



InterCode: An Interview Experience Enhancement Software

A Report Submitted
in Partial Fulfillment of the Requirements
for the Degree of
Bachelor of Technology
in
Computer Science & Engineering

by

**Siddharth Majumdar(20164044), Sunil Kumar(20164012),
Saurabh(20164104) and Pradyumna Pandey(20164159)**

Group: CS-26

to the

**COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
MOTILAL NEHRU NATIONAL INSTITUTE OF TECHNOLOGY
ALLAHABAD, PRAYAGRAJ
November 22, 2019**

UNDERTAKING

I declare that the work presented in this report titled “*InterCode: An Interview Experience Enhancement Software*”, submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology Allahabad Prayagraj, for the award of the ***Bachelor of Technology*** degree in ***Computer Science & Engineering***, is my original work. I have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, I accept that my degree may be unconditionally withdrawn.

November 22, 2019
Allahabad

(Siddharth Majumdar)

(Sunil Kumar)

(Saurabh)

(Pradyumna Pandey)

CERTIFICATE

Certified that the work contained in the report titled “*Inter-Code: An Interview Experience Enhancement Software*”, by *Siddharth Majumdar, Sunil Kumar, Saurabh and Pradyumna Pandey*, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Er. Rajesh Tripathi)

Computer Science and Engineering Dept.
M.N.N.I.T. Allahabad

November 22, 2019

Preface

When you look at the software industry, or any industry in general, the number of jobs have been increasing exponentially. With the significant increase in jobs, the process of selecting candidates for these positions has also come under scrutiny, for its efficiency and quality. It is, sometimes, not possible to interact with each individual, face-to-face, hence, comes the necessity of the concept of remote interviews. When, we talk about remote interviews, the important components should be a support for audio and/or video communication between the interviewer and interviewee, along with some additional utilities for different purposes. Hence, in this project, we will try to build a web application to support all aspects of a remote interview.

Acknowledgments

We would like to thank everyone who contributed in some way in helping us achieve our goals with regards to our project. We are especially grateful to our mentor Er. Rajesh Tripathi for assisting us in our project. Without his expertise and suggestions we would never have been able to complete our work.

We would also like to thank Dr. Shashwati Banerjea for the formation of groups for the project which enabled us to work on this project.

Abstract

With the advent of the concept of remote interviews, more and more companies and organizations have taken the help of it to recruit candidates. Remote interviews, in general, can be defined as an interview, where the interviewer and the interviewee aren't located in close proximity and the procedure takes place over the internet, with the help of several tools. The basic requirement of a remote interview should be an audio/video support for the purpose of communication between the two(or more) people. Video communication should be preferred, as it will ensure that the candidate does not indulge in any unfair means in the duration of the interview. Text communication (or simply, a chat window) will also help in some situations. When it comes to coding interviews, another important component will be a code editor to type in the required code.

For the purpose of this project, we have created a web application called *Inter-Code* that will encompass all the necessary features of conducting remote interview, which includes video communication support, text communication support, a collaborative code editor, along with a compiler and executor and also the feature to upload a resume.

Contents

| | |
|--------------------------------------------------|-----------|
| Preface | iv |
| Acknowledgments | v |
| Abstract | vi |
| 1 Introduction | 1 |
| 1.1 Motivation | 2 |
| 2 Related Work | 3 |
| 3 Proposed Work | 5 |
| 3.1 Objective | 5 |
| 3.2 Software Requirement Specification | 6 |
| 3.2.1 Introduction | 6 |
| 3.2.2 Overall Description | 7 |
| 3.3 Platforms Used | 8 |
| 3.3.1 PhpStorm | 8 |
| 3.3.2 MongoDB | 9 |
| 3.3.3 Postman | 10 |
| 3.4 Overview of Approach | 11 |
| 3.5 Description of the Features | 12 |
| 3.5.1 Login and Registration | 12 |
| 3.5.2 Collaborative Code Editor | 13 |
| 3.5.3 Video Communication | 14 |

| | | |
|----------|-----------------------------------|-----------|
| 3.5.4 | Text Communication | 14 |
| 3.6 | The User Interface | 15 |
| 4 | Results And Analysis | 18 |
| 4.1 | Analysis | 18 |
| 4.1.1 | The Testing Process | 18 |
| 4.2 | Results | 18 |
| 4.3 | Shortcomings | 19 |
| 5 | Conclusion and Future Work | 20 |
| 5.1 | Conclusion | 20 |
| 5.2 | Future Work | 21 |
| | References | 22 |

Chapter 1

Introduction

In recent years, with the advent of transmission technology, video communication has become an integral part of many software and applications. It is not only restricted to simple video calling applications such as Skype, by Microsoft, which only involves audio, video and text communication, but also extends to several integrated applications available on the internet. The transmission takes place over a network, and the methodology and protocol varies according to the type of transmission taking place, such as audio or video.

The concept of audio and video transmission dates back to the 80s and 90s with several modifications and enhancements arriving with the advent of technology. With this technology, the computer science domain has seen a lot of progress, in terms of reach and exploration. It has led to the world becoming a smaller place with people staying connected with one another in a better way.

The concept of the collaborative editor was first demonstrated in the 60s. The concept has then been modified from just being a text editor to becoming a code editor which would implement several other features such as auto-indentation of code, highlighting of variables and functions according to the language chosen. A collaborative editor ensures that the two parties in question maintain the same data, as edits by either party are reflected on the other as well.

1.1 Motivation

When we talk about industrial software development, it is always about the integration of smaller components to form one big utility software that fulfils various aspects of a particular task in question. Constructing a software by combining all basic tools of a remote interview, will surely help various entities and organization, including college placements cells, to help conduct interviews and give students a bright career.

Chapter 2

Related Work

Since the project is essentially an integration of small components, each component's history and existence will be talked about here. The main components of the project are: video communication, text communication, collaborative code editor, along with database support.

Digital video transmission or communication dates back to the 90s, when the concept of audio and video transmission system and receiver system was patented by Yurt et al in 1992[7]. There has been a significant amount of literature associated with this topic, as people have attempted variations of video communications over different types of networks such as wired or wireless, and using different types of mechanisms. The arrival of broadband and ISDN technologies proved to be helpful for the enhancement of the technology behind video communication or video telephony.

Text communication, or more precisely the concept of chat rooms dates back to the 1970s, when the first chat system was used by the US Government in 1971. It was developed by *Murray Turoff*, a PhD scientist at Berkeley, and used for the purpose of President Nixon's policies. Today, it has been in the use for more than four decades, and used by multiple clients such as Ares Galaxy, eMule etc.

The concept of collaborative text editor was first demonstrated by *Douglas Englebart*, in a landmark event called *The Mother of All Demos* in 1968. Then came the *Instant Update* from *ON Technologies*, which came for both macOS as well as

Microsoft Windows in the 1990s. In 2006, Google acquired a product called *Writely*, which later became *Google Docs*, and provided simultaneous edits on the entirety of a document, though changes from other users were only reflected after the client program polling the server (every half-minute or so). There were some beta versions of softwares trying to implement standalone collaborative code editors, but they lacked features such as auto-indentation and code compilation in them.

Chapter 3

Proposed Work

3.1 Objective

The development of a complete web application to cover all aspects of a remote interview is a key requirement of the software industry today, where the recruitment process has become very crucial. Hence, we aim to develop a software that will provide all the features such as video communication, text chat room, collaborative code editor and compiler etc.

3.2 Software Requirement Specification

3.2.1 Introduction

■ *Purpose*

The aim of this document[1] is to provide a detailed description of InterCode : The Interview Experience Enhancement Tool. It will cover the applications and features of the system, the interfaces of the system, what the system is expected to do, the constraints that the project will work under and how it behaves in response to external stimuli. This is intended for both the developers and the users of this system.

■ *Scope*

The system is intended for making an interview experience enhancement tool, which covers all the requirements of a remote software/coding interview. As of now, the project is hosted on a local network, we can publish it for the global community in the future. Some features such as identifying patterns among the candidates interviewed to enhance the experience as well as the interview the content and providing a pre-defined question bank can also be implemented.

■ *Glossary*

- Collection: A collection in a NoSQL databases is analogous to tables in a RDBMS.
- Document: Each entry of a collection is called a document.

■ *Overview of Document*

The next section, the Overall Description section, of this document gives an overview of the functionality of the project. It describes the informal requirements and is used to establish a context for the technical requirements specification in the next section. The third section, Requirements Specification section, of this document is written primarily for the developers and describes in technical terms the details of the functionality of the product. Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

3.2.2 Overall Description

■ *System Architecture*

The primary purpose of the web application is to provide all the features that are necessary for a conducting an online coding interview. Now, the software has three main components, a collaborative code editor-cum-compiler, a video communication window and a chat room feature. The software has a back end developed on Node.js[5], and the front end support has been provided by Handlebars, a template generation engine, that helps in the auto generation of HTML based web pages. The database support has been provided using MongoDB, which is a NoSQL database program. The web application is hosted on a particular IP address and a port number, which is used to access it from a remote client, which is connected to the same network as the server computer.

3.3 Platforms Used

3.3.1 PhpStorm

PhpStorm is a PHP integrated development environment developed by JetBrains, and serves the purpose of developing web applications. It provides support to a host of other languages through its software suite like IntelliJ IDEA(Java), Python(PyCharm), JavaScript(WebStorm), SQL(DataGrip) etc. The interface of the software has been illustrated in Figure 1.

In our project, PhpStorm has just served as an IDE, that enhances the developer experience, and helps in debugging the code and understanding it through auto-indentation and comments. Since the web application is built on Node.js, the execution is directly done through the terminal, using Node Package Manager (NPM).

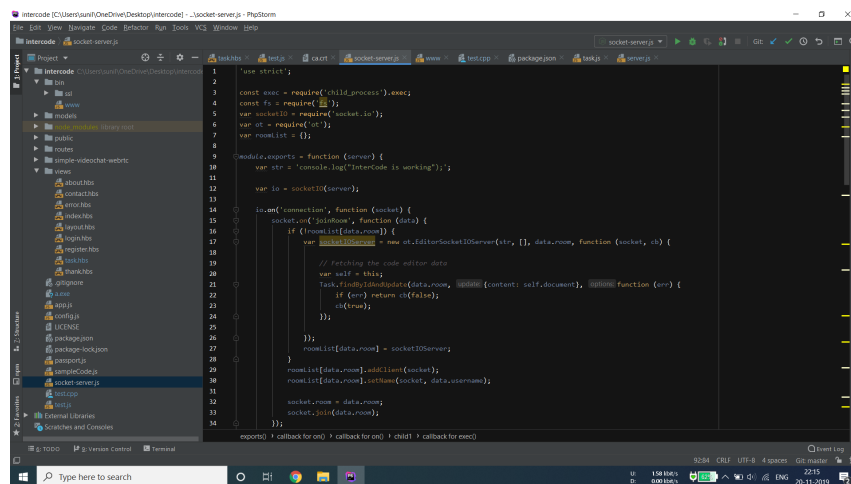


Figure 1: PhpStorm user interface

3.3.2 MongoDB

MongoDB is an open source database program that works on the NoSQL principle. NoSQL principle provides an alternative to the conventional concept of relational databases[3]. NoSQL databases prove to be useful, when the data in question is large and distributed.

NoSQL databases do not essentially follow a traditional and established relational schema. These databases are used in large-scale organizations such as Google to focus on narrow operational goals, and employ relational databases where high-grade data consistency is necessary.

For better visualisation of the databases, we used a tool called *Robo 3T*, which is essentially a GUI for the databases created using MongoDB, which has been illustrated in Figure 2.

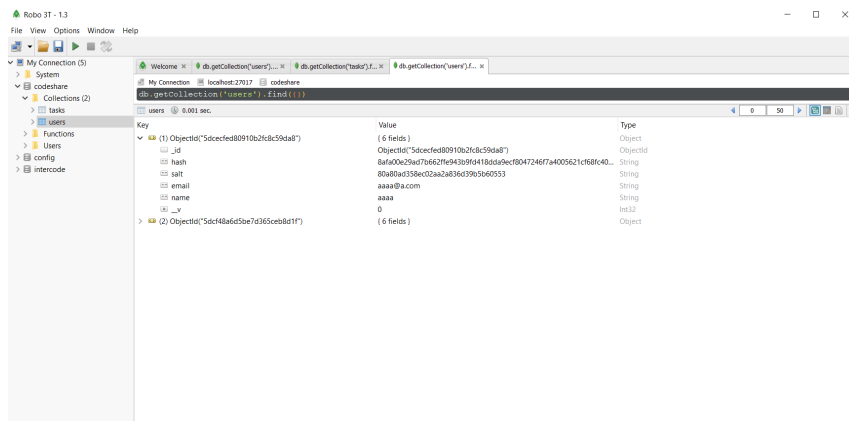


Figure 2: Robo 3T user interface

3.3.3 Postman

Postman is a collaboration platform for API development. The features of Postman simplify each step of building an API and streamline collaboration to help create better APIs.

In our project, Postman was used to basically test the output of the GET and POST requests that were being sent to the given IP address that was behaving as the host system. It helped in debugging the code as well as resolve logical issues in the structure of the project, which included the back end as well as the front end. The user interface of the Postman app has been illustrated in Figure 3.

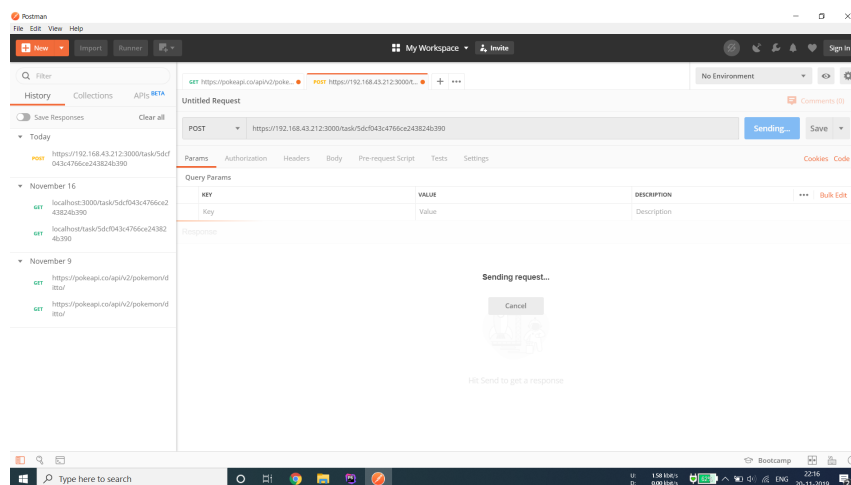


Figure 3: Postman user interface

3.4 Overview of Approach

For the purpose of developing this web application, we started by looking at existing collaborative text editors, which are available online, and also gained expertise on Node.js, as it is a rather novel language, when it comes to the college students.

Then, we decided on the additional features that will be provided in this web application, to enhance the experience of the interviewer and the interviewee, apart from the collaborative editor and the video communication. We decided upon providing a feature of a chat room and the provision of uploading the candidate's resume as a file. The feature of logging in, using Facebook credentials was also added, to simplify the registration process in the web application.

As the web application was developed, it provided seamless video communication and effective collaborative coding experience to the two parties, who also had the option of communicating via text messages.

3.5 Description of the Features

When we look at the web application, we can broadly divide it into four main features, which will be discussed below.

3.5.1 Login and Registration

The entire user data for the web application has been stored in the form of a collection in a MongoDB database, by the name *Users*. Each document contains the following fields:

- ID: A unique identification assigned to every document.
- Name: This field stores the name of the user.
- Email: This field stores the e-mail address of the user.
- Salt: This field stores a random string that is used as additional input to a one-way function that encrypts the password entered by the user.
- Hash: This field stores the output of the hash function applied on the password along with the salt value.

When a user chooses to register on the web application, the user can either enter his details manually on the form available on the website, which includes his name, email address and password, or he can choose to register using his Facebook credentials. In case he chooses to register using his Facebook credentials, the document for his account will not contain the *hash* and *salt* fields.

When the user tries to login to his account, the validation process will have two methods, depending upon the way the user had registered. If the registration was done with email and password, the validation process, will match the entries of the Email and Password fields to the Document fields. The password entered will be hashed using the hash function with the salt value of the corresponding email as an additional input, and the final output will be compared to the hash value. In case, the registration is done using Facebook, the password authentication process will take place using the Facebook API.

3.5.2 Collaborative Code Editor

The implementation of this feature required the help of a concept called Operational Transformation. This concept was pioneered by C. Ellis and S. Gibbs in the GROVE system in 1989. It has been hosted as an open source JavaScript utility, which comes with another utility called *CodeMirror*. CodeMirror is a specialised text editor which specialises in editing the code over a particular application.

The Operational Transformation[2] utility based on JavaScript has been integrated into the web application, with the socket object being passed at every instance of the CodeMirror object that is being created with the creation of a new task. Every change in the editor is represented as an *operation*. For example, insertion of a character 'c' at position 14, can be called an operation. Now, in order to handle concurrent operations, there is a function that takes two operations that have been applied to the same document state (but on different clients) and computes a new operation that can be applied after the second operation and that preserves the first operations intended change.[6]

The CodeMirror simply provides the editor for the Operational Transformation utility, and is the place where the changes caused by the collaborative editing can be visualised. As of now, the code editor includes auto-indentation and variable highlighting for JavaScript, C and C++. The code can also be compiled on both ends at any point of time. Compilation of codes in C, C++ and JavaScript are supported. This helps the interviewer for the immediate evaluation of the code written by the interviewee.

Now, simple application of Operational Transformation using Code Mirror causes a problem. If the editor window is refreshed, contents of the code editor are erased, and cannot be retrieved by any means. Thus, to solve this problem, we take help of the database. Now, every task is stored as a separate document in the collection, with the contents of the editor, being constantly updated in the corresponding document.

3.5.3 Video Communication

The implementation of video communication in this web application has been achieved by using *WebRTC*[4], which is an open source project that provides real time communication to web browsers and mobile applications using simple APIs.

For the purpose of this project, we use *simple-peer*, which is a package in Node.js that implements WebRTC and helps in achieving real time communication. The communication is taking place over a network using a socket which has been assigned a particular port number, which stays constant throughout the communication. In our project, the port number used for video communication is **9000**.

Now, it is important to know that none of the modern web browsers support video communication over HTTP. All of them require a HTTPS certificate, i.e. HTTP over SSL. Therefore, in order to implement the video communication, we had to use the HTTPS certificate, along with public and private key exchanges, so that the browsers allow video communication on the web application.

3.5.4 Text Communication

In order to implement the text communication feature, we used a library called *Socket.IO*, which is available in multiple languages including JavaScript. This enables real-time bidirectional event-based communication and consists of a Node.js server and a JavaScript client library.

Since we wished to implement a chat room feature, each user was assigned a user ID and each task was also assigned a particular task ID. The task ID was used to identify the users part of the same task, and therefore, the same chat room. This ensures that the text messages are not sent to a person who is not a part of the communication, and the messages remain inside the chat room. The Task ID is a unique combination of alphanumeric characters and has a capacity to keep a track of millions of such tasks. The provision of uploading a file (most probably a resume) has also been given in this text communication feature.

3.6 The User Interface

All of the major components have been discussed in the respective sections above. The front end of the web application has been created using *Handlebars*, a template generation engine, which is used for auto-generation of HTML-based web pages.

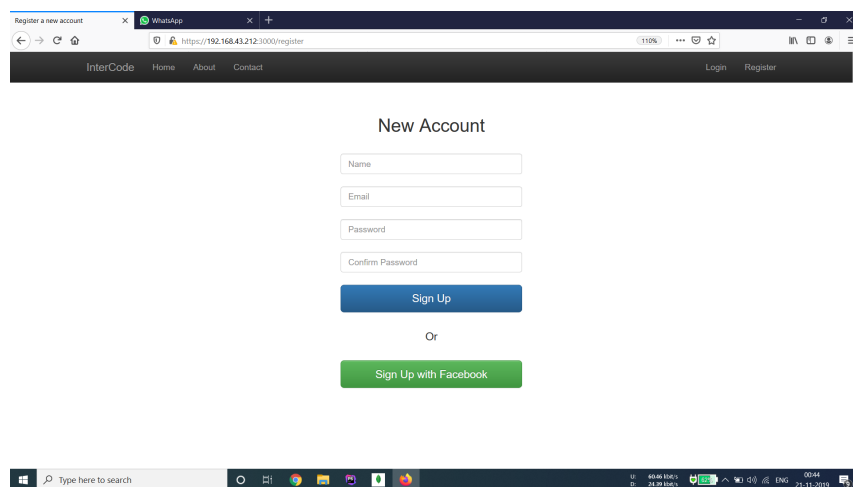


Figure 4: The Account Registration Screen

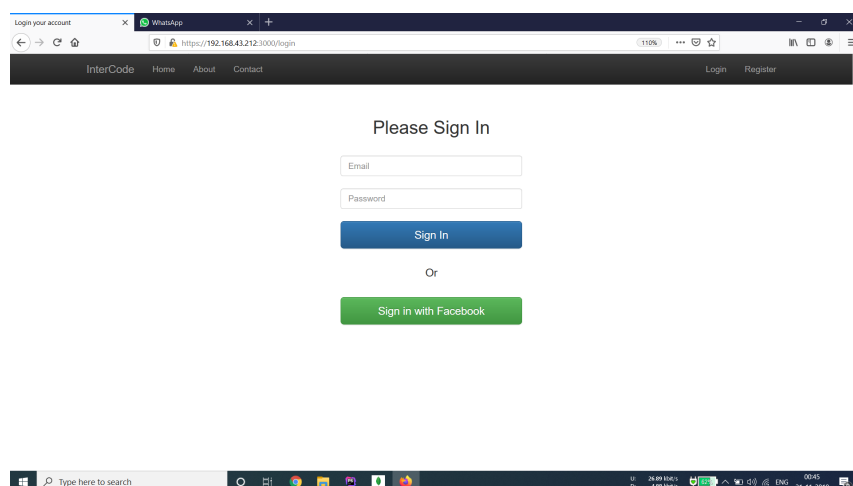


Figure 5: The Login Screen

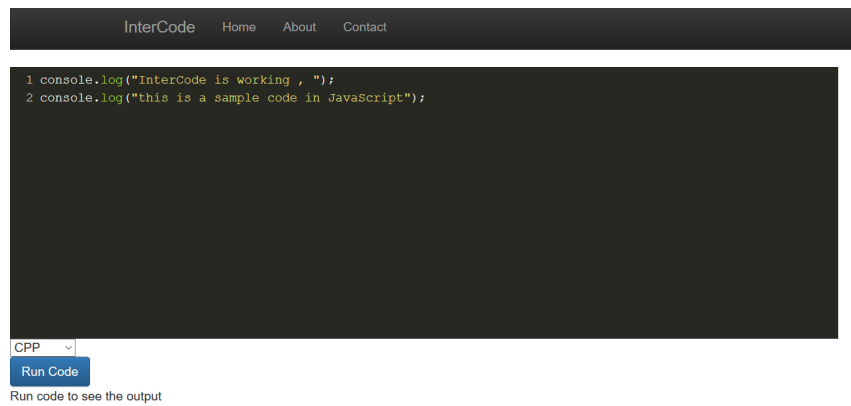


Figure 6: Collaborative code editor with JavaScript Code

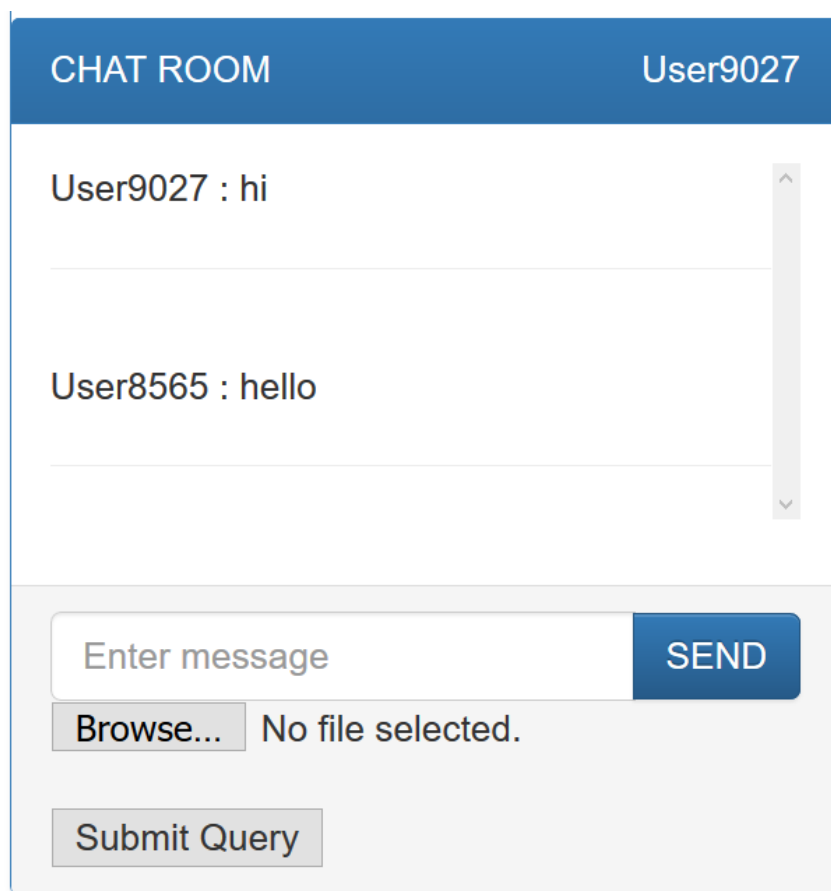


Figure 7: The Chat Room Screen

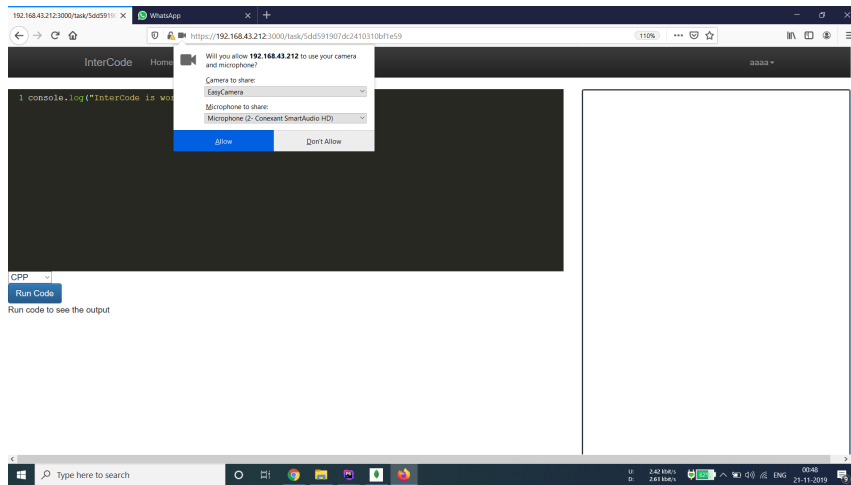


Figure 8: A task window

Chapter 4

Results And Analysis

4.1 Analysis

4.1.1 The Testing Process

The software testing is a very important part of the software engineering for this project. For testing purposes, two of our team members volunteered to perform the roles of the interviewer and interviewee, and performed communication over the web application, which included video communication and code collaboration as well.

The tool Postman was also used to check the validity of the web application, by sending various GET and POST requests, and comparing the generated results to the expected ones.

4.2 Results

The results of testing the software, as illustrated in the Figure 9 and Figure 10 below, were positive. The team members were able to perform seamless communication without any hassle. The collaborative code editor also worked perfectly, with each modification taken into account, and reflecting respective changes by each individual client.

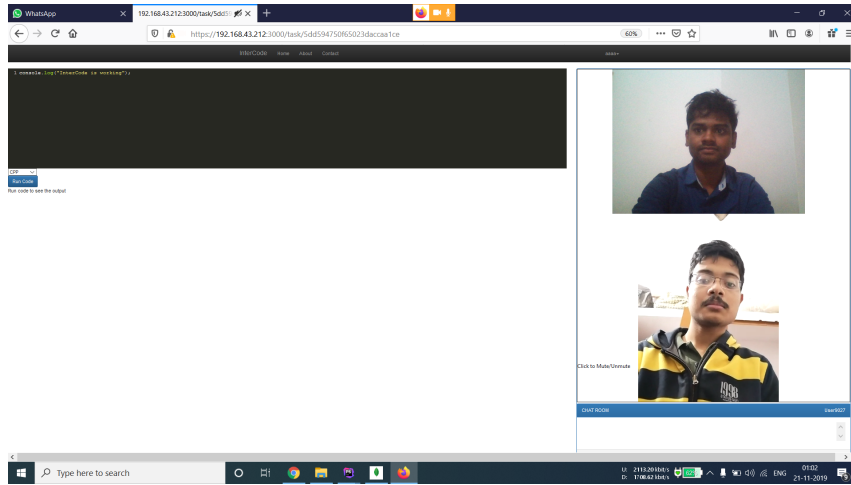


Figure 9: Team members successfully communicating using the video chat feature

4.3 Shortcomings

The maintenance of traffic on the server remains an unresolved issue in the project, which is definitely a scope of improvement. Apart from that, the user interface of the web application can be enhanced to improve the user experience.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The main aim of undertaking the project was the development of an integrated web application that shall serve all requirements of a remote interview. After working on the constructive criticism of the project given by our project mentor, a lot of steps were taken and a number of modifications were done on the project that enhanced its working. This application will surely help the organisations and the placement cells of colleges in conducting a smoother recruitment process. For colleges with a lack of infrastructure, where it is not possible to conduct face-to-face interviews for all students simultaneously, this web application will prove to be really helpful, as all students can take the interviews from their respective locations. A better recruitment process will surely lead to better jobs for the bright students of the country, who will in turn, contribute their part, in improving the country's economy, which is facing a serious crisis now.

5.2 Future Work

The project was aimed at creating a one-stop solution for remote interviews. However, given the fact that the infrastructure we used, it was not possible for us to handle large amount of traffic on the server. A potential improvement to this web application can be the usage of load balancing algorithms on the server and hosting them on a distributed system. For the purpose of managing traffic, a limitation of the application has been imposed wherein more than two people cannot be part of the same room. These are some issues, which should be addressed in near future.

References

- [1] Recommended practice for software requirement specification. Standard IEEE 830-1998, IEEE, 1998.
- [2] GOLDMAN, M. *Software Development with Real-Time Collaborative Editing*. PhD thesis, Massachusetts Institute of Technology, September 2012.
- [3] GYORODI, C., GYORODI, R., PECHERLE, G., AND OLAH, A. A comparative study: MongoDB vs. MySQL.
- [4] JANG-JACCARD, J., NEPAL, S., CELLER, B., AND YAN, B. WebRTC-based video conferencing service for telehealth. *Computing by Springer 98*, 169–193.
- [5] Node.js Official Documentation. <https://nodejs.org/en/docs/>.
- [6] SPIEWAK, D. Understanding and Applying Operational Transformation, 2010. [Online; accessed 17-May-2010].
- [7] YURT, P., AND BROWNE, H. L. Audio and video transmission and receiving system, U.S. Patent 5 132 992, Jul. 21, 1992.