

FULL STACK – II

PROJECT REPORT (2020-2021)

ON

“We-CHAT”

Department of Computer Engineering & Application
Institute of Engineering & Technology



GLA University,
Mathura- 281406

Submitted To:
Mr. Pankaj Kapoor

Submitted By:
Jaideep Lalchandani
Mansi Goyal
Nidhi Gupta
Prashant Tomar
Radhika Singh



Department of Computer Engineering and Applications

GLA University, Mathura

17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,
Mathura – 281406

DECLARATION

I/we hereby declare that the work which is being presented in the Bachelor of technology. Project “**We-CHAT**”, in partial fulfilment of the requirements for the award of the ***Bachelor of Technology*** in Computer Science and Engineering and submitted to the Department of Computer Engineering and Applications of GLA University, Mathura, is an authentic record of my/our own work carried under the supervision of **Mr. Pankaj Kapoor, Technical Trainer, Dept. of CEA, GLA University.**

The contents of this project report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree.

Jaideep Lalchandani (181500290)

Signature of Candidate:

Mansi Goyal (181500370)

Signature of Candidate:

Nidhi Gupta (181500422)

Signature of Candidate:

Prashant Tomar (181400492)

Signature of Candidate:

Radhika Singh (181500529)

Signature of Candidate:

Course: B.Tech. (Computer Science & Engineering)

Year: 3rd

Semester: VI



Department of Computer Engineering and Applications

GLA University, Mathura

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,
Mathura – 281406**

CERTIFICATE

This is to certify that the project entitled “**We-CHAT**” carried out in Full Stack-II Project is the work done by Jaideep Lalchandani, Mansi Goyal, Nidhi Gupta, Prashant Tomar and Radhika Singh, and is submitted in partial full fulfilment of the requirements for the award of degree Bachelor of Technology (Computer Science and Engineering).

Name of Supervisor:

Mr. Pankaj Kapoor

Date:

Signature of Supervisor:



Department of Computer Engineering and Applications

GLA University, Mathura

**17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,
Mathura – 281406**

ACKNOWLEDGEMENT

We take this opportunity to thank all those who have helped us in completing the project successfully.

We would like to express our gratitude to Mr. Pankaj Kapoor, who as our guide/mentor provided us with every possible support and guidance throughout the development of project. This project would never have been completed without his constant encouragement and support.

Our heartiest thanks to Dr. (Prof). **Anand Singh Jalal**, Head of Dept., Department of CEA for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project. After doing this project We can confidently say that this experience has not only enriched us with technical knowledge but also has unparsed the maturity of thought and vision, the attributes required for being aprofessional.



Department of Computer Engineering and Applications

GLA University, Mathura

17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,
Mathura – 281406

ABSTRACT

We are creating “A Web-browser based application” named “**We-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.

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 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)



Department of Computer Engineering and Applications

GLA University, Mathura

17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,
Mathura – 281406

CONTENT

Declaration	2
Certificate	3
Acknowledgment	4
Abstract	5
1. Introduction	7
Overview	7
Motivation	7
2. Problem Statement	8
What socket io is?	8
What css is?	8
What is html?	9
What is javascript?	9
What node.js is?	9
3. Area of computer science	10
4. Hardware requirement	11
5. Software requirements	11
6. Methodology	12
7. Implementation details	13
8. Appendices	15
Future extension	37
References	38

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 We-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?

- Node.js was developed by many developers but the original author was Ryan Dahl in 2009.
- Node.js 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
- Crome
- Opera

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-

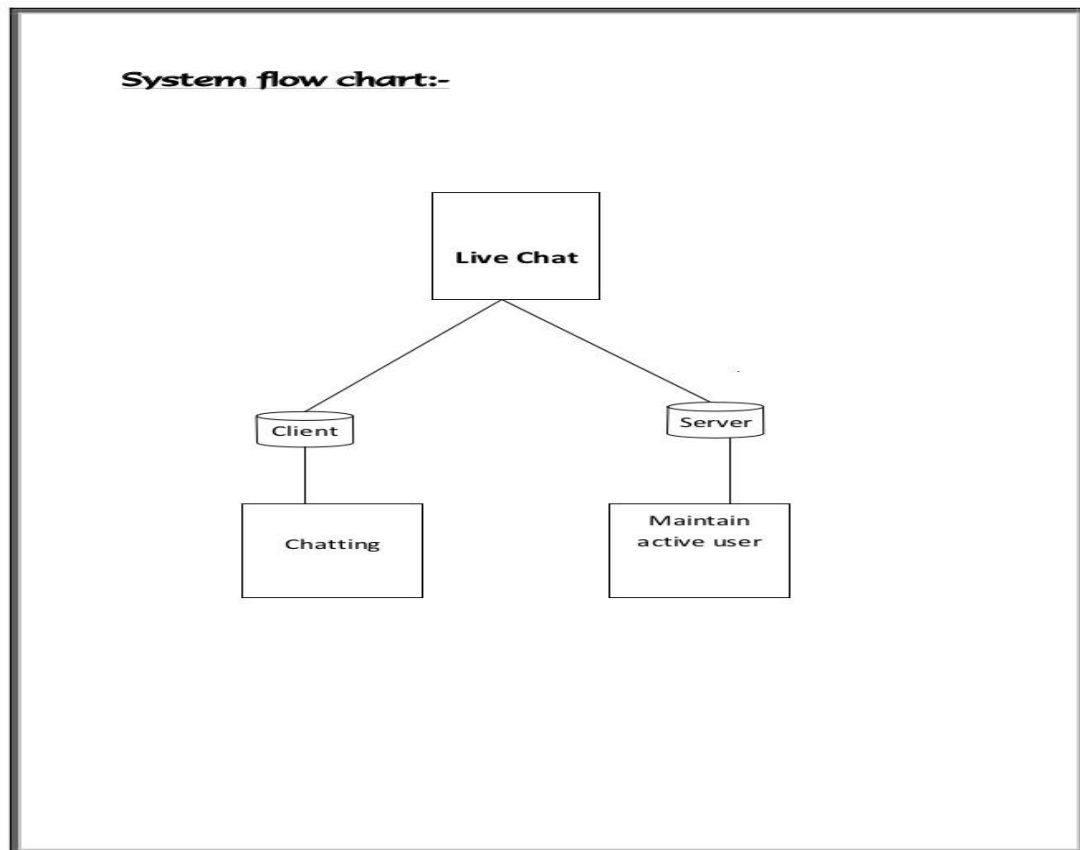
Operating system- windows7

Others- visual studio

Methodology

Our methodology is designed to help you take maximum advantage of the internet technologies. It toincorporate 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.

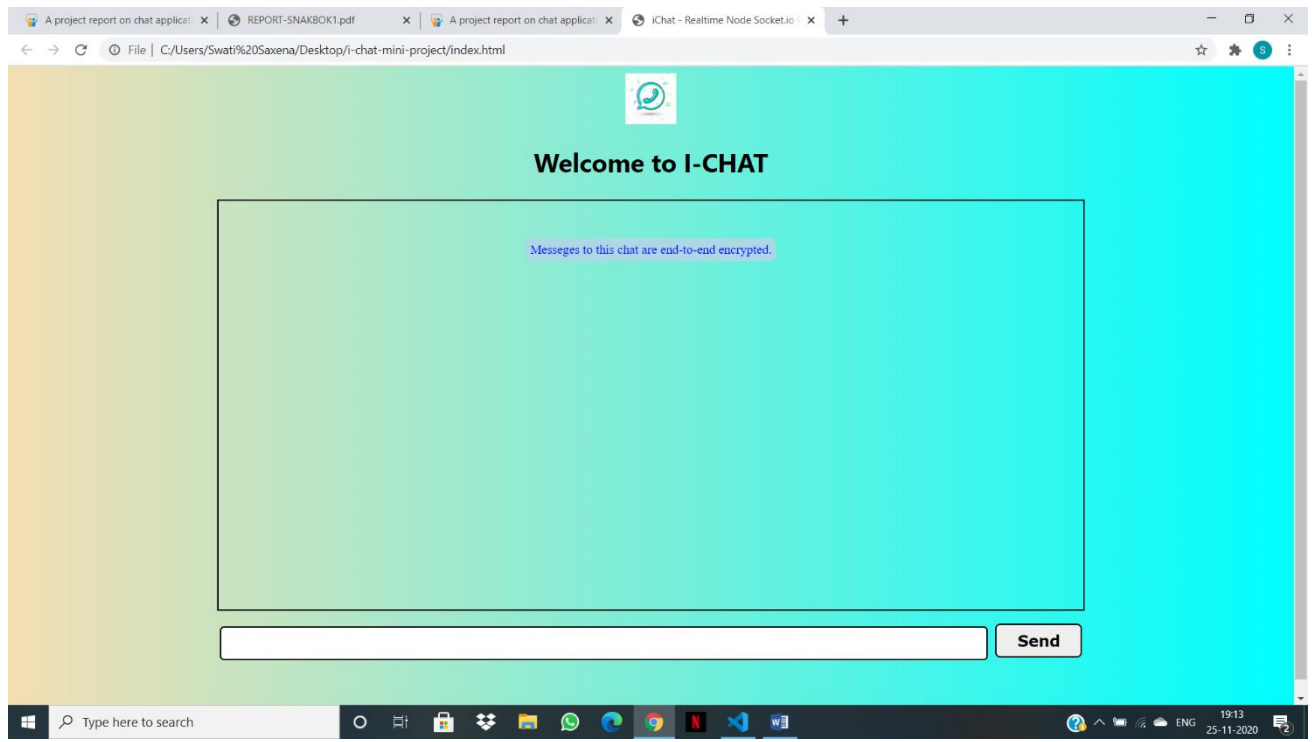
Part 3:- Final Submission

- 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 App</title>
    <script defer src="http://localhost:4000/
socket.io/socket.io.js"></script>
    <script defer src="js/client.js"></script
>
    <link rel="stylesheet" href="css/style.cs
s">
</head>

<body>
    <nav>

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

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

    <div class="send">

        <form action="#" id="send-
container">
            <input type="text" name="messa
geInp" 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 App</title>
  <script defer src="http://localhost:4000/socket.io/socket.io.js"></script>
  <script defer src="js/client.js"></script>
</head>
<body>
  <nav>

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

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

```

    </div>

    <div class="send">

        <form action="#" id="send-
container">
            <input type="text" name="messa
geInp" id="messageInp">
            <button class="btn" type="subm
it">Send</button>

        </form>
    </div>

</body>
</html>

```

Client.js:

```

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

const form = document.getElementById('send-
container');
const messageInput = document.getElementById(
'messageInp')
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')
    })
```

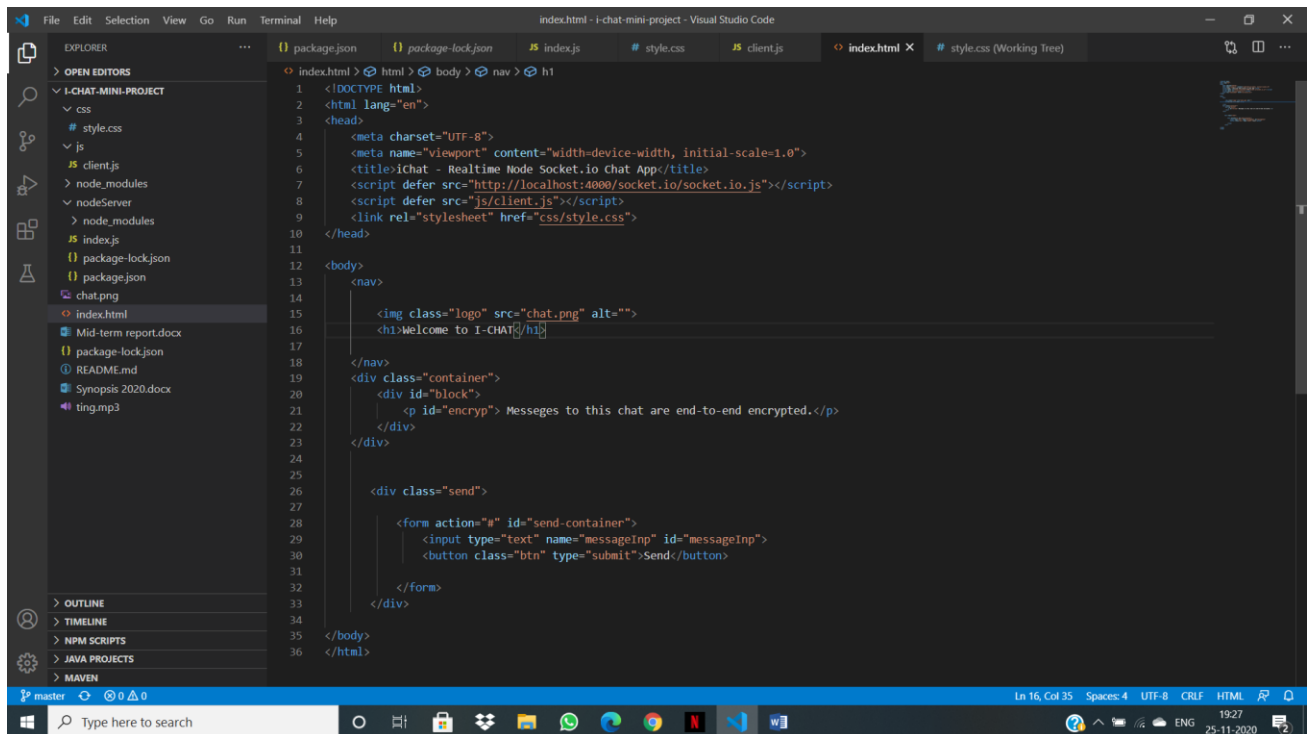
Back-end:

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

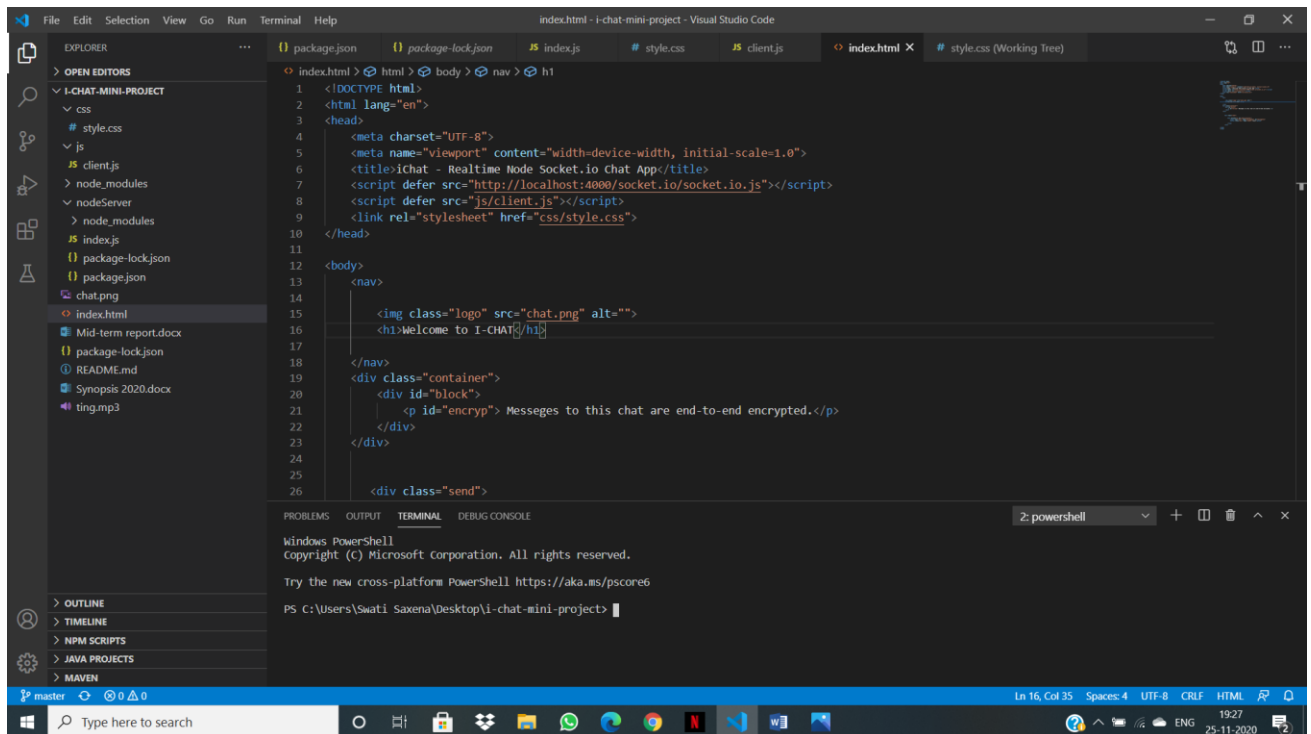
Appendix 2

Output:

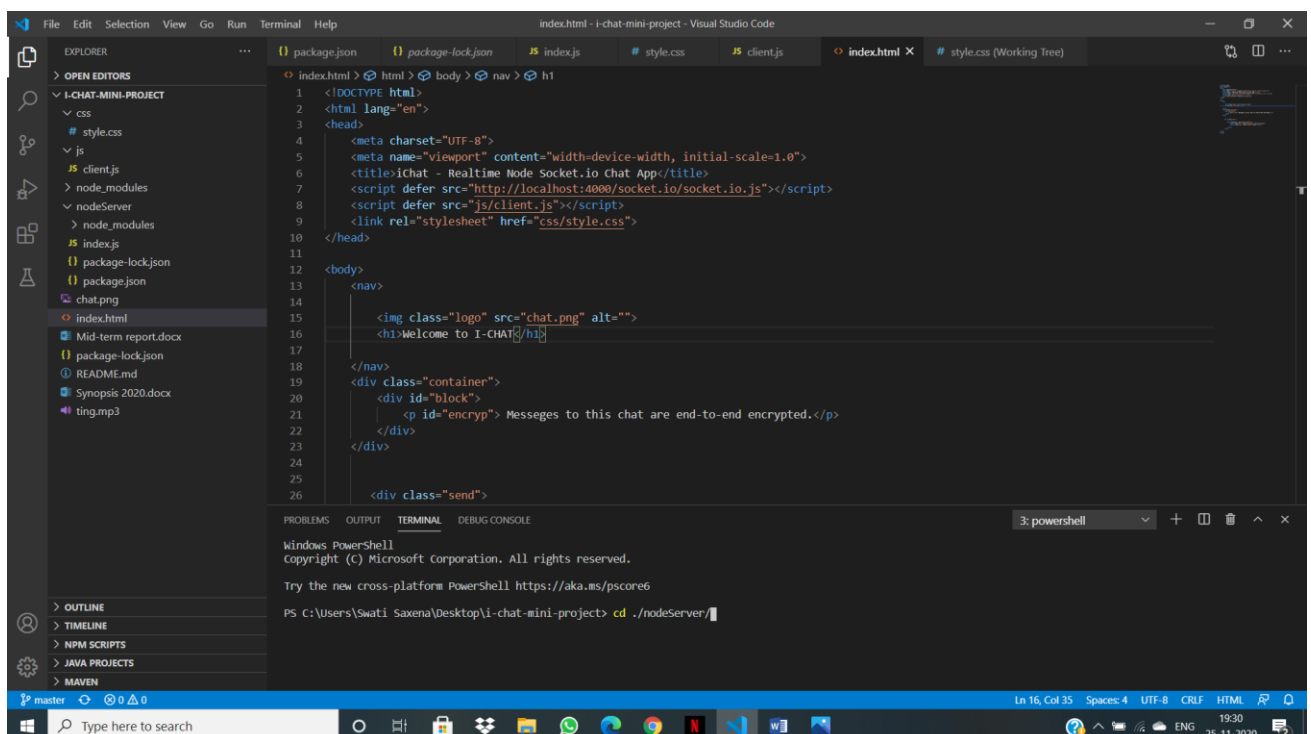
Firstly the node server will be run on vs code's terminal.so that our project should run.



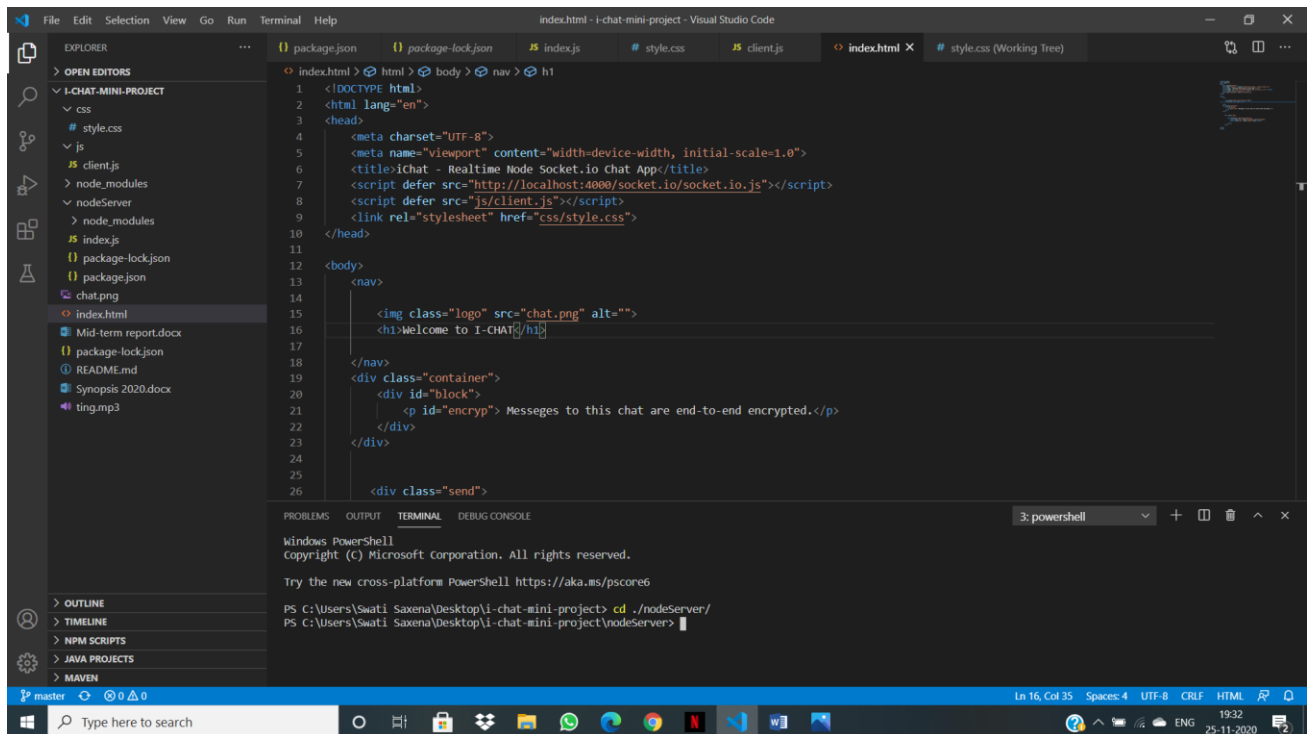
Click on terminal >> new terminal



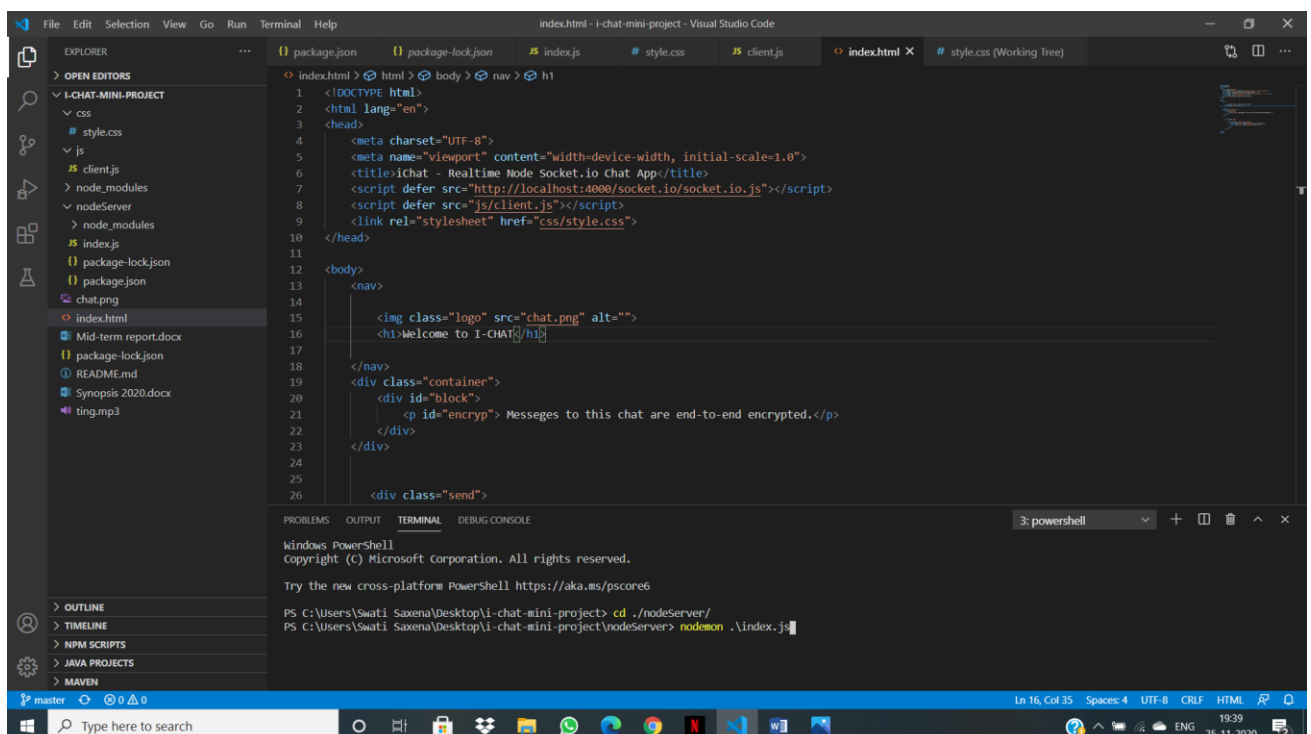
Then write the command `cd ./nodeServer/`



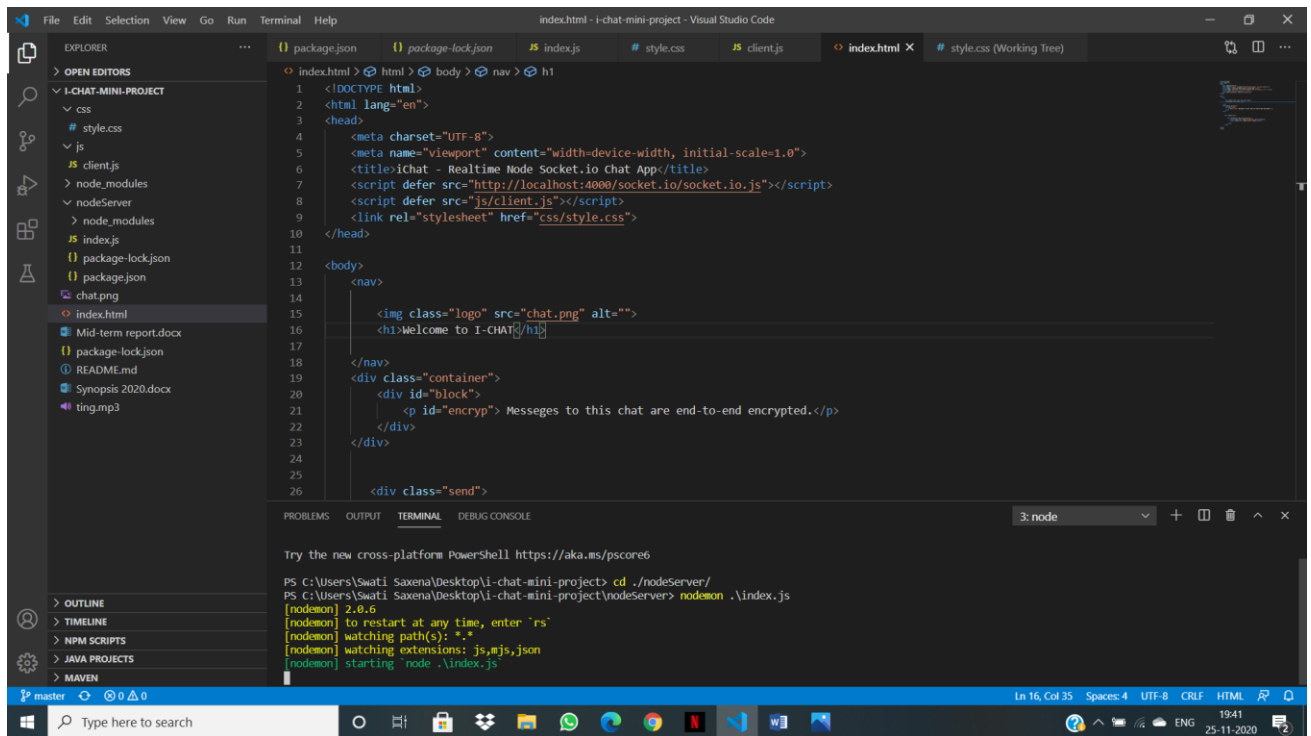
Then give the path where you want to connect this node server.



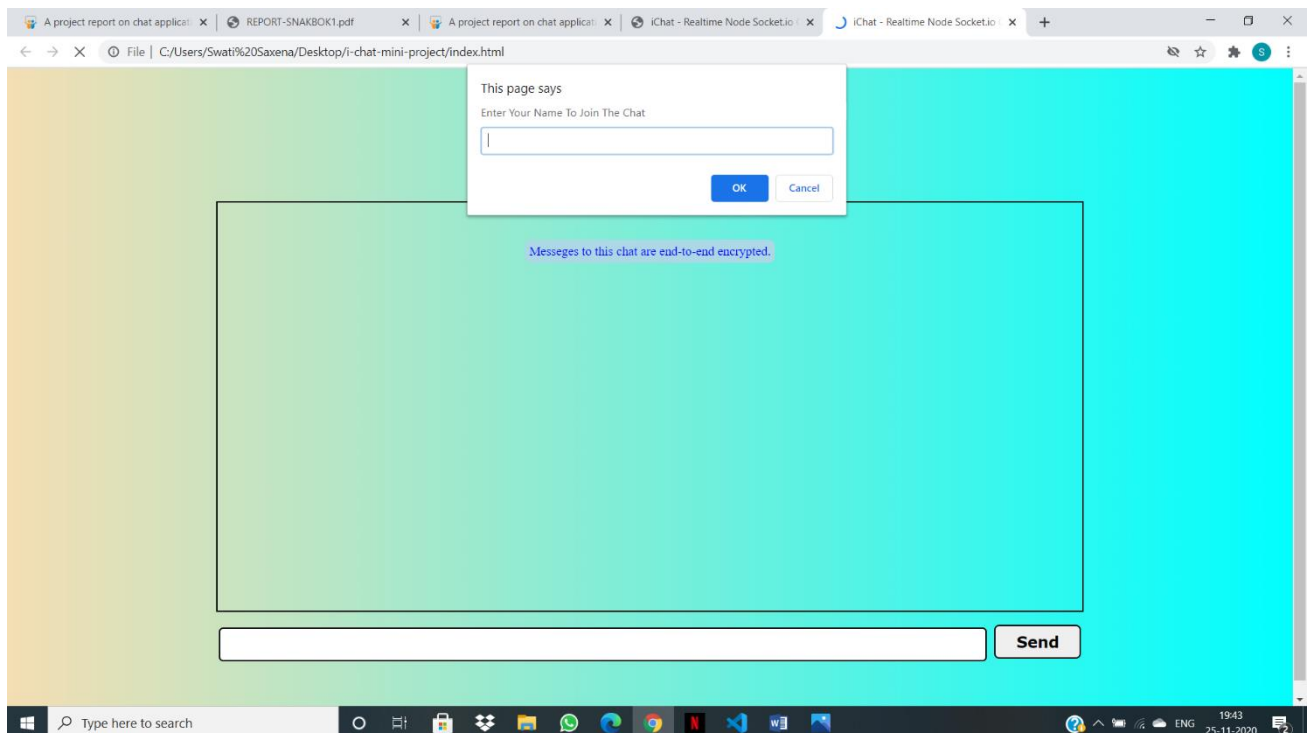
Write command nodemon ./index.js



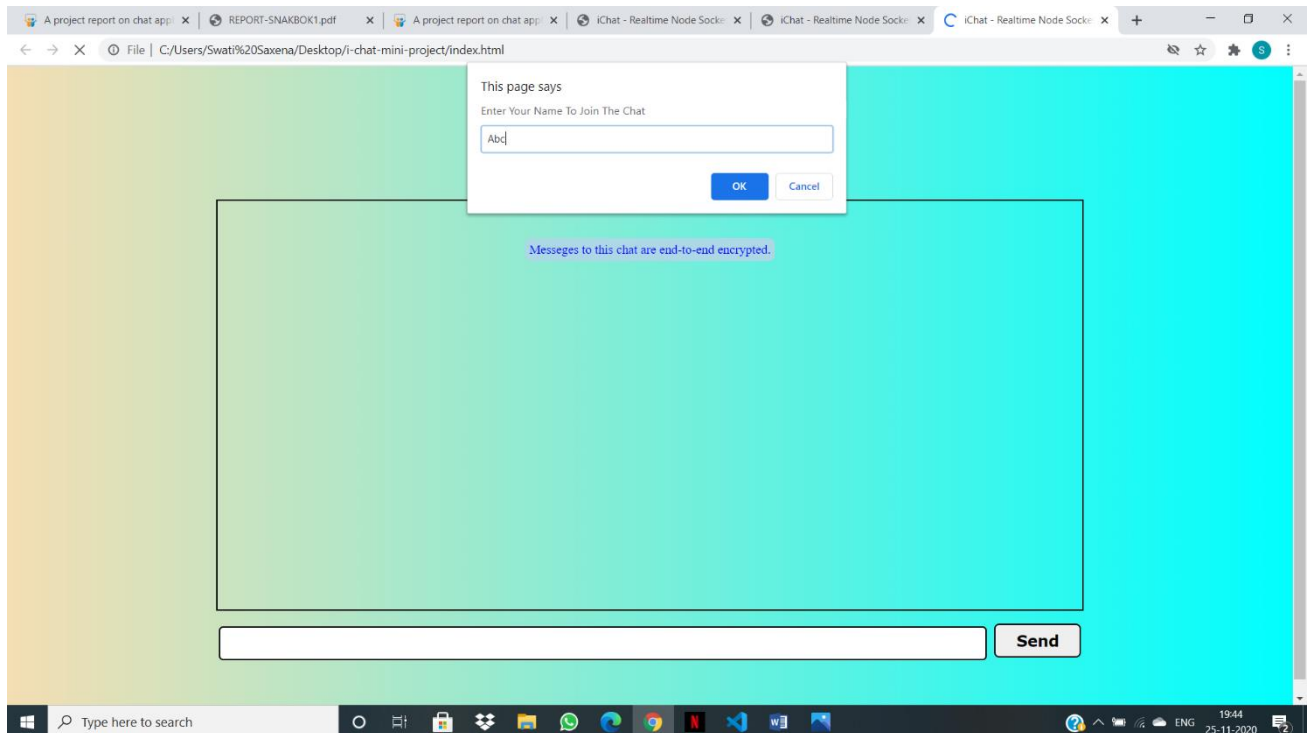
So after writing the command nodemon ./index.js. The nodeServer gets connected to the Index.js.



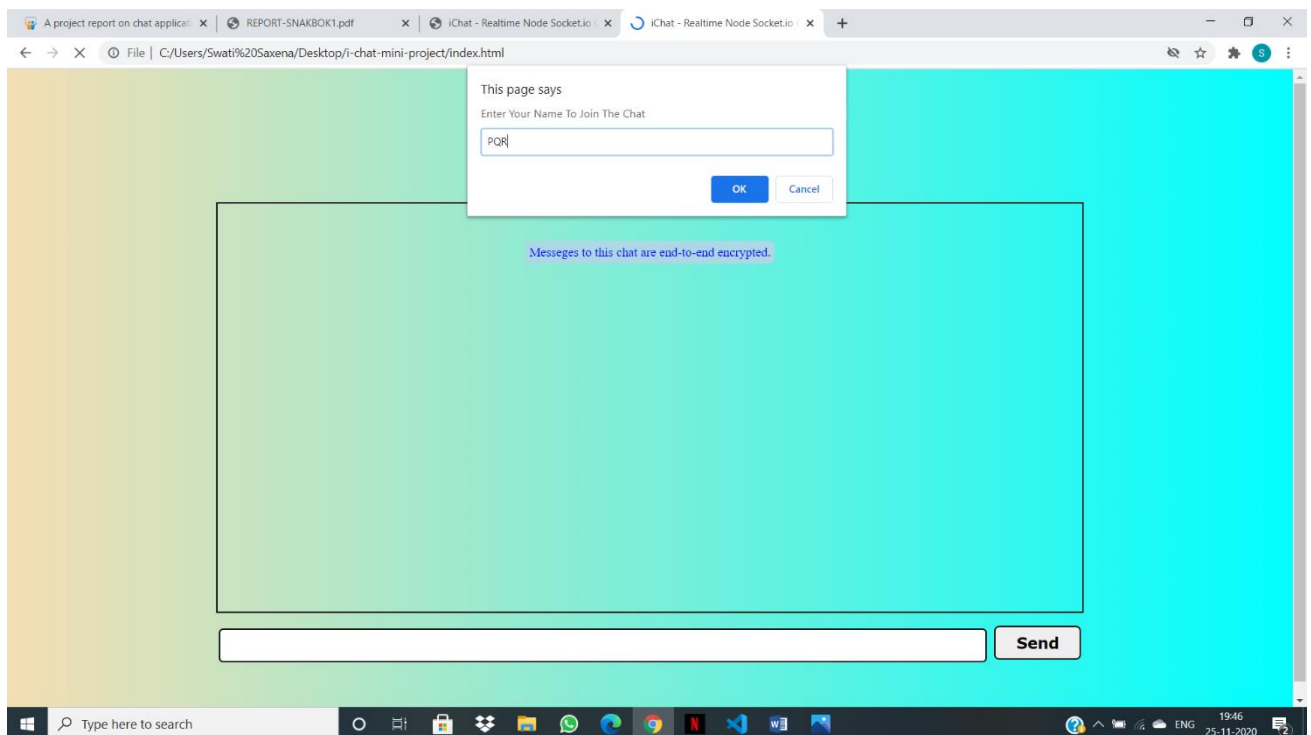
Then we will start the live Server it will look like this:



Here user will enter the name to join the chat.

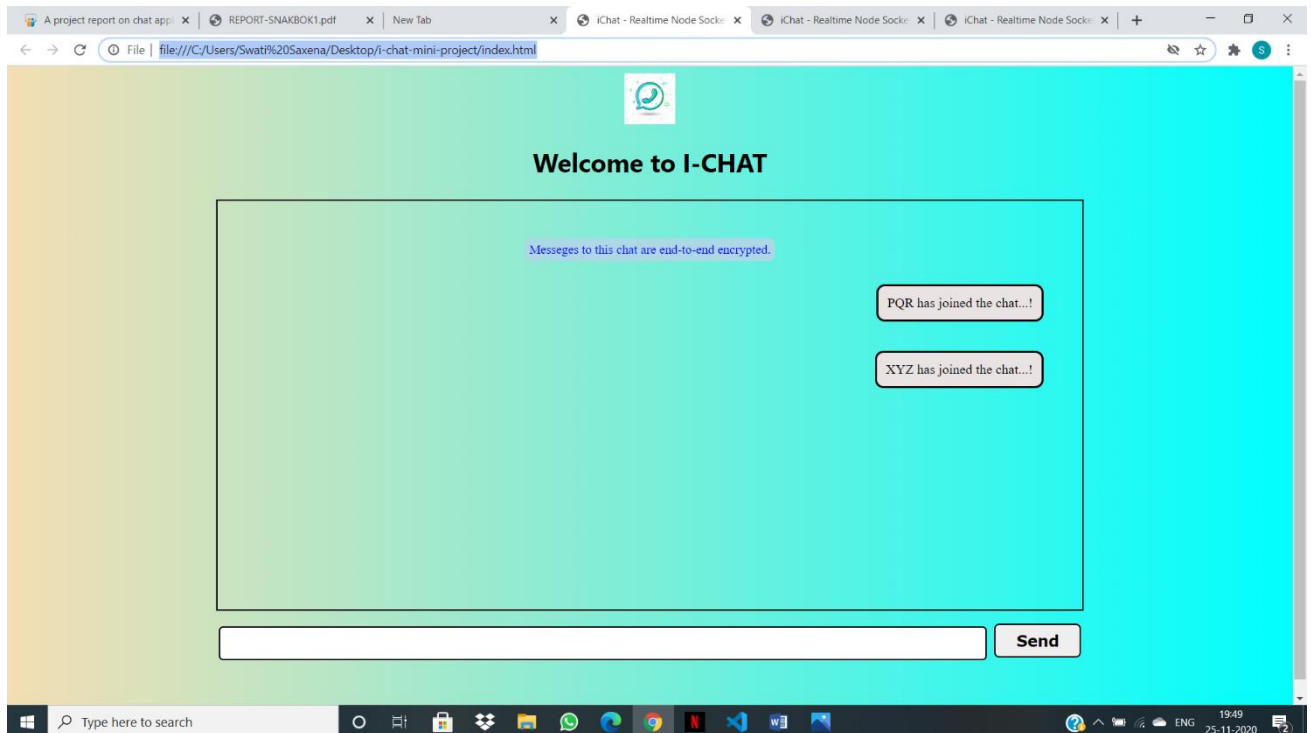


Then let the another user joins the chat.

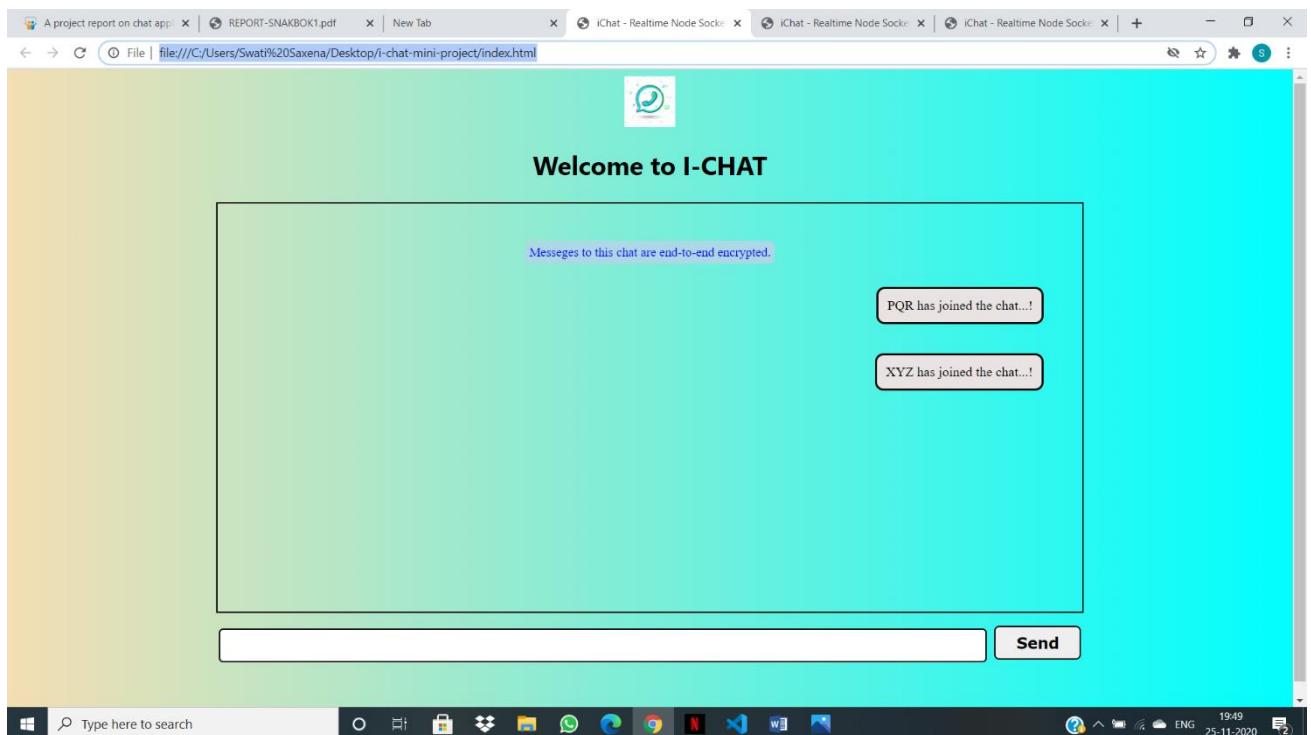


Then the notification to the users apart from the user joined will be send with the help of socket.io.

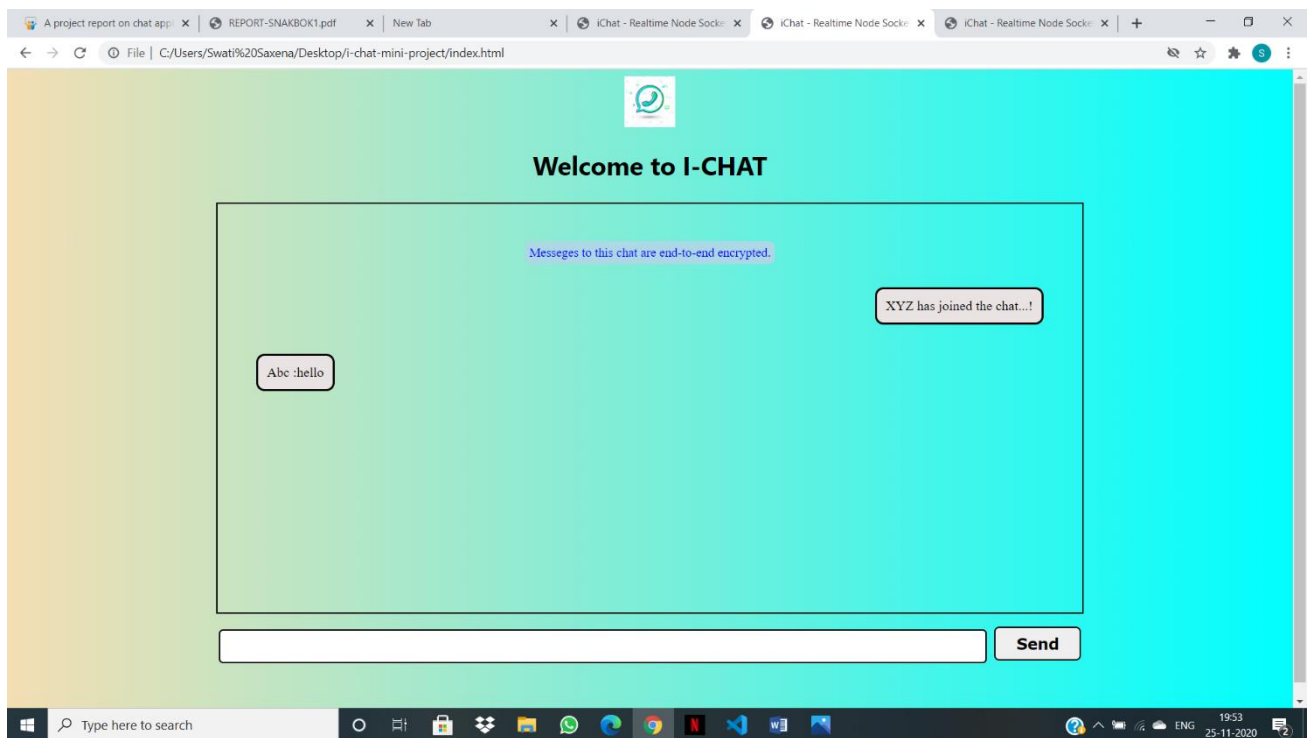
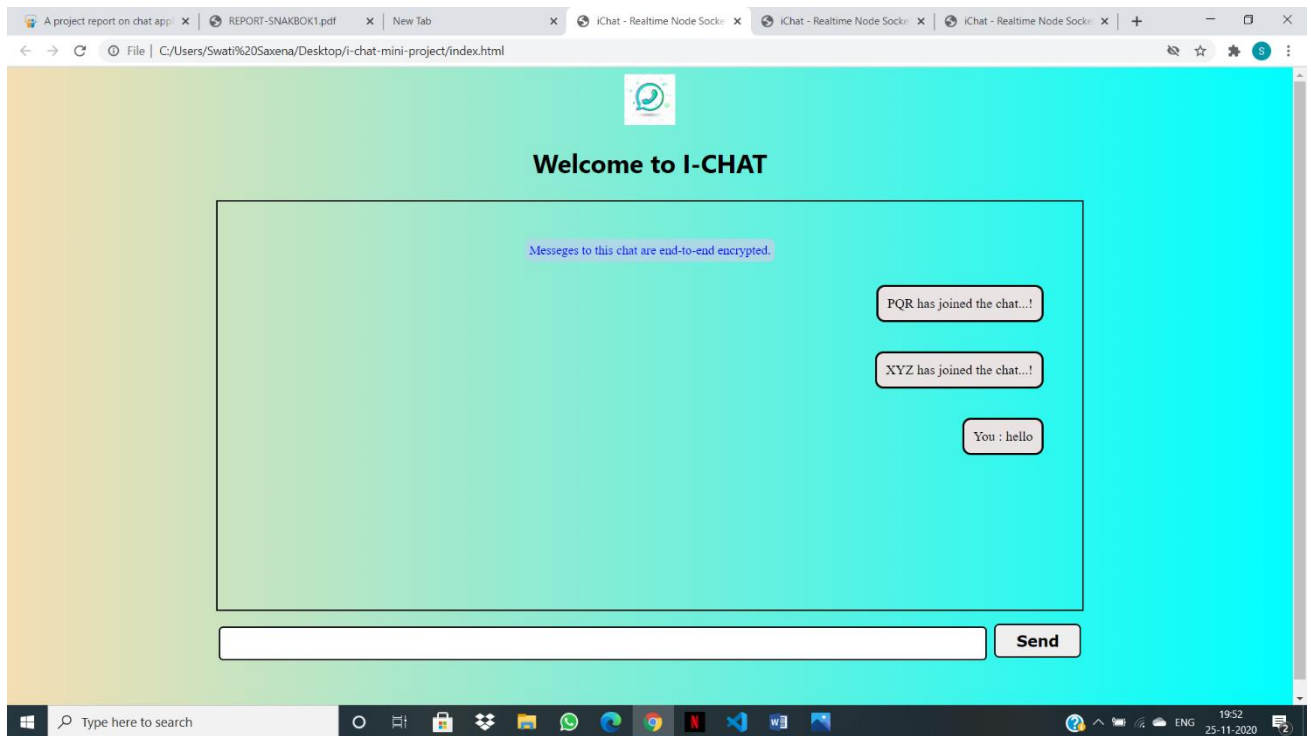
Notification be like XYZ has joined the chat...!.



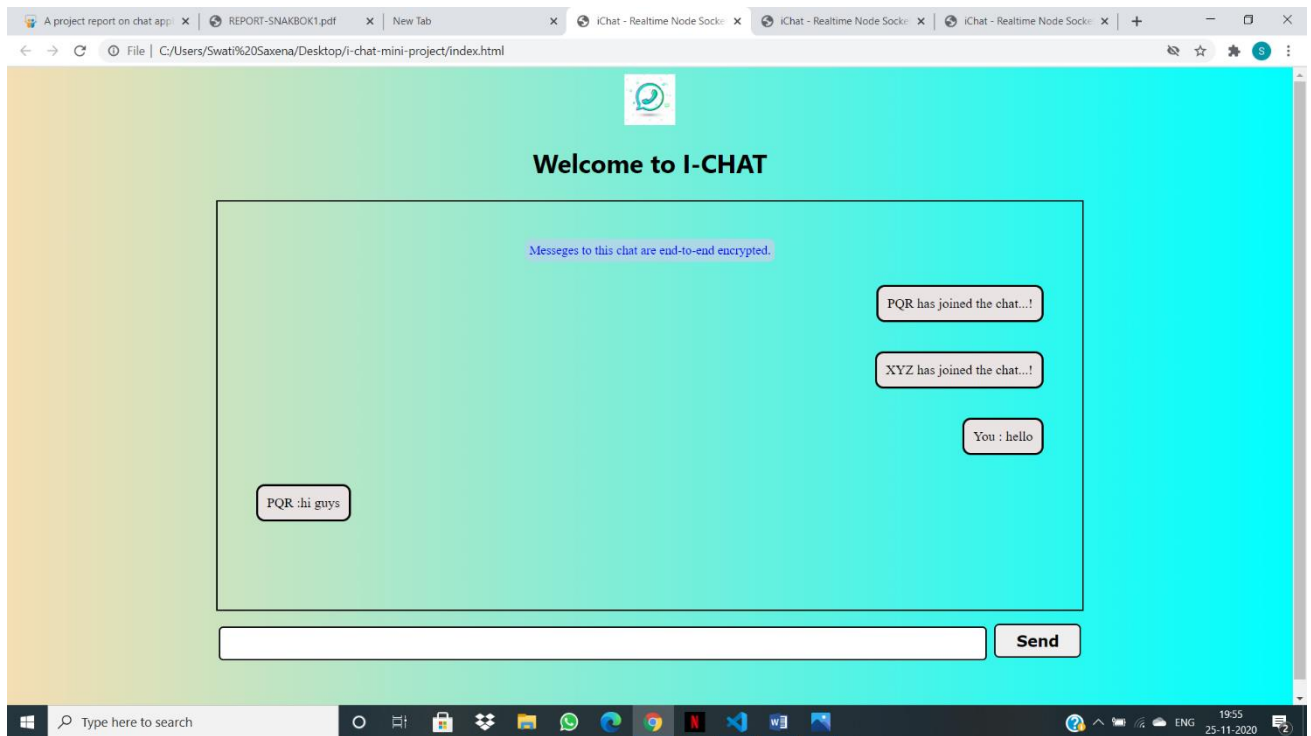
Likewise,



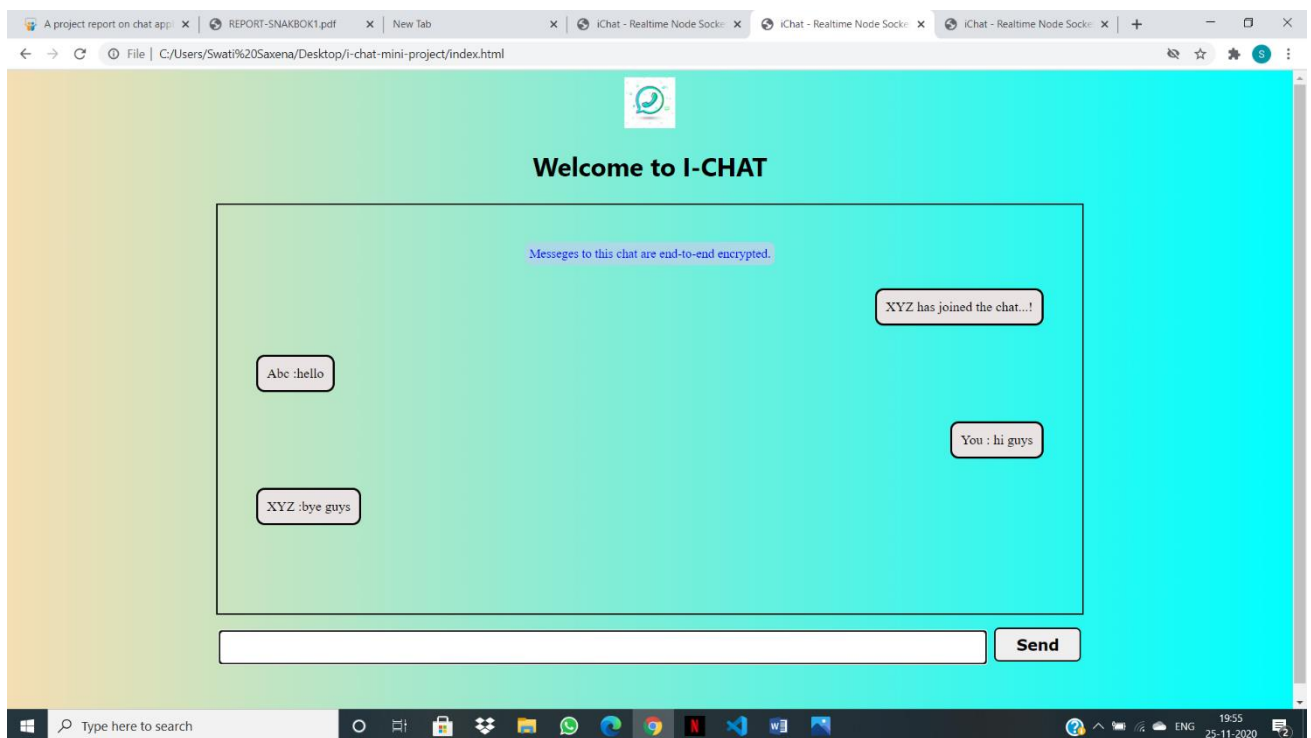
Then any user types the message then it would reflect to another windows too.



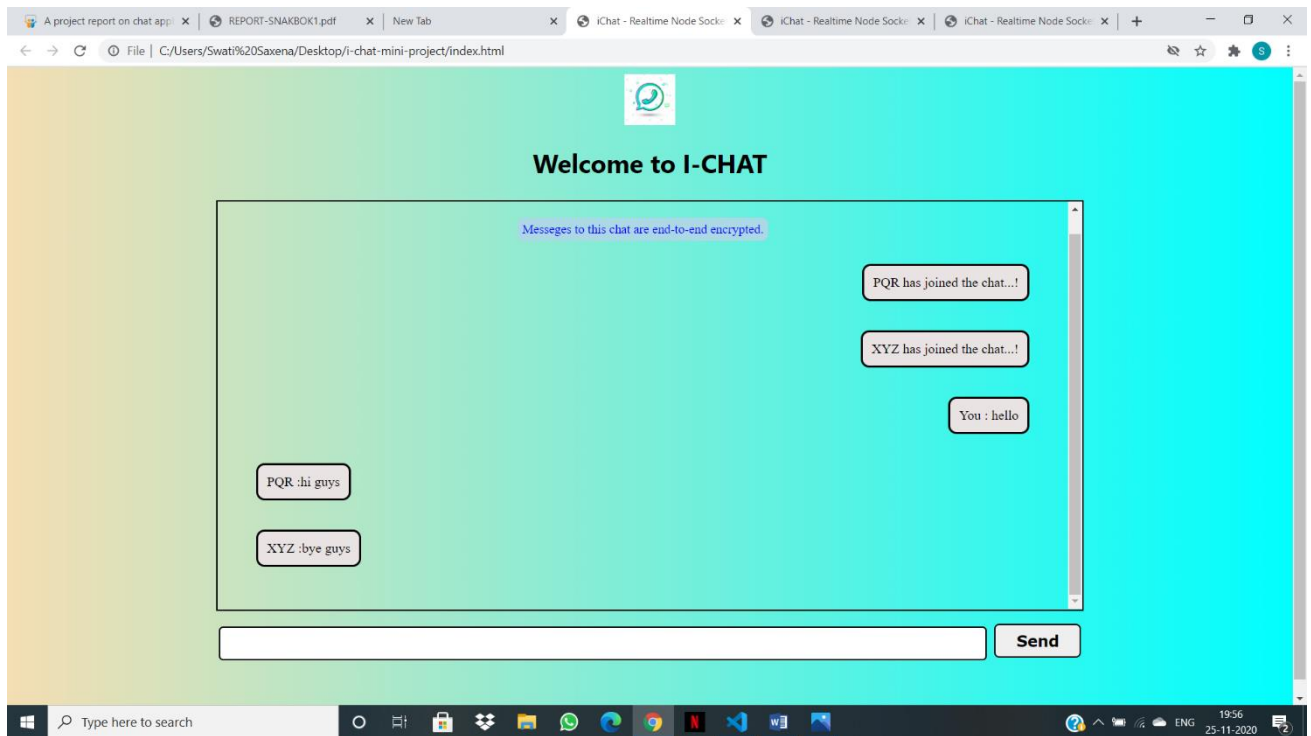
The message will receive with a notification sound. Offcourse that cant be shown.



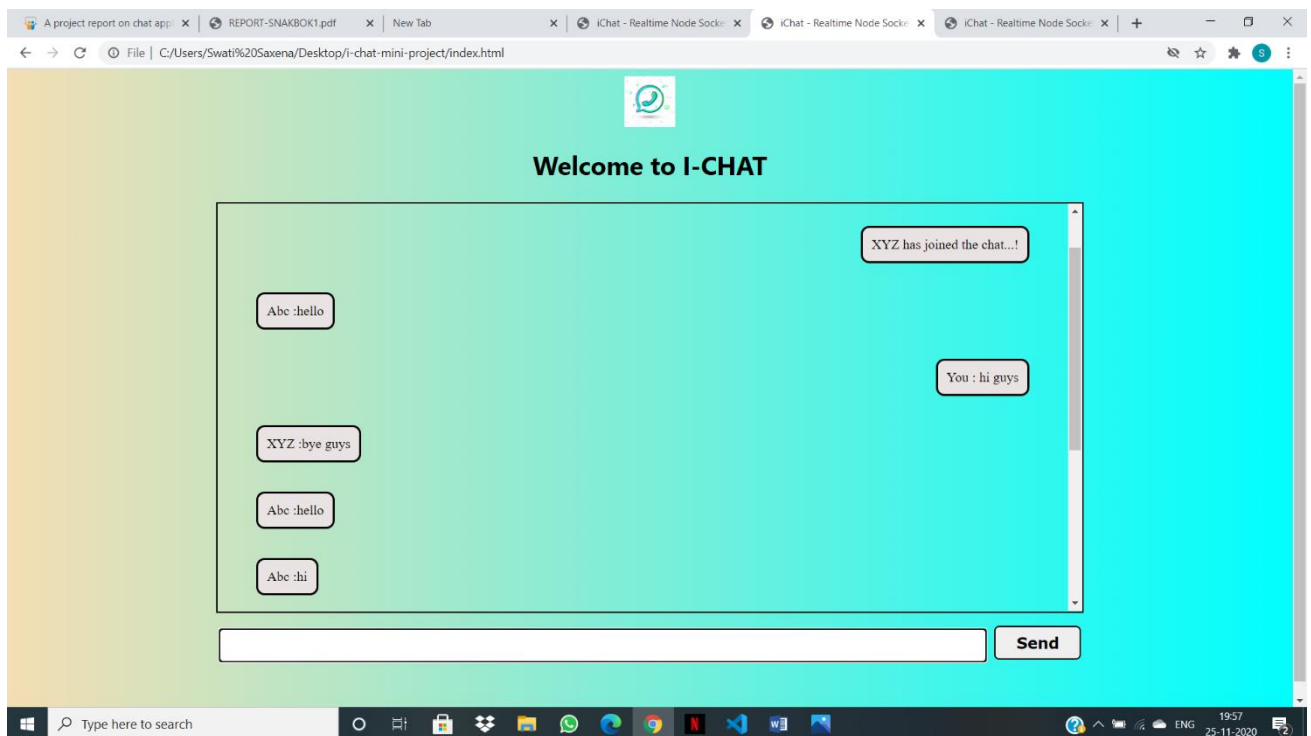
Likewise,



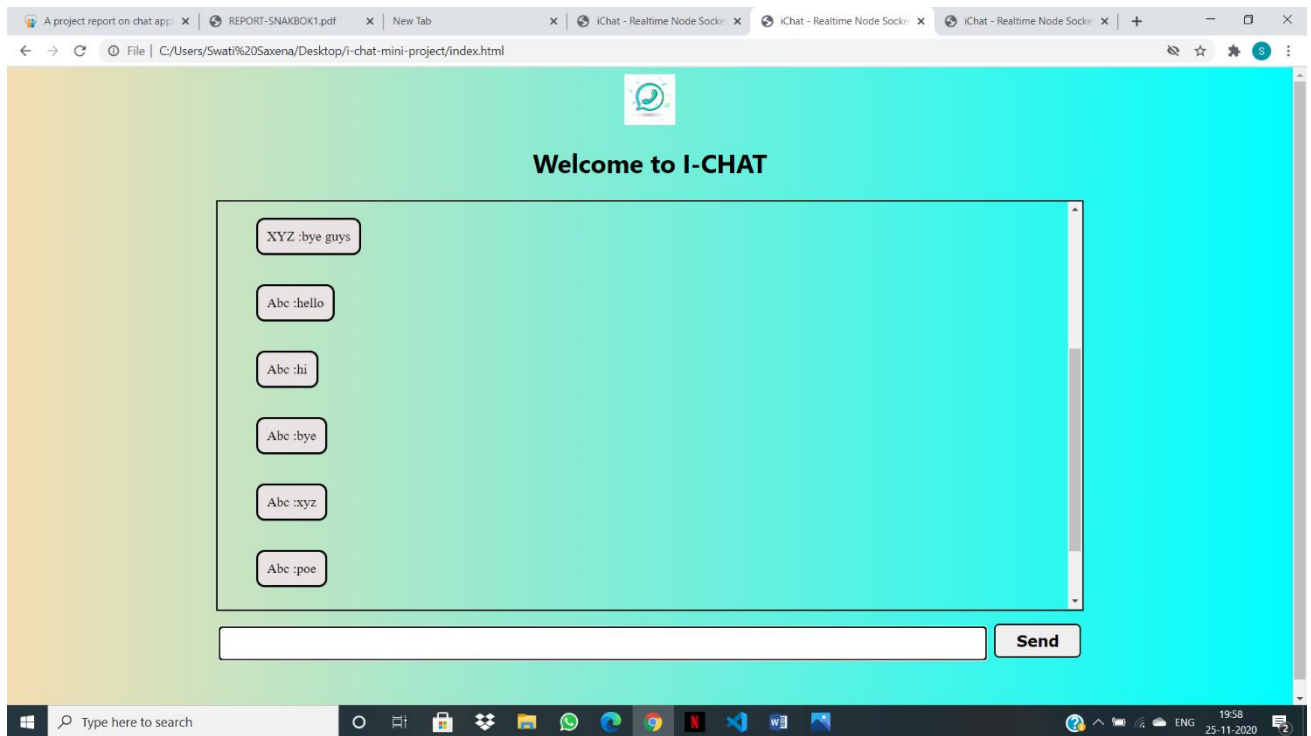
And another



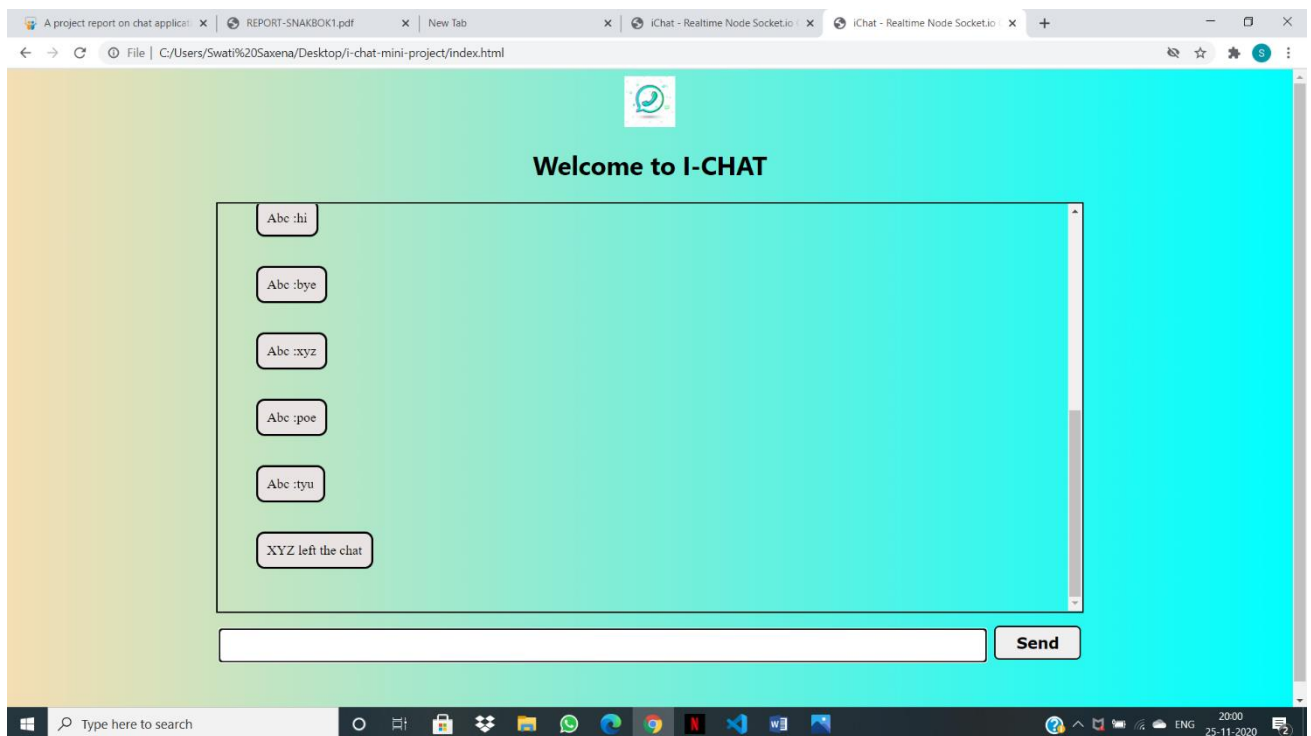
If message gets overflow then to manage all of these. There is a scrolling bar on right.



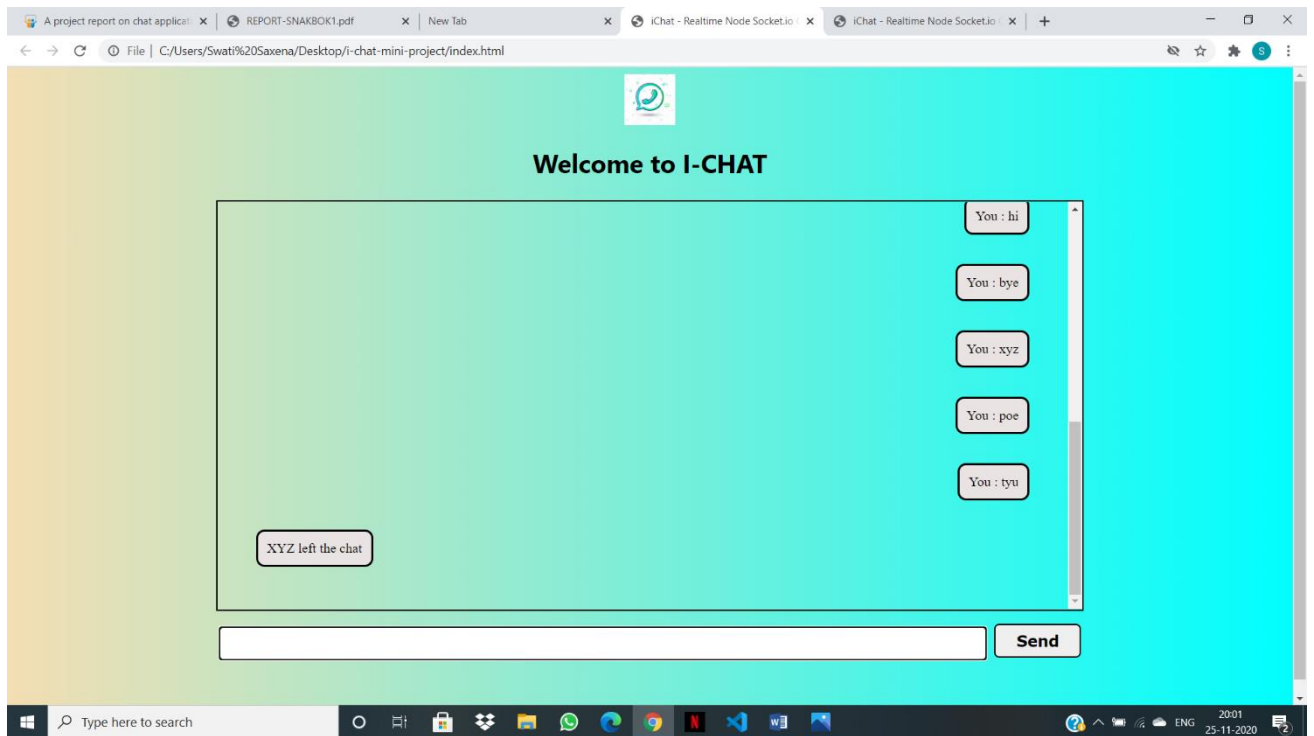
Likewise,



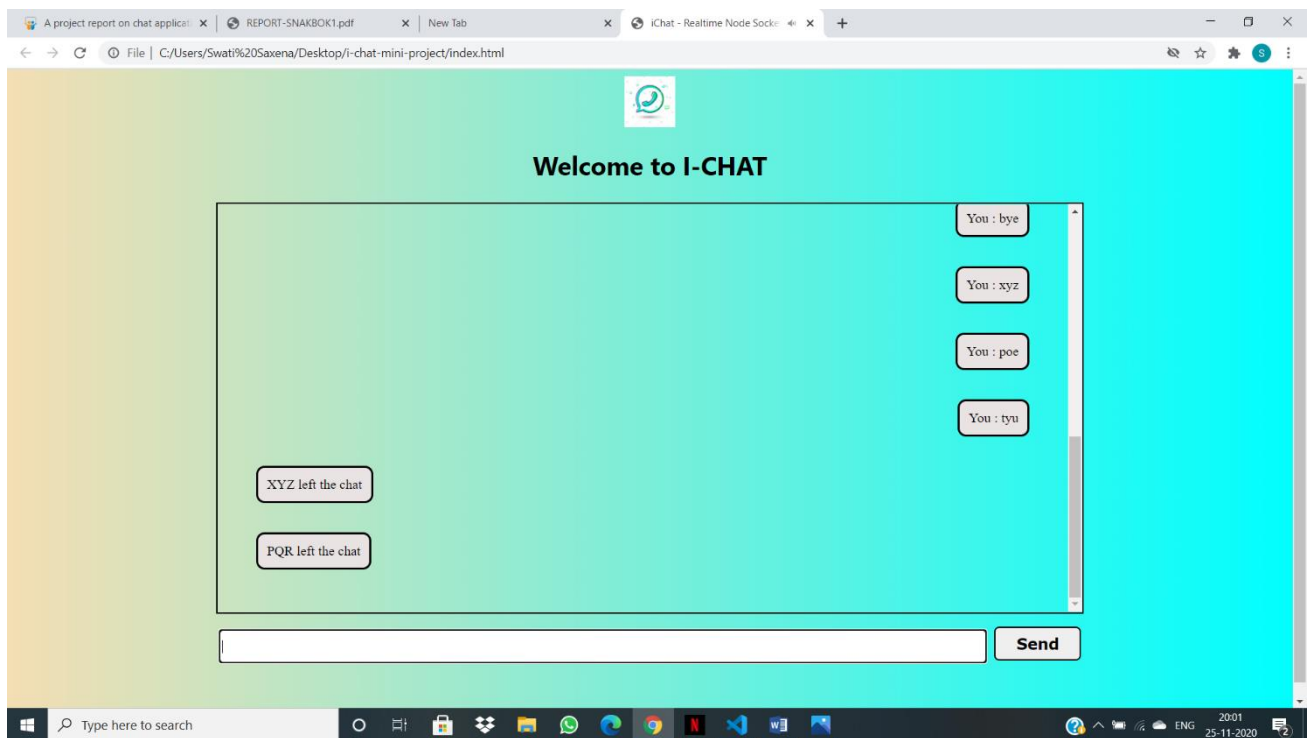
By chance any user left the chat then it would notify another's users present in the chat that so & so left the chat with notification sound.



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 We-chat

In the extension of the “We-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 feeling. 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>