InterCode : The Interview Experience Enhancement Tool

CS-26

Siddharth Majumdar 20164044
Sunil Kumar 20164012
Saurabh 20164104
Pradyumna Pandey 20164159

Under the guidance of Er. Rajesh Tripathi
CSED, MNNIT Allahabad, Prayagraj

November 28, 2019

Outline

Introduction

- Rapid growth of the software industry has led to requirement of more software professionals
- This necessitated the presence of an efficient recruitment strategy
- Advent of real time communication has facilitated remote interviews
- Remote interviews reduce hassle for both the interviewee and the interviewer
- Absence of a full-fledged software to fulfil all requirements of a remote coding interview

Problem Description

A remote coding interview has several aspects to it, which includes real-time communication, preferably video communication, code editing and compilation, resume and some additional features. We aim to develop a consolidated web application that will provide all the above functionalities to the user as a one-stop solution.

Requirements

- Any IDE like WebStorm, PHPStorm can be used to develop project
- Database facility using MongoDB: A NoSQL database program
- A system with webcam support and can handle test-level traffic
- A small-sized network to implement the web application

Screenshot

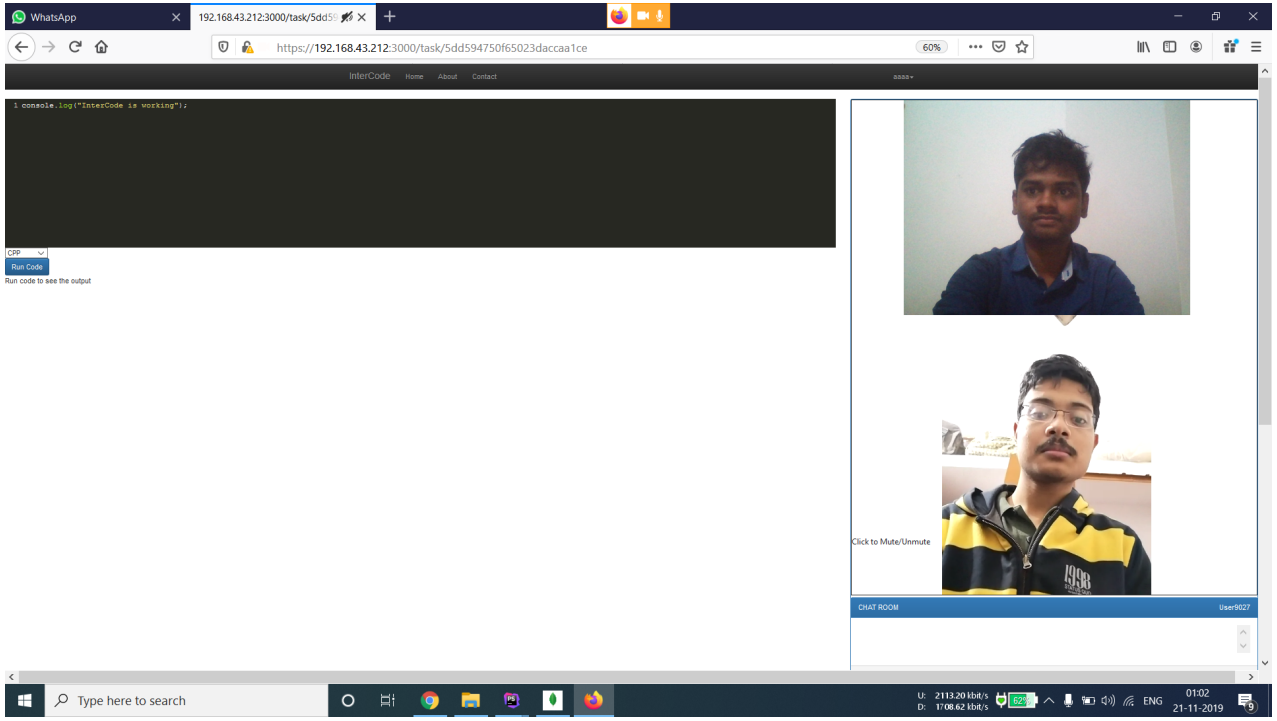

Figure 1: InterCode in action

Registration and Validation

- Registration can be done using two ways: Email or Facebook
- Email registration creates a new MongoDB document with multiple fields like hash,salt etc. that are used during validation
- In case of Facebook registration, only name and email address is stored, and the validation is done using Facebook API
- Usage of SHA-256 encryption for password provides added security

Sign Up Page

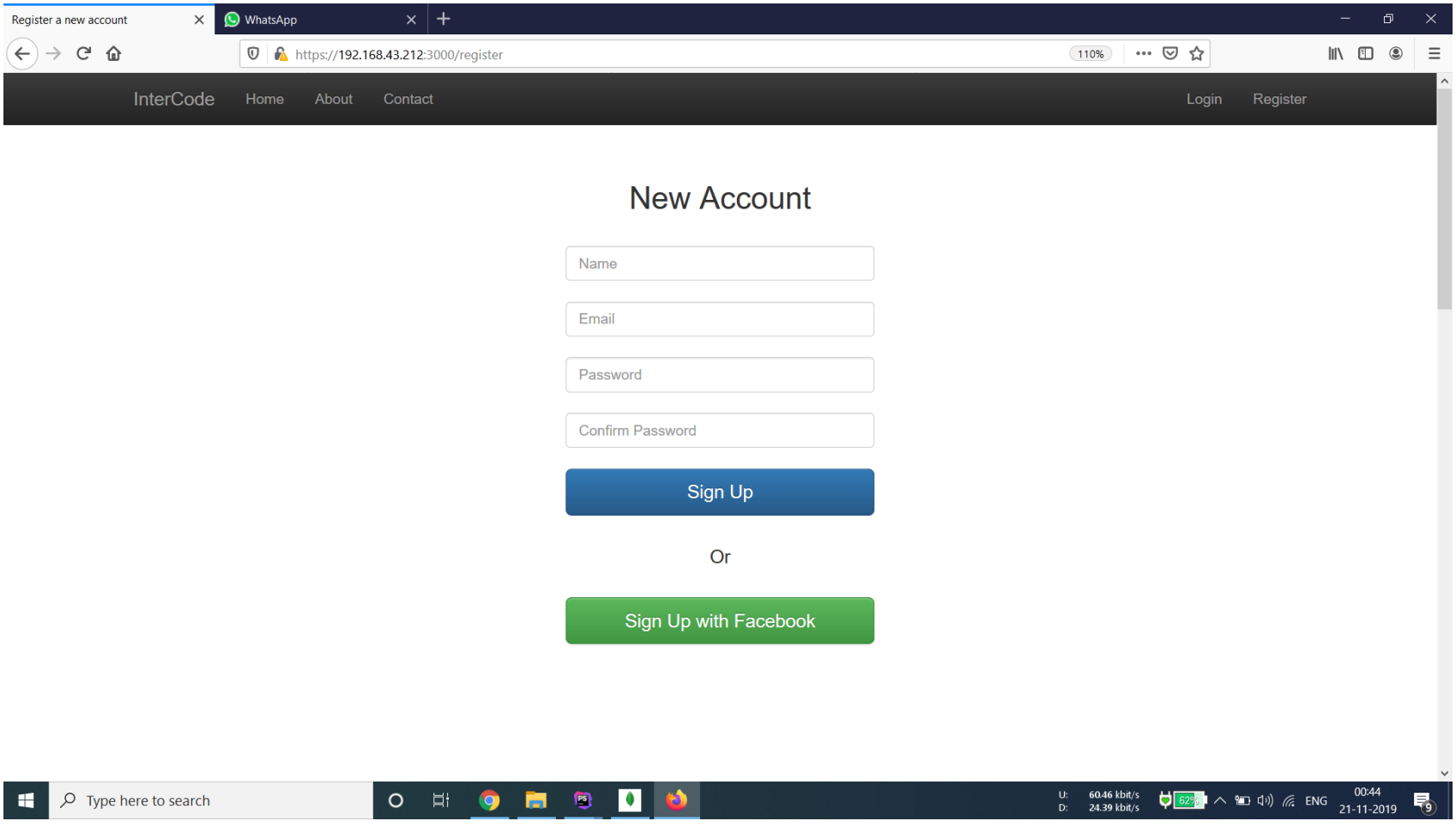

Figure 2: The Sign Up Page

Login Page

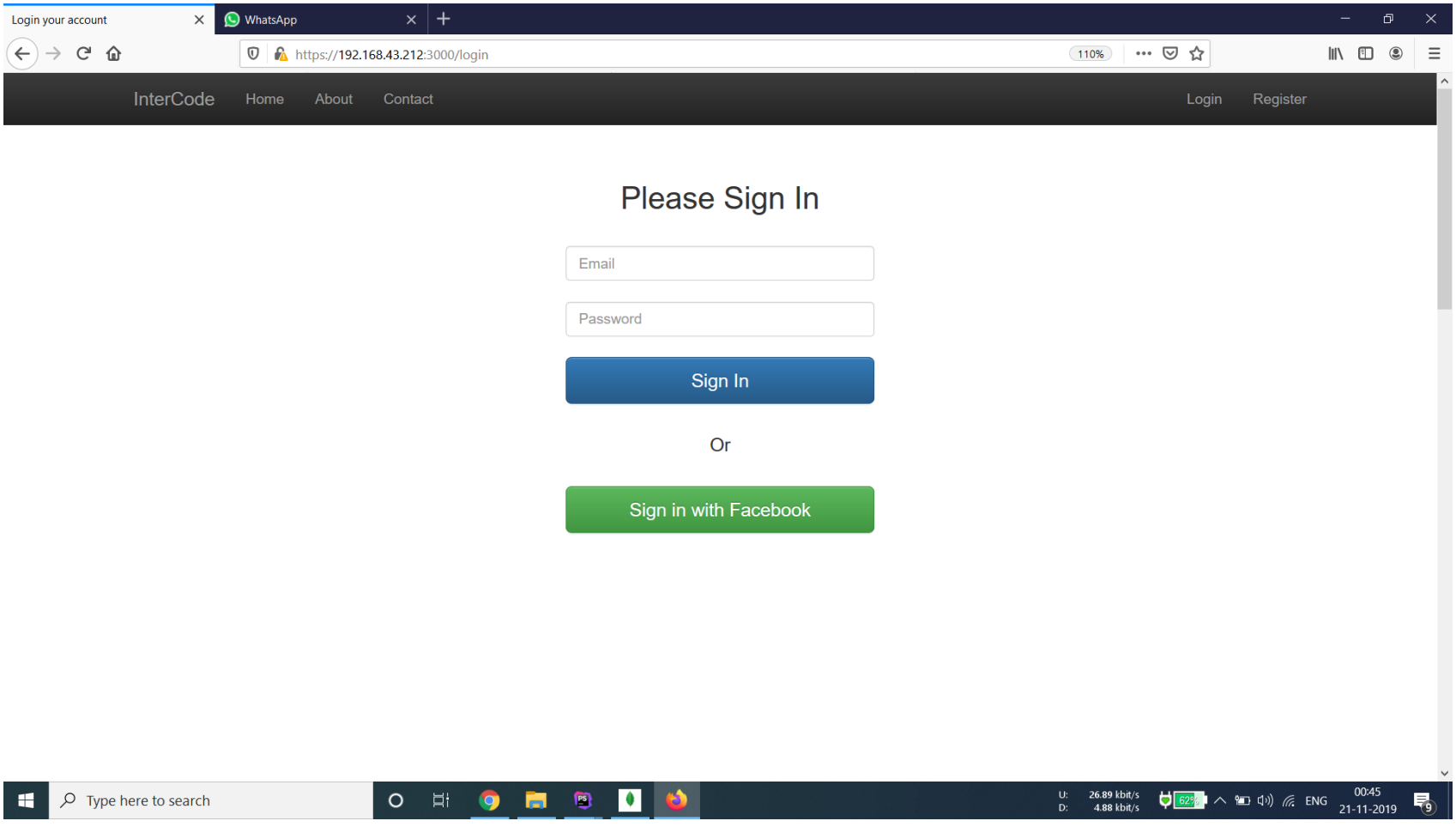

Figure 3: The Login Page

Video Communication

- Implementation achieved by using *webRTC*, an open source project
- *Simple Peer* is the Javascript implementation of webRTC
- Modern web browsers require HTTPS (HTTP over SSL) certificate for allowing video communication
- Separate port allotment for video communication for independent video chatting

Collaborative Code Editor

- Usage of the concept of *Operational Transformation* for collaborative text editing
- CodeMirror package helps achieve code beautification
- Code editor progress saved as database document
- Task defined by unique ID restricts the number of collaborators
- Support for C++ and JavaScript editing and compilation currently

Text Communication

- *Socket.IO* library forms the basis of chat platform on the web application
- Unique Task ID helps create a chat room involving concerned persons
- Each user is assigned an ID, without revealing credentials
- Provision of uploading a resume is added to the feature

Database Description

- MongoDB, a NoSQL database program is used as the database management utility in this web application
- Database serves two purposes in this application: User Details and Task Progress
- User details are stored in a collection in the form of a document, and number of fields in the document varies
- Task progress is stored with Task ID as the unique key, and is constantly updated
- Varied structure of data in a single table creates the requirement of a NoSQL based database program

Scenario

Two of the team members took upon the role of interviewer and interviewee, and simulated a remote interview over a local network. Coding problems were given to verify the working of the editor, whereas simultaneous video communication took place to maintain contact between the two parties.

Collaborative Code Editor

Figure 4: Code editor with C++ in action

Text Communication

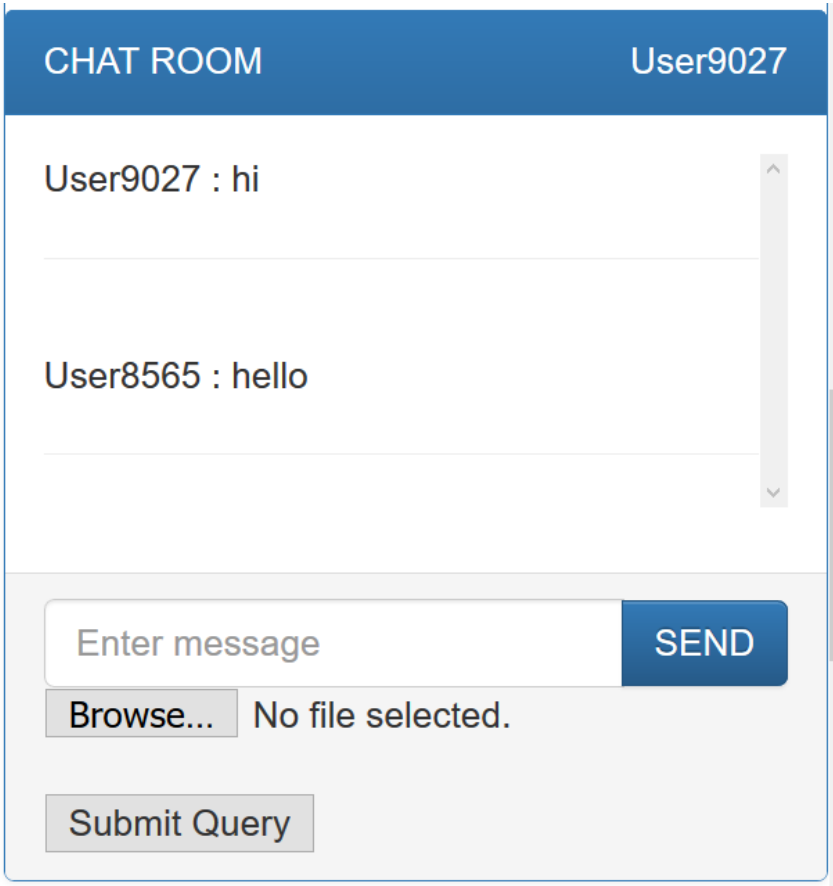

Figure 5: Text chat taking place

Testing and Analysis

- Postman is a collaboration platform for API development
- A Postman utility was used to test the web services of this application
- Particular GET and POST requests were sent and evaluated against expected response
- Code editing and compilation was successful from both ends
- Text messages were sent from both ends

Conclusion and Future Work

Conclusion

The successful development of the web application gives us a utility that will provide a one-stop solution to all requirements of a remote interview and can be used by various organisations and college placement cells to make the recruitment process a little more hassle-free, and hence boost the recruitment.

Future Work

The important work to be done is deployment of this web application on a global network using apps like Heroku, so that everyone can access it over a wide network and make use of its functionalities.