Abhinav Paudyal

Developing Video Chat Application with ReactJs And WebRTC

Metropolia University of Applied Sciences

Bachelor of Engineering

Name of the Degree Programme

Bachelor's Thesis

15 April 2021

Abstract

Author:	Abhinav Paudyal
Title:	Developing video chat application with ReactJs and WebRTC
Number of Pages:	34 pages
Date:	15 April 2021
Degree:	Bachelor of Engineering
Degree Programme:	Information Technology
Professional Major:	IoT and Cloud Computing
Supervisors:	Janne Salonen, Head of Department ICT

As you know due to corona virus all the social interaction has moved online, that has triggered a remarkable growth in video sharing services such as Zoom, Skype, Discord, and many others. As we know human beings are social animals and communication with one another is vital in our lifestyle. But since most people, for past 2 years have been using these video chatting app the data flow for these applications has not been at its best. After encountering interruptions when using these video chatting apps to communicate with friends and relatives, the concept for this project was born. This project was created to allow friends and family to communicate freely. This thesis will focus on how to build an app that allows you to chat visually with your friends and colleagues.
This project was completed with a fully functional Chat App. The chat App was reviewed in a variety of web browsers and yielded the predicted performance.

Keywords: Chat App, React Js, HTML, CSS , Node Js, Material UI

Contents

List of Abbreviations

HTML:	Hypertext markup language
CSS:	Cascading style sheets
DOM:	Document object model
ID:	Identification
JS:	Java Script
JSON:	Javascript object notation
JSX:	Javascript XML
NPM:	Node Package Manager
UI:	User interface
XML:	Extensible markup language
GUI:	Graphical user interface
API:	Application program interface

1 Introduction

Currently communication plays a vital role in our life. As we know due to covid for past two years all the social life has moved online, because of this online video calling app such as Zoom, Discord, Skype has had a rise in its use for school, work and to communicate with family and close people. As the uses of these app had a rise in its use, I experienced that the services they provide were not always functioning well. The main idea behind this project was to have a simple website that close friends and family can use to talk with each other without worrying about another web chatting app not working.

The purpose of this project was to create a simple web chatting application using WebRTC and React. This thesis will discuss about what kind of software I used while creating this web application, technology used, and how I implemented all those software and technology in this project, also end results on how the web application functions, and conclusion.

For the purposes of determining the project's complexity, the project would be reviewed simultaneously with the development of the application.
A directory for authorized users would be built and reviewed, as well as a menu, a client/server configuration, and graphical user interfaces will be built and tested with the advantage of the consumers in mind. When the chat program is close to completion, further testing may be performed to ensure that it is less buggy and more user oriented.

The targeted audience are my family and friends for easier communication between each other.

2 JavaScript

JavaScript is an interpreted, lightweight programming language. It is intended to use in the development of network-centric software. It works in conjunction with Java and is also compatible with Java. It is extremely simple to incorporate due to its integration with HTML. It is cross-platform and open source.

JavaScript has been referred to as a scripting language since 1995. However, its applications in web creation are numerous nowadays. NodeJS sparked the development of various JS frameworks and libraries by web professionals.

JS was initially developed as an in-browser scripting language with the aim of adding interactivity to a static web page. However, it has recently expanded beyond its initial boundaries. A variety of JavaScript frameworks have appeared in a comparatively short amount of time, enabling programmers to create web apps of greater sophistication and quality in less time. Additionally, the programming language has gained attention with non-frontend developers. A growing number of app developers are beginning to understand its utility in desktop and mobile growth. As a result, businesses in the information technology and web creation industries also have a broader range of opportunities for designing various types of applications. One of them is designing software exclusively in JS for various platforms. It is not possible to list all the JavaScript frameworks and libraries available as JavaScript's universe is too wide and too dynamic. Some useful JavaScript frameworks are Angular, React, jQuery, Vue.js, Ext.js, Ember.js, Meteor, Mithril, Node.js, Polymer, Aurelia, Backbone.js etc.
[1]

2.1 Brief history on JavaScript

Brendan Eich developed JavaScript in 1995 working at Netscape Communications Corporation, the company that produced the venerable Netscape Navigator web browser. Java was gaining popularity at the time, and

Netscape Communications attempted to integrate it into Netscape Communicator.

Mocha was the name given to the early implementation of JavaScript. It was renamed LiveScript not long after a version of mocha was integrated into Netscape communicator. It was changed to LiveScript in May 1995. In December of the same year, it was renamed again, this time to JavaScript. This initial iteration of JavaScript described many of the great characteristics (such as its object-model) for which this coding language is now famous. Additionally, it possessed plenty of advanced features that allowed it to gradually outgrow its original intent. These characteristics include the following:

- First-class object functions
- Similar Syntax to Java
- Prototype-based object model [2]

2.2 Pros and Cons of Java Script.

Majority of developers believe JavaScript to be the most common programming language, according to the 2019 Stack Overflow Survey. This describes its widespread usage by developers working in a variety of product domains. Although it is no secret that JavaScript is a particularly preferred among most developers, there are certain advantages and disadvantages to JavaScript. These JavaScript benefits and drawbacks are crucial when choosing JavaScript. Without diminishing the value and utility of JavaScript, it is important to recognize there can be instances where a different language is a better fit.

A side-by-side comparison of the advantages and disadvantages of JavaScript is undoubtedly the perfect way to determine if this is the correct programming language for your next project. [3]

2.2.1 Advantages of JavaScript

- Despite of where you host JavaScript, it is often run on the client side to save bandwidth and speed up the production phase.
- XMLHttpRequest is a critical entity in JavaScript that was created by Microsoft. The entity call provided by XMLHttpRequest serves as an asynchronous HTTP request to the server to transfer data between the two parties without reloading the page.
- The primary benefit of JavaScript is its potential to help and deliver similar results in all modern browsers.
- Global corporations contribute to community growth by launching vital programs. Google (which invented the Angular framework) or Facebook are two examples (created the React.js framework).
- JavaScript is a scripting language that works well with other languages and can be used in a wide range of applications.
- Numerous open-source programs assist developers in adding JavaScript.
- There are numerous courses available in the field of JavaScript, that provide knowledge and resources to learn it rapidly and easily.
- There are many methods for using JavaScript through Node.js servers. It is possible to create a complete JavaScript application utilizing just JavaScript. [4]

2.2.2 Limitation of JavaScript

- Since JavaScript is directly used in web pages and client browsers, it can exploit the user's mechanism and executing malicious code on the client computer.
- Different browsers view JavaScript differently at times. Different layout engines can make JavaScript different, resulting in functionality and interface inconsistency. The majority of JavaScript is based on manipulating the browser's DOM components. Additionally, various

browsers, specifically Internet Explorer, provide different types of access to objects.

- JavaScript is a very old machine-based scripting language, and there is other technology that perform the same function (for example, JQuery) more efficiently and easily.
- If JavaScript is disabled in the window, the whole JavaScript code is not executed.
- The JavaScript file is downloaded to the client computer, allowing others to translate and reuse the text [5]

3 React

React.js is a free and open-source JavaScript library for developing user interfaces for single-page apps. React enables developers to build massive web apps that dynamically update data without requiring the user to refresh the website. React's primary goal is to be fast, scalable, and basic.

3.1 History

In 2011, Facebook developers were having issues maintaining their code. Facebook was growing expeditiously and with increasing amounts of features it was getting impossible for their team to keep up with all the updates and maintenance for their website. They needed a more efficient and fast way of maintaining their website. [6]

Jordan Walke, a software engineer working for Facebook, created a library which in later days came to be known as React.js. React is heavily influenced by XHP which is a HTML component framework for PHP.

React was first implemented in Facebook's newsfeed in 2011. After seeing its results in Facebook, Instagram also implemented React in their system. In due

course React became popular and grew exponentially and in May 2013 it was made Open Source at JSConf US. [5]

3.2 Features

React consists of many useful features to make life easier for developers. Some of the features of react are below.

- JSX

  JSX is a syntax extension to JavaScript. JSX specifies how the user interface appears. The primary purpose of JSX is to write HTML structures inside the same file as JavaScript code, which simplifies the code's comprehension and debugging. This avoids the usage of complicated JavaScript DOM frameworks.

- Virtual DOM

  A virtual DOM reflects the DOM in a virtual environment. It is essentially a memory within ReactJs for the individual items being generated for the website. It is implemented as a JavaScript Object. Virtual DOM was one of the primary reasons of React.js's speciality and simplicity.

  ReactJS generates a variety of objects that reflect a portion of the DOM. For example, rather than making a DIV entity with a UL element, it generates a React.div object with a React.ul object. By doing so, it can alter these objects very easily without modifying the entire DOM. Thus, if a part needs to be rendered, it uses the Virtual DOM rather than the real DOM to determine what to do. Manipulating the actual DOM takes much longer than manipulating the Virtual DOM. When the state of an entity shifts, Virtual DOM updates only the object in the actual DOM, rather than updating all items. [7]

- One-way Data Binding

  The term "one-way data binding" refers to data that can only flow in one direction and cannot be reversed. This ensures the data is communicated effectively. The parent component transmits data to the child component, but the child component cannot transmit data back to the parent component. This is how React.js's one-way data binding functions.

- Components

  ReactJS is heavily component based. Components are basically building blocks for any react application.

  There are two types of components, Functional component, and Class component. Functional components have no state of their own and can only use or only have a render method whereas Class components can have their own state and have a different specific render method for returning JavaScript syntax extensions on the screen. [5]

- State and props

  State is a built-in object in React. Value that belongs to the component is stored in the state object. The state object is where you keep all the component’s property values. The component re-renders when the state object changes. Another critical issue that could be considered is where state can be used. Initially, state could be found only in class components, it could not be used in functional components. Because of that functional component have been referred to as stateless components. But now state can be found in both class and functional components after the implementation of React Hooks.

Props are used to transfer data between React components and are short for properties. Between components, data flow is unidirectional in React (from parent to child only). [8]

3.3 Comparison to other frameworks.

Any frontend developer is familiar with three frameworks for developing web applications: React, Vue.js, and Angular. React is a graphical user interface library, Angular is a full-featured front-end platform, and Vue.js is a modern framework.
They can be used to create front-end apps almost interchangeably, but they are not identical, so it is important to compare them and appreciate their variations.
All three frameworks are heavily component based and enables accelerated development of user interface functionality. However, they each have a unique framework and design — so let us begin by examining their architectural distinctions to get a better understanding of the theory behind them.

React is a graphical user interface library, Angular is a full-featured front-end platform, and Vue.js is a modern framework.

Although they can be used almost interchangeably to develop front-end apps, they are not identical, and therefore it makes sense to compare them and consider their distinctions.

3.3.1 React vs. Vue

Although there are many parallels between React.js and Vue.js, there are a few variations that have a significant effect on what each is better suited for. The primary distinction between Vue and React is in the methods used to make data into the DOM. Vue uses HTML templates as well as JSX, while React only uses JSX. JSX is an enhancement that helps you to inject HTML directly into JS code. Although JSX can help speed up and simplify bigger, more complicated activities,

it can also make what should be a simple process more difficult. Components, DOM manipulation, and part state control are also core React features. The culture develops and supports everything more. Though experienced developers might prefer this degree of control, newcomers may be intimidated by the number of third-party libraries and resources available.
Though Vue has many community-built solutions, the core team often creates and maintains widely used platforms and libraries like Vue-router, Vuex, and Vue CLI. This mix of pre-built and third-party tools caters to the needs and preferences of both novice and experienced programmers. [9]

3.3.2 React vs. Angular

React is a JavaScript library that is far older than Angular. React is a library that can be bundled with other programming libraries. Angular is a self-contained approach. It is simpler to understand than Angular. However, when combined with Redux, it becomes more challenging to remember. For beginners, learning Angular is not fast. As a result, extensive preparation is needed. In terms of group support, react falls short, angular possesses a viable and dependable mechanism of mutual service. React needs some time to set up. However, it is extremely quick for delivering programs and developing applications. Although Angular is simple to set up, it can result in a rise in coding time, which may result in project delays. React allows you complete control over the tools, design, ad libraries used to create an application. Angular provides a small degree of liberty and versatility. React's data linking is one-way, which ensures that the UI components cannot be modified without also modifying the corresponding model state. On the other side, Angular employs the two-way data linking technique. It enables you to ensure that the model state adjusts automatically in response to any update. [10]

4 WebSocket

WebSocket is a stateful mechanism the ensures that communication between the client and server can continue before one of the parties wishes to close it (client or server). After the server or client closes the connection, it is terminated at both ends. The following sections discuss the main characteristics of web sockets.

- Real-time web application

  Real-time web application allows use of a web socket to view data that is continuously sent by the server at the client end. WebSocket's allow continuous data transmission over an already-established network, which speeds up the application.

- Gaming application

  The server receives data in gaming applications without requiring the user interface to be refreshed. In gaming apps, the server collects data without refreshing the user interface; the UI is automatically reset without initiating a new request, which is incredibly advantageous in a gaming program.

- Chat program

  The chat application makes use of WebSocket to establish a connection only once for the purpose of exchanging, posting, and broadcasting messages between subscribers. It makes use of the same WebSocket connection to send and receive messages, as well as one-to-one message transfer. [11]

5 Technology Used

5.1 HTML

HTML is an abbreviation for Hypertext Markup Language. It enables users to build and organize web pages and applications by allowing them to create and structure sections, paragraphs, heading etc.

HTML does not support complex features. Other than that, it enables the organization and formatting of documents in a manner comparable to Microsoft Word. As we deal with HTML, we mark up a website page using basic code structures (tags and attributes). For instance, we can construct a heading by enclosing the text within a beginning <h1> and ending with </h1>.

In 1991, Tim Berners-Lee, a scientist at Switzerland's CERN science centre, released the first edition of HTML. Since then, each modern edition of the HTML language has introduced new markup tags and attributes (tags modifiers). There are currently 140 HTML tags, several of which do not support modern browsers. The language's most significant update occurred with the release on HTML5 in 2014. It added many new semantic tags to the markup, such as <article>, <header>, <footer>, that disclose the context of their own material. [12]

5.2 CSS

CSS is an abbreviation for cascading style sheets. In a nutshell, CSS is a programming language that enhances the visual appeal of a website over bland or uninspiring text. If HTML is primarily concerned with textual material, CSS is concerned with visual structure, layout, and aesthetics. HTML is a markup language, and CSS is a language for creating style sheets. [13]
CSS is a rule-based code, which ensures that you define rules specifying style classes to be applied to individual elements or groups of elements on your web page.

To begin the rule, a selector is used. This option specifies the HTML attribute to be styled. And there are the curly braces. Among such would be one or more statements in the context of property and value pairs. Each pair specifies a property of the chosen element(s), accompanied by the value to be added to the property.

Before the colon, we have the property and after the colon, we have the value. Allowable values for CSS properties differ due to the property being described. In the following example we are styling the paragraph or <p> tag. Hence, p is a selector. We have one declaration inside the curly braces which has the property of color, and value is given red. Hence, it will change the color of all paragraphs in the HTML document to blue.

```
p {
    color: blue;
    }
```

Both web technology, such as HTML and CSS, are specified in lengthy documentation referred to as specifications (or referred to as "specs" by the W3C, ECMA, or Khron).

CSS was developed by the CSS Working Group of the World Wide Web Consortium. This group consists of organisations and individuals committed to CSS. The CSS Working Group creates new CSS rules in response to web developer requests for new functionality or browser requests for enhanced capabilities... CSS features are being introduced at a rapid rate. [14]

5.3 Node.js

One of the most common areas of confusion for newcomers to Node.js is determining what it is. Is it a distinct language, a structure built on top of another, or something else? Node.js is not a modern language, nor is it merely a JavaScript framework. It can be thought of as a JavaScript runtime framework

designed on top of Google’s V8 engine. Thus, it offers a framework in which we can write JavaScript code on any platform that supports node.js.
Now a little history. In 2009, Ryan Dahl delivered a lecture at JSConf that permanently altered the course of JavaScript. He added that Node.js to the JavaScript world during his introduction. He ended his nearly 45-min speech with a standing ovation from the crowd. He was motivated to create Node.js after seeing a basics progress bar for file uploads on Flickr, an image sharing website. Recognizing that the site was approaching the procedure incorrectly, he determined that there had to be a better way.
[15]

5.4 Express JS

Express is a framework for Node.js that makes it significantly simpler and cleaner to setup and use APIs and web servers. Being lightweight and containing most of the NodeJS features makes it very popular among the developers.

5.5 Material UI

Google created a UI component library focused on the company's material design guidelines. It is composed of numerous open and configurable UI widgets, and the components are self-contained, injecting only the styles required for show. With a large community behind it, it is one of the most common component libraries. The primary selling points are the user-friendly architecture and usability elements derived from Google's familiarity with user interface interfaces.
We can install Material UI by using a simple command,
`npm install @material-ui/core` inside the terminal.

5.6 Visual Studio Code

In April 2015 at Build developer conference, Microsoft launched Visual Code, which runs on OS X, Linux, and Windows. The code editor got updated and enhanced over the years and now is the one of the most popular code editors in the world. [16] The main features of Visual Studio Code include

• Integration of Git

Although using Git can be difficult at times, Visual Studio Code contains embedded Git, which allows quick GUI-based addition, commit, pull, and push modifications to a remote Git repository.

- Debugging

Configuring debugging software in Visual Studio Code is a breeze thanks to the comprehensive API. Debugging is a wide topic that often varies by language/stack. Debugging plugins are required for the language you are using, and you will be able to debug the code using breakpoints when creating.

- Integrated Terminal

While a terminal is typically accessible to the side or elsewhere on the computer while running code, Visual Studio Code has an Integrated Terminal that simplifies production.

- Plugins and Themes

Visual Studio Code has a robust plugin API, which enables developers to create truly amazing plugins. I will mention a couple of the more popular

ones I have seen, but for a more detailed list, we can visit the Visual Studio Code Marketplace. [17]

5.7 Socket.IO

In 2010, Socket.IO was created. It was created to promote real-time connection, which was already a relatively new concept at the time.

Socket.IO enables bi-directional connection between the client and the server. Bi-directional communication is allowed when a client's browser includes the Socket.IO package and a server also includes the Socket.IO package. Though data may be transmitted in a variety of formats, JSON is the most straightforward.

Socket.IO makes use of Engine.IO to create the connection and share data between the client and server. This is a low-level implementation that is used internally. Engine.IO is used for the server implementation, while Engine.IO-client is used to implement the client. [18]

5.8 Web RTC

WebRTC is an HTML5 standard that enables the addition of real-time media contact between the browser and the user. WebRTC allows the integration of voice and video contact within websites without installing any additional plugins on the website. WebRTC was introduced in 2011 and has slowly increased in prominence and acceptance since then. [19]

5.9 Npm

Npm is a Node.js package manager that has hundreds of thousands of packages. Though it does contribute to the creation of the directory structure/organization,

this is not the primary objective. The primary objective is automatic dependency and package management. This ensures that you can specify any of the dependencies for your project within the package. json file, so if you (or someone else) want to get started on your project, they may just run npm install and all the dependencies will be enabled automatically. Additionally, it is possible to define which variants your project relies on to avoid patches destroying your project.

Manually downloading the libraries, copying them to the appropriate folders, and using them is possible. However, as the project (and its list of dependencies) expands, this becomes more time-consuming and messier. Additionally, it complicates collaboration and discussing the project.

Basically, npm is an invaluable element in workflow as a JavaScript developer (both client-side and server-side).

6 Implementation

In this project we used different dependencies such as material UI, react, simple-peer and socket IO client.

We used the npm command to install these dependencies. We created three component Notifications, options, video player.
In this project we have two important packet.json file which contains all the important dependencies.
The folder structure for this project is shown in Figure 1.

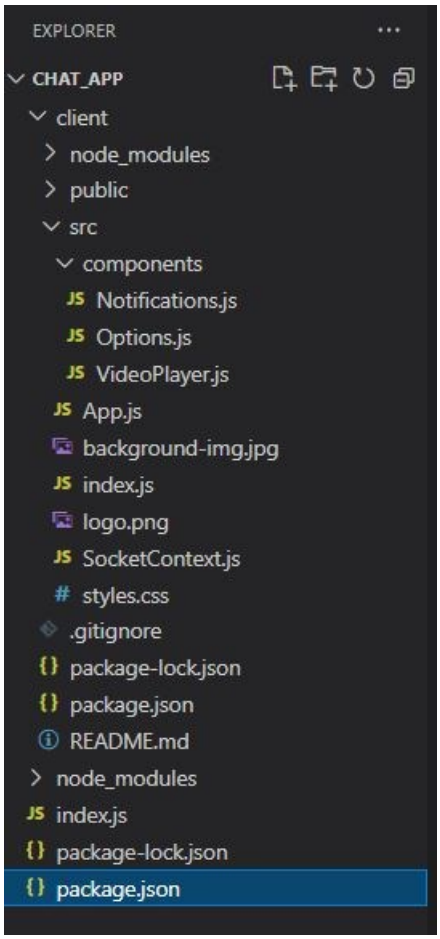

Figure 1: Folder Structure

6.1 Frontend Development

The front of the application was developed using Material UI, HTML, CSS. Material UI is the most preferred React UI library, as there is no other React UI library that can be even suggested. In the figure below we can see that the main content is in the centre of the page. This is done by using gridContainer. We created a logo from third parties and inserted it on the top of the page.
The width of the video player for the desktop is set to (490) pixel and for the mobile devices it is set to (300) pixel. After that we made two buttons to copy user's id and another button for calling.

Figure 2 below shows packet.json file of the client folder.

Figure 2: Packet.json of the client folder

The initial phase of the working website without styling was developed which is shown in figure 3 below.

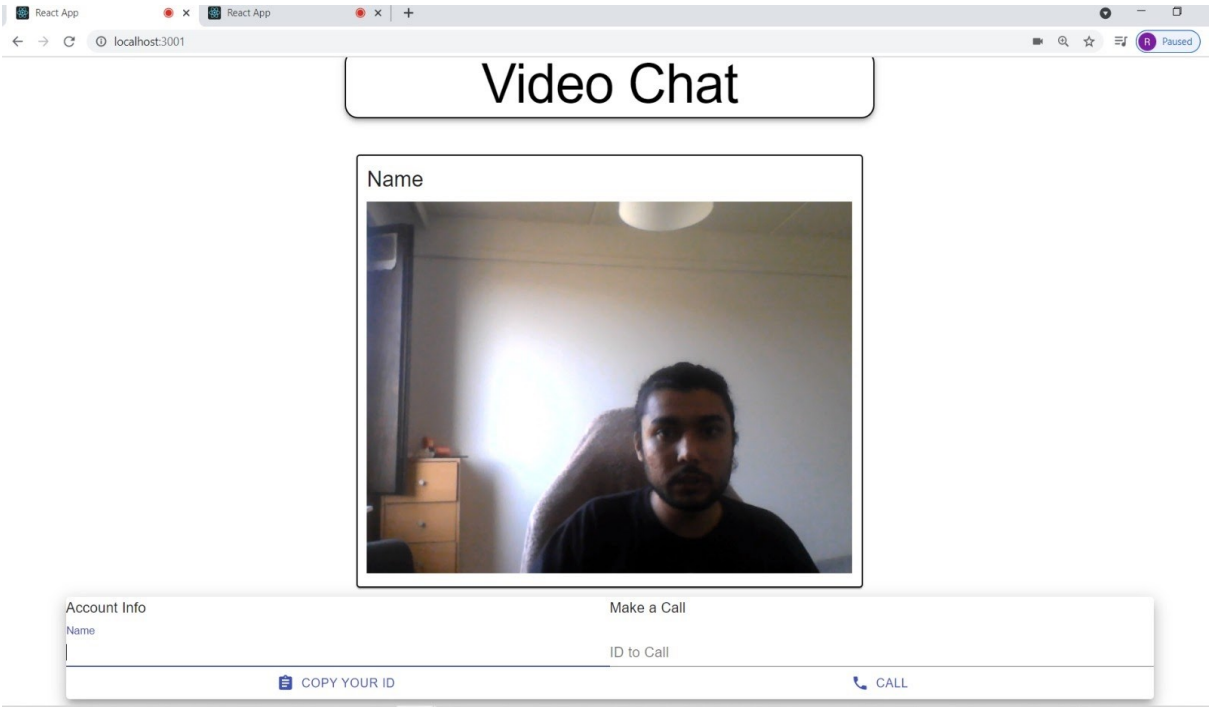

Figure 3: Initial phase of web application

6.2 Back Development

A packet.json file for chat_app was created with all the dependencies needed as shown in figure 4 below.

```
{
  "name": "chat_app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  ▷ Debug
  "scripts": {
    "babaramdev": "nodemon index.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "cors": "^2.8.5",
    "express": "^4.17.1",
    "nodemon": "^2.0.7",
    "simple-peer": "^9.11.0",
    "socket.io": "^4.0.2"
  }
}
```

Figure 4: Packet.json for chat_app

Backend of this web application consists of a simple nodeJS server which runs on port 5000 as shown 5 in figure below.

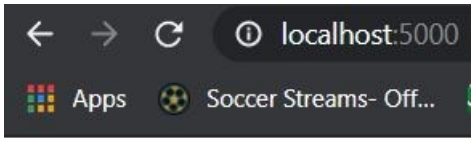

Figure 5: Backend server running.

All functionality of socket.io and webrtc is developed in the file socketcontext.js. It was decided to put all the socket logic inside one file so that it will be easier to read and will be inside the react context.
As shown in the listing 1 below these components were imported from react, socket.io-client and simple-peer.

```
import React, {createContext, useState, useRef, useEffect} from 'react';
import { io } from 'socket.io-client';
import Peer from 'simple-peer';
```

Listing 1. Importing components from react, socket.io and simple-peer.

After creating the initial context, the initial instant of socket.io is created as shown in listing 2 below.

```
const SocketContext = createContext();
const socket = io('http://localhost:5000');
```

Listing 2. Creating initial instant of socket.io

All the functions that we need to create our video chat work will be wrapped inside a context provider. The major functions are as follows:

- answerCall

  After we answer the call, we set the call accepted to be true. Once we have peer connection, we receive the data about the signal. Now we use sockets intertwined with our peers to finally establish that video connection. In the similar way we get the stream from another user.

```
 const answerCall = () => {
setCallAccepted(true);
const peer = new Peer({ initiator: false, trickle: false, stream
});

peer.on('signal', (data) => {
socket.emit('answerCall', { signal: data, to: call.from });
});

peer.on('stream', (currentStream) => {
userVideo.current.srcObject = currentStream;
});
peer.signal(call.signal);
connectionRef.current = peer;
  };
```

  Listing 3. Above shows answerCall() method.

- callUser

  In the call user function, we emit a callUser object that contains properties like userToCall. signalData, from etc. as show in listing 4 below

```
peer.on('signal', (data) => {
      socket.emit('callUser', { userToCall: id, signalData: data,
from: me, name });
    });
```

  Listing 4. Emitting callUser object.

- leaveCall

  To leave the call we setcallEnded property to be true and we destroy the current connection as show in listing 5 below:

```
const leaveCall = () => {
setCallEnded(true);
connectionRef.current.destroy();
window.location.reload();
};
```

  Listing 5. Destroying connection.

As soon as the page loads it asks the user for their permission to use audio and video. This is done by wrapping our code inside useEffect from react as shown in the Listing 6 below.

```
useEffect(() => {
    navigator.mediaDevices.getUserMedia({ video: true, audio: true })
    .then((currentStream) => {
    setStream(currentStream);
myVideo.current.srcObject = currentStream;
});
```

Listing 6. Implementing useEffect from react.

So, every time when we run our app, the browser asks permission to use audio and video as shown in the figure 6 below.

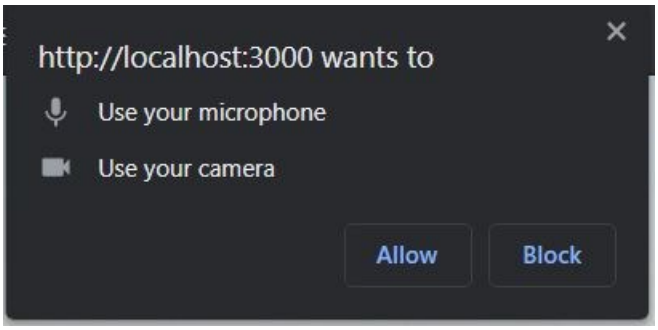

Figure 6:Permission to use audio and video.

After getting the audio and video we save the currentStream as a react state and populate our media by using useRef from react. After the connection, the id emitted from the backend server is received and set it as our state as well as caller's data such as name and signal.

Finally, we returned the <SocketContext.Provider> component which holds the major functionality of the current connection. Which can be used throughout the application.

7 Results

Finally, a working video chat app was created where you can call your friends and family.

The following are the steps to use the web app.

- Step 1: As shown in the figure 7 below the web application will ask for users' permission to use the microphone and camera.

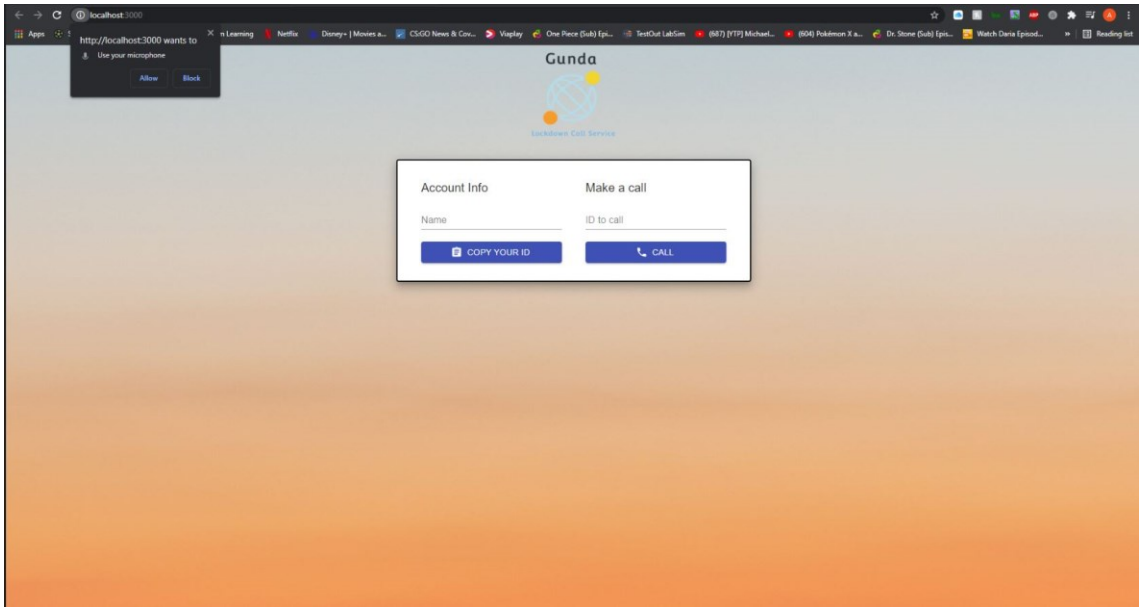

Figure 7: Asking permission to use audio and video.

- Step 2: After giving permission to use microphone and camera, web application will show the users video and can hear user's audio. This is shown in the figure 8.

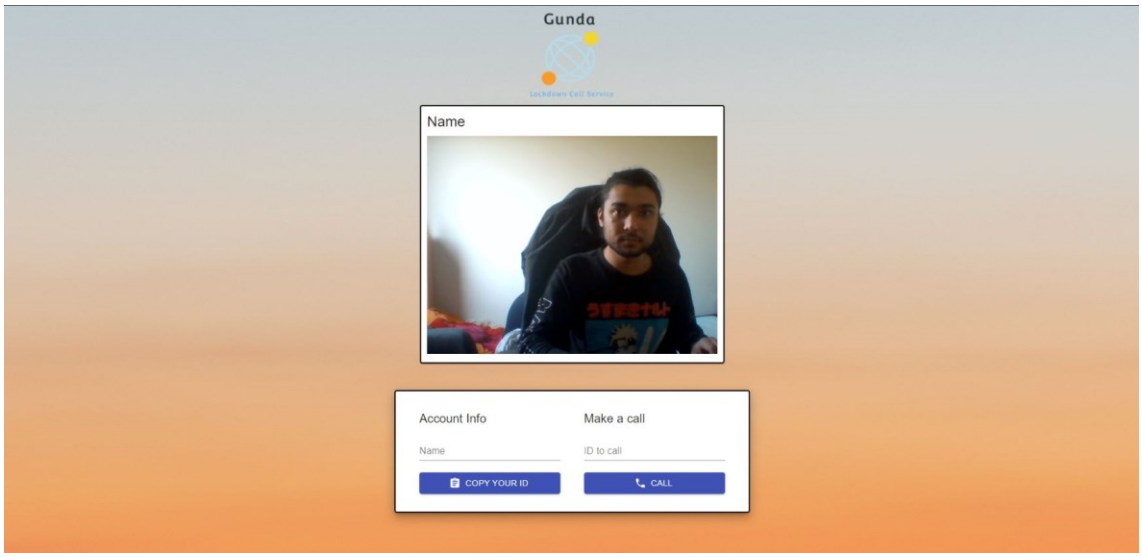

Figure 8: Showing users video and audio.

- Step 3: Third step is to put your name and ask the other user you want to call for their unique ID to call. This is shown in the figure 9.

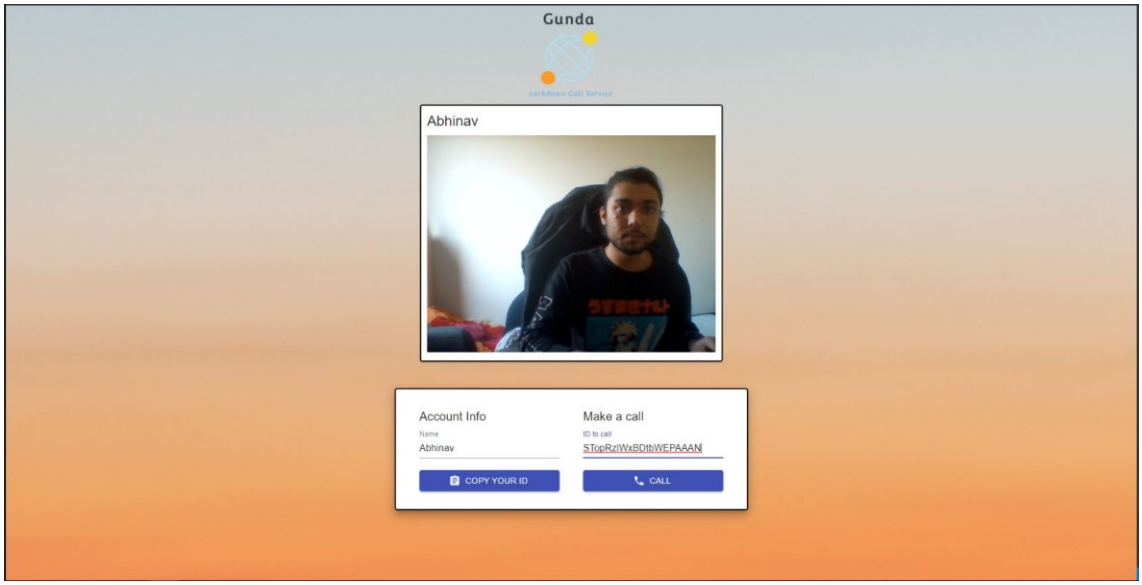

Figure 9: Adding Name and ID to call

- Step 4: After putting you name and a unique ID to call you can click call and then your friend will receive an option to accept the call. This is shown in figure 10.

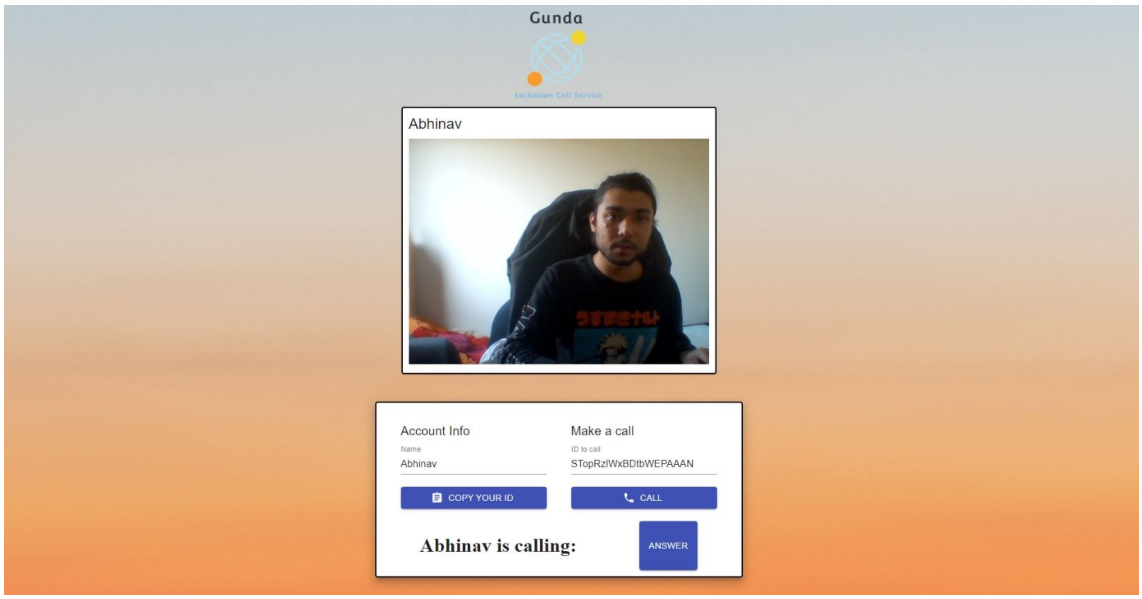

Figure 10: Option to accept call.

- Step 5: After answering the call user will be connected to the other user who is calling and can video chat with one another as shown in the figure 11.

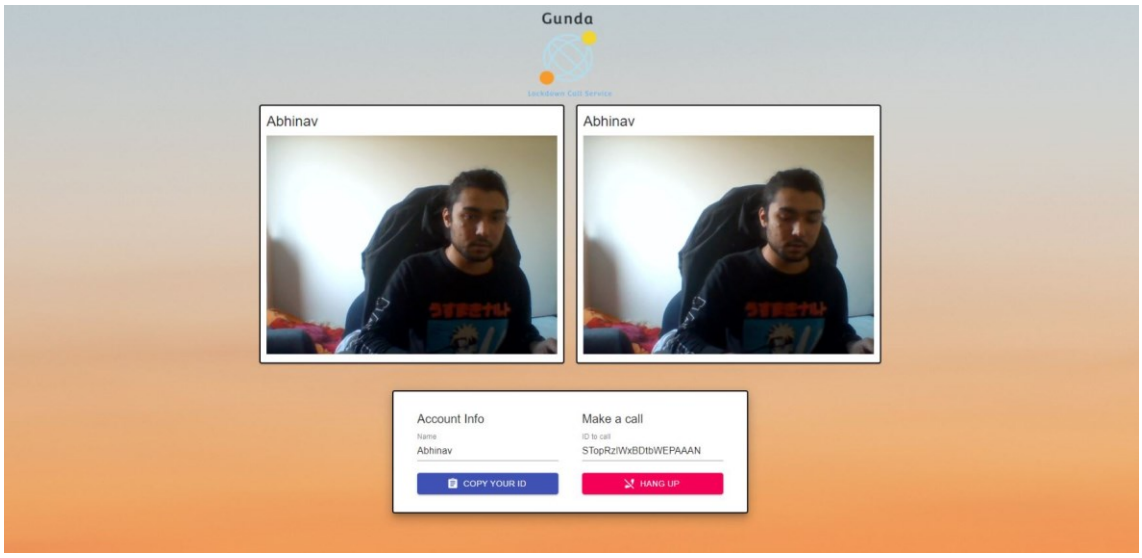

Figure 11: Video chatting

8 Conclusion

Today, React Js is one of the most common and efficient front-end technologies available. React Js is the only library that can be implemented anywhere. It performs admirably in all applications. Additionally, React Js is highly compatible with a variety of formats, browsers, and applications. Following the creation of the Chat App, it was obvious that React Js is an easy to learn and simple to implement framework.

In conclusion this project was a simple project in which you can video call your friends and family, but future development is very much possible by adding much more features in the application.

9 References

[1] “https://www.tutorialspoint.com/,” Tutorials Point, [Online]. Available: https://www.tutorialspoint.com/javascript/index.htm. [Accessed 07 05 2021].

[2] T. Kidder, “https://www.learnacademy.org/,” LEARN ACADEMY, 10 02 2020. [Online]. Available: https://www.learnacademy.org/blog/who-introduced-javascript-originally-called-why-created/. [Accessed 02 05 2020].

[3] “https://insights.stackoverflow.com/,” StackOverflow, 2019. [Online]. Available: https://insights.stackoverflow.com/survey/2019. [Accessed 02 05 2021].

[4] “https://www.geeksforgeeks.org/,” GeeksforGeeks, 25 11 2020. [Online]. Available: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-javascript/. [Accessed 04 05 2021].

[5] “http://net-informations.com/,” net-informations, [Online]. Available: http://net-informations.com/js/iq/advan.htm. [Accessed 01 05 2021].

[6] F. Hámori, “https://blog.risingstack.com,” RingStack, 04 04 2018. [Online]. Available: https://blog.risingstack.com/the-history-of-react-js-on-a-timeline/. [Accessed 06 05 2021].

[7] T. Sufiyan, “https://www.simplilearn.com,” Simplilearn, 09 04 2021. [Online]. Available: https://www.simplilearn.com/tutorials/reactjs-tutorial/what-is-reactjs. [Accessed 06 05 2021].

[8] C. Eygi, “https://www.freecodecamp.org,” freeCodeCamp, 09 02 2020. [Online]. Available: https://www.freecodecamp.org/news/react-js-for-beginners-props-state-explained/. [Accessed 21 04 2021].

[9] E. Lvova, “https://dzone.com,” DZone, 2021 03 19. [Online]. Available: https://dzone.com/articles/react-vs-vue-in-2021-best-javascript-framework. [Accessed 22 04 2021].

[10] “https://www.guru99.com,” Guru99, [Online]. Available: https://www.guru99.com/react-vs-angular-key-difference.html#9. [Accessed 28 04 2021].

[11] “https://www.geeksforgeeks.org/,” GeeksforGeeks, 04 12 2019. [Online]. Available: https://www.geeksforgeeks.org/what-is-web-socket-and-how-it-is-different-from-the-http/. [Accessed 02 05 2021].

[12] D. G, “www.hostinger.com,” Hostinger, 25 11 2019. [Online]. Available: https://www.hostinger.com/tutorials/what-is-html. [Accessed 08 05 2021].

[13] “https://blog.devmountain.com,” Devmountain, [Online]. Available: https://blog.devmountain.com/what-is-css-and-why-use-it/. [Accessed 25 04 2021].

[14] “https://developer.mozilla.org/,” MDN Web Docs, [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS. [Accessed 07 05 2021].

[15] B. J. D. J. K. Mithun Satheesh, “Web Development with MongoDB and NodeJS,” Packt, Birmingham, 2015.

[16] F. Lardinois, “https://techcrunch.com/,” TechCrunch, 29 04 2015. [Online]. Available: https://techcrunch.com/2015/04/29/microsoft-shocks-the-world-with-visual-studio-code-a-free-code-editor-for-os-x-linux-and-windows/?guce_referrer=aHR0cHM6Ly93d3cud2lraXdhbmQuY29tLw&guce_referrer_sig=AQAAAHAoQu-S6Qm61BXPY87sdFQPZky83wVXY1iiicEp3EwfHpRTNWzJ. [Accessed 14 04 2021].

[17] C. Ganga, “https://scotch.io/,” Scotch.io, 25 07 2018. [Online]. Available: https://scotch.io/bar-talk/my-top-8-visual-studio-code-tips-and-features. [Accessed 17 04 2021].

[18] G. Lewington, “https://ably.com/,” Alby, [Online]. Available: https://ably.com/topic/socketio#:~:text=Socket.IO%20allows%20bi%2Ddirectional,integrated%20the%20Socket.IO%20package.&text=To%20establish%20the%20connection%2C%20and,.IO%20uses%20Engine.IO.. [Accessed 25 04 2021].

[19] “https://bloggeek.me/,” Blogeek.me, 15 12 2019. [Online]. Available: https://bloggeek.me/what-is-webrtc. [Accessed 02 05 2021].

[20] “https://www.education-ecosystem.com,” Education Ecosystem, 2019. [Online]. Available: https://www.education-ecosystem.com/guides/programming/react-js/history. [Accessed 27 04 2021].

[21] L. Kalshteyn, “https://dev.to,” DEV Community , 21 10 2020. [Online]. Available: https://dev.to/gorgutzz/material-ui-intro-50md. [Accessed 03 05 2021].