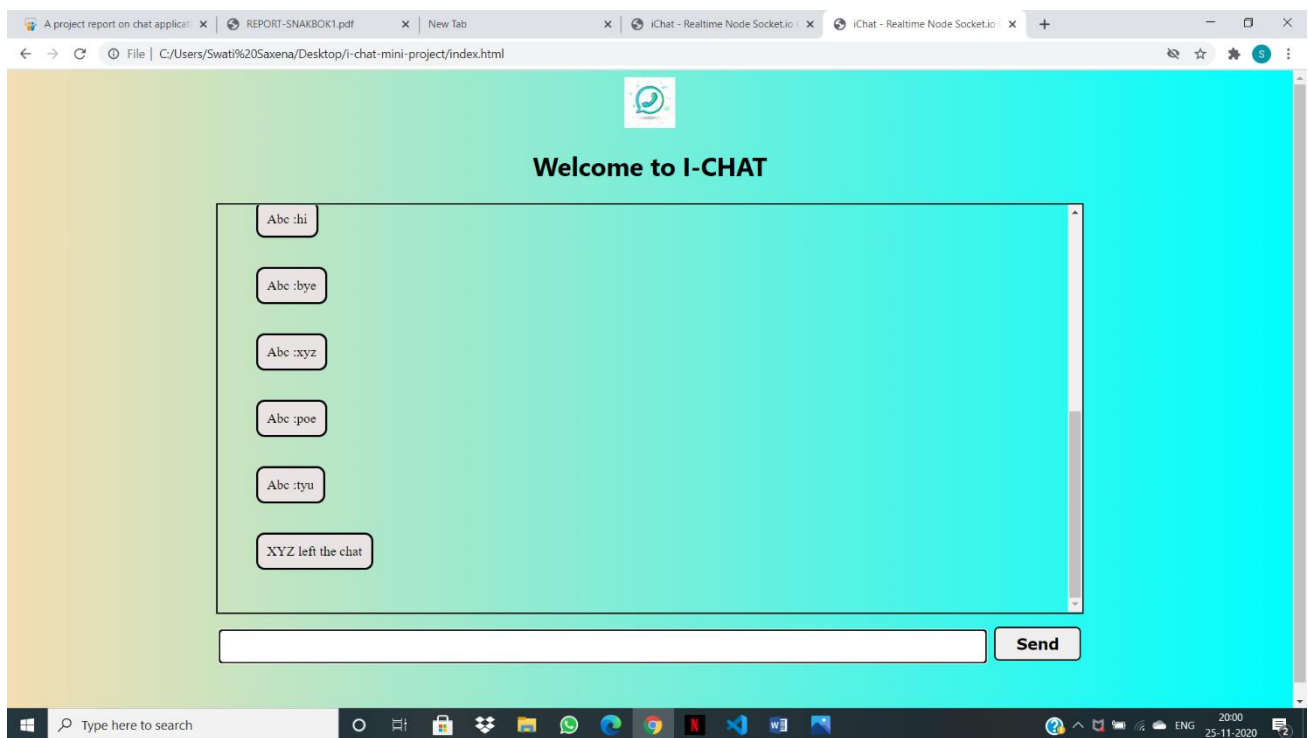
If message gets overflow then to manage all of these. There is a scrolling bar on right.



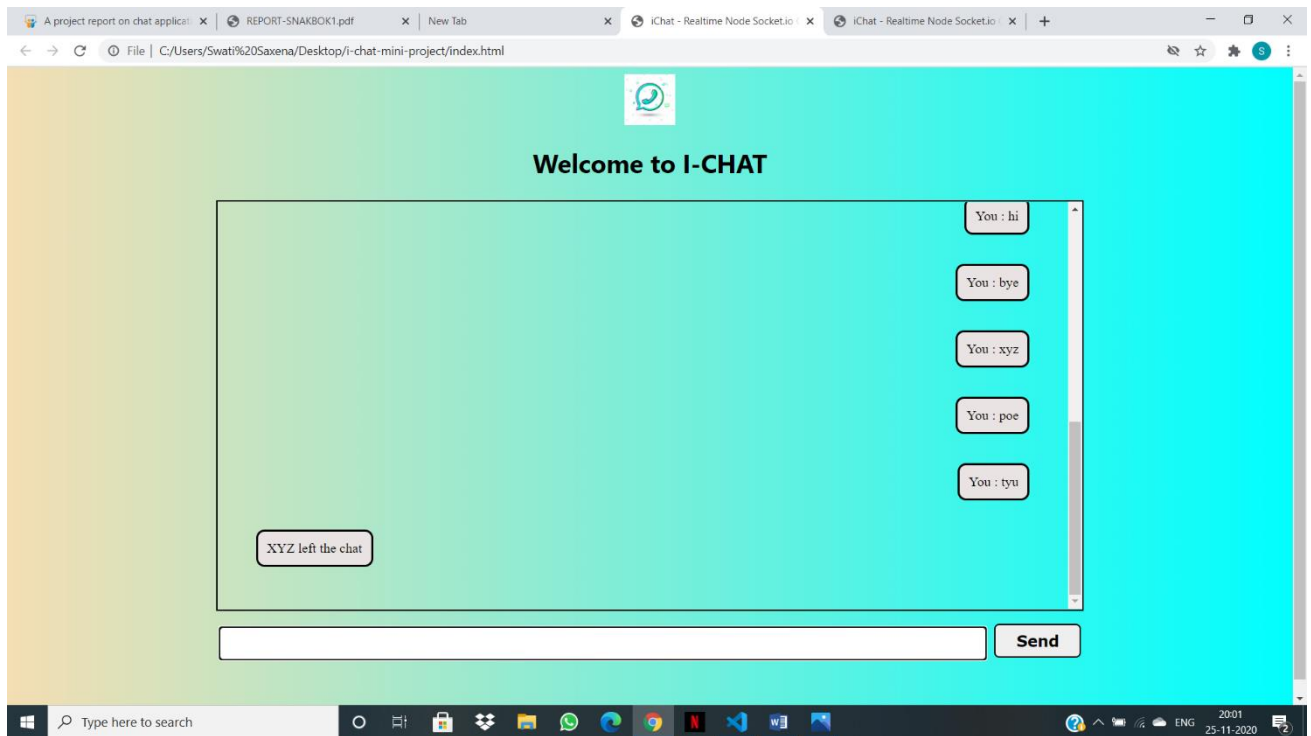
Likewise,



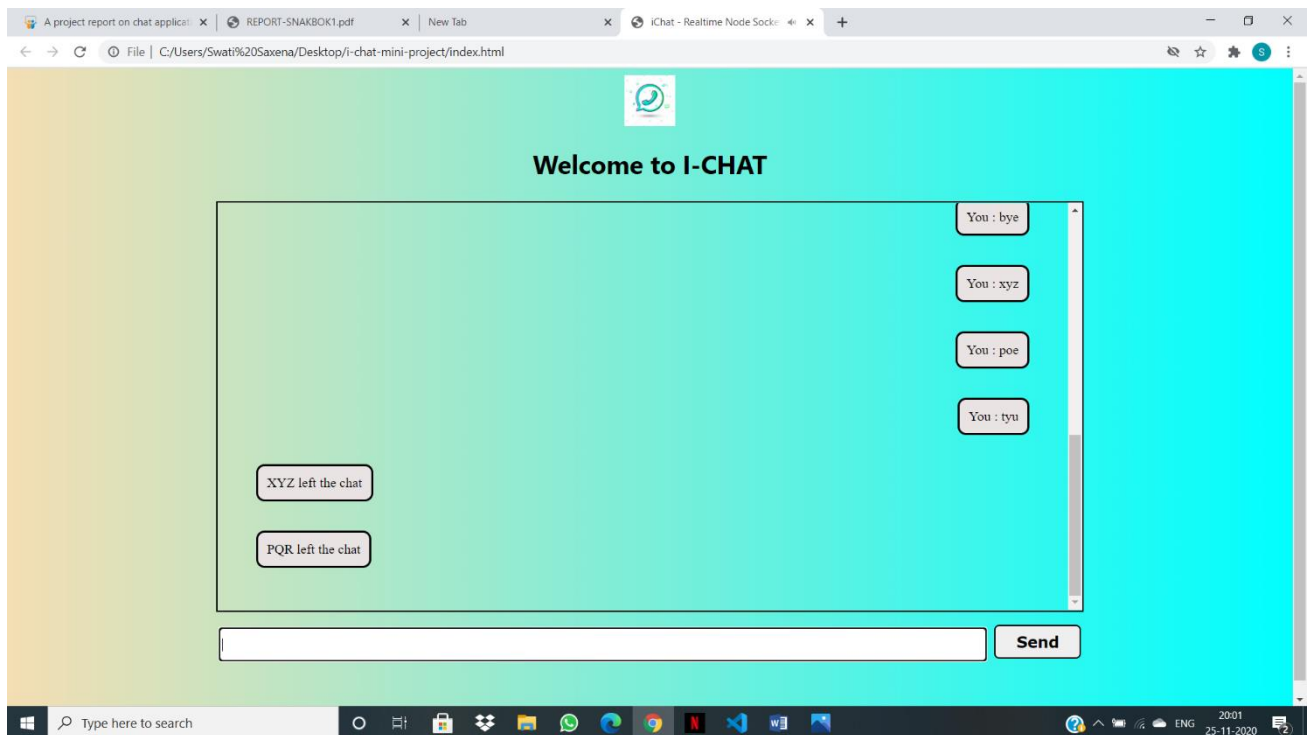
By chance any user left the chat then it would notify anothers users present in the chat that so & so left the chat with notification sound.



Likewise to the another user present.



And,



Some of the features are:-

1. Scrolling bar at the right:-

Whenever there are a lot of messages on the screen, to see them all, a scrolling bar on the right-hand side will appear on its own.

2. Lots of designing by using CSS:-

We will be using a nice lot of RGB colors, and many more CSS to enhance our webpage.

3. Message notification sound:-

With every message that arrives, there is a notification alert with a minute ringtone noise.

### **Future Extension of I-chat**

In the extension of the “i-chat”, we would be adding some modules into it.

Some of the modules are enlisted below:

- **Emoji:** As we all know in today’s world emotions play a vital role, but virtually we cannot show or present our felling. So to nullify these cons, emoji is there so that people can present their expression by texting some emoji. So we will be providing a module for emoji/emotions in the future.
- **Grammarly:** In this era, “**Formal Chatting**” is also an important aspect, which requires proper English and vocabulary. So the module for the proper vocab and grammar would be included in the extension of the project.

- **Different languages:** By default, the chatting language would be English. But if the user wants to text in other language like “French”, “Hindi”, “Spanish” etc, they can do with the help of some imports.
- **A box of users:** We would be adding a box in which the name of users chatting in that server at a time with simple HTML and an amount of CSS to enhance it.  
And many more.

## **References**

- <https://www.w3schools.com/js/DEFAULT.asp>
- <https://nodejs.dev/learn>
- <https://developer.mozilla.org/en-US/docs/Learn/CSS>
- <https://www.javatpoint.com/html-tutorial>