Implementation of Virtual Class using WebRTC

Vinodh Kumar P #1, Punith Kumar N*2, Sunil V.R *3, Srinivas P Anvekar*4, Sanjeetha R #5
Department of Computer Science & Engineering,
M.S. Ramaiah Institute of Technology,
Bengaluru, India

Abstract - **The latest technology of WebRTC has enabled users to develop traditional applications like virtual classrooms in a more efficient way. The legacy applications required installing sophisticated softwares at client and servers which added lot of complexity. Using webRTC, the information can be shared by using only a browser to act as both client and server and transfer text, audio, video and images between them with ease.**

Keywords: WebRTC; Virtual Classroom; ICE; STUN

I. Introduction

WebRTC is an open source project that enables web browsers for Real-Time Communications (RTC) using simple JavaScript APIs. The purpose of WebRTC components is to help build a strong RTC platform that works across multiple web browsers and across multiple platforms. WebRTC Uses the web as the setup and end point for a call. This is different than other calling systems, which use SIP, XAMPP or other signaling systems. It allows site access to video camera feeds for graphically manipulating the video feeds or allowing users to record videos. WebRTC establishes a point-to-point connection between browsers. It also does NAT traversal. Video is displayed as a part of the content on the web browser. WebRTC uses RTCPeerConnection to communicate streaming data between browsers (aka peers), but also needs a mechanism to coordinate communication to send control messages, a process known as signaling. Signaling WebRTC does not specify methods and protocols: signaling is not part of the RTCPeerConnection API. Instead, WebRTC app developers can choose their desired messaging protocol they prefer, such as SIP or XMPP, and any appropriate duplex (two-way) communication channel. The apprtc.appspot.com example uses XHR and the Channel API as the signaling mechanism. The codelab we built uses Socket.io running on a Node server. WebRTC provides the following functionalities : User discovery and communication, Signaling, NAT/firewall traversal, Relay servers in case peer-to-peer communication fails, STUN protocol and its extension TURN are used by the ICE framework to enable RTC Peer Connection to cope with NAT traversal and other network vagaries. ICE will first use STUN with UDP to directly connect peers and, if that fails, will fall back to a TURN relay server.

Abbreviations and Acronyms

- ICE - Interactive Connectivity Framework
- STUN - Session Traversal Utilities for NAT
- TURN - Traversal Using Relays around NAT
- NAT-Network Address Translation
- SDP-Session Description Protocol.

II. Design & Implementation

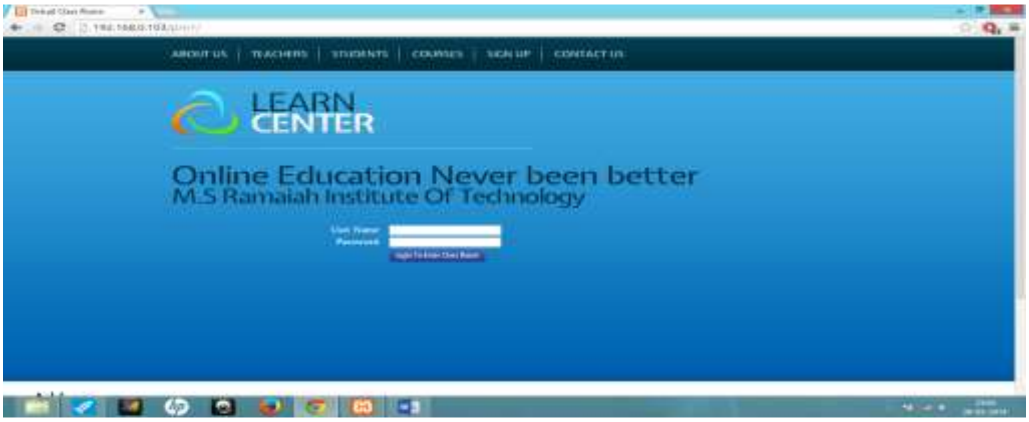
We explore the capabilities of webRTC by implementing a Virtual Classroom. A virtual classroom is an online learning environment that enables student and the instructor to interact as if they were face to face in a real classroom. Virtual Classroom provides flexibility for Students to take up classes at their convenient time and from the comfort of their home. It benefits students who are in rural areas to get assistance from expert professors without incurring travelling or accommodation cost, and students who are currently working to take up courses of their choice.

Using webRTC the audio, video and text data (ppt) is shared between the instructor and students online. It provides two interaction by allowing students also to ask doubts during the sessions. The lecture notes is made available on the website for future reference. WebRTC mirrors the screen of the instructor on student's machine leading to better understanding of the topics. Students can join the live session by using their browsers alone.
WebRTC is designed to send media directly via peers. Often this occurs between two clients, but many applications require the server to act as a media peer. Integrating the WebRTC App in the System consists of creating a server on a LAN and then hosting the site that contains the WebRTC App.

List of API's Used in our project include:

API Name	**Summary**
LocalMediaStream	The LocalMediaStream object is the MediaStream object returned from the call to getUserMedia(). It has all the properties and events of the MediaStream object and the stop method.
MediaStream	The MediaStream interface of theWebRTC API describes a stream of audio or video data. A MediaStream object represents a linear, potentially infinite timeline. It is not preloadable, nor is it seekable.
MediaStreamTrack	A MediaStreamTrack is one of two kinds, audio or video, and represents the media source, such as a camera.
MediaStreamTrackList	A MediaStream has two MediaStreamTrackList objects, one for the video tracks and one for the audio tracks.
RTCDataChannel	Represents a bi-directional data channel between two peers.
RTCPeerConnection	Provides for the connection between remote peers, the transmission of locally generated MediaStream data and arbitrary data between peers.
RTCSessionDescription	Describes the session with a serialized Session Description Protocol object.

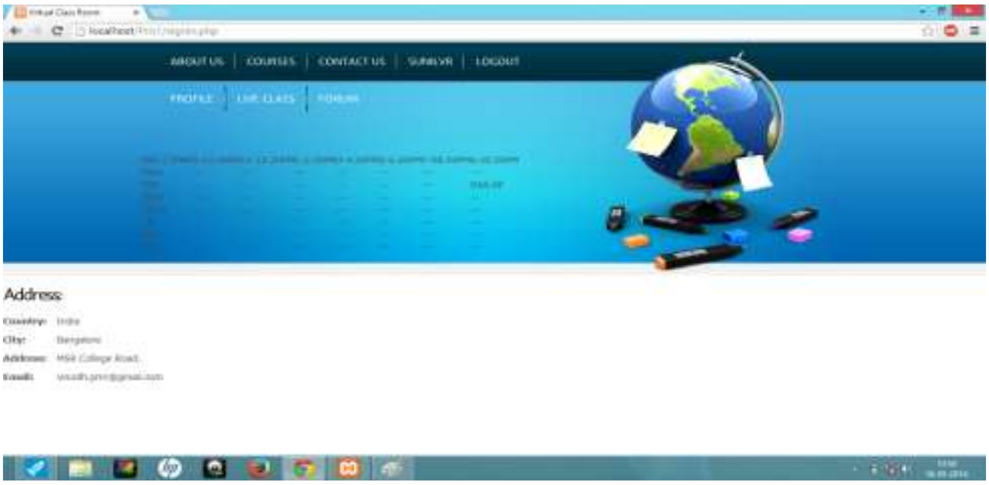
III. SCREENSHOTS



Online Education Never been better
M.S Ramaiah Institute Of Technology

User Name
Password

Figure 1. Screenshot of Login Page


Virtual Class Room
localhost
ABOUT US | COURSES | CONTACT US | SUNILVR | LOGOUT
PROFILE | LIVE CLASS | FORUM

Address

Country: India
City: Bangalore
Address: MSR College Road
Email: [illegible]

Figure 2**.** Screenshot of Timetable Page

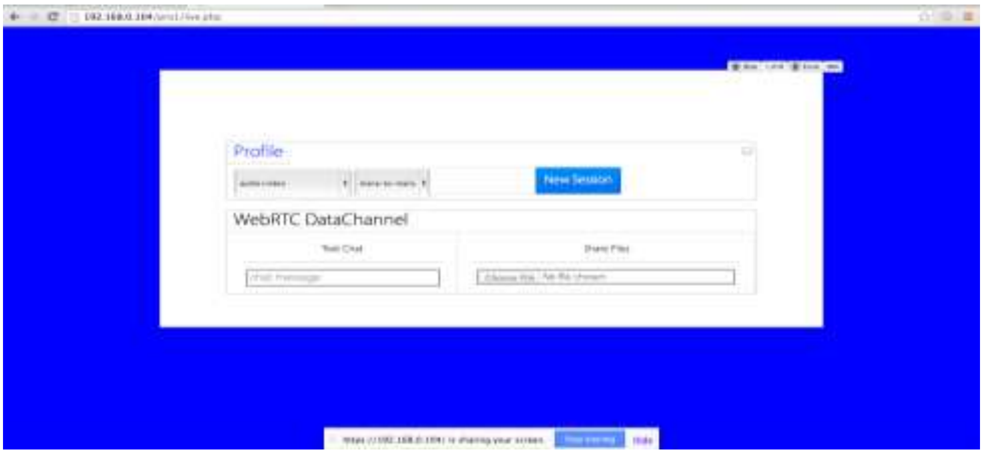
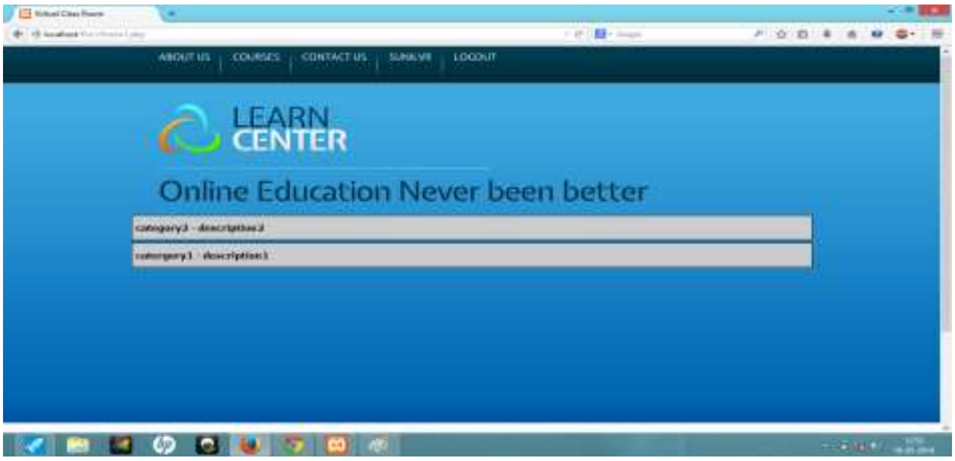


Figure 3. Screenshot of Live Class


Virtual Class Room
localhost
ABOUT US | COURSES | CONTACT US | SUNILVR | LOGOUT
LEARN CENTER
Online Education Never been better

category2 - description2
category1 - description1

Figure 4. Screenshot of Forum Page

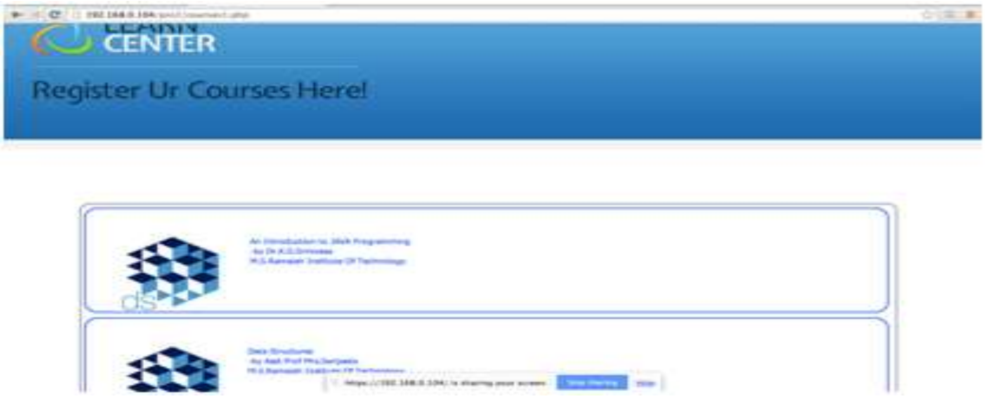


Figure 5. Screenshot of Courses Page

IV.CONCLUSION.

WebRTC is a new technology that will help to develop applications with ease. The APIs and standards of WebRTC can democratize and decentralize tools for content creation, communication and many other applications. We were successful in implementing this technology with the legacy Virtual Classroom application.

V. FUTURE SCOPE

WebRTC can be used to implement applications on smartphones. It can also be used for applications that can be deployed on cloud.

ACKNOWLEDGMENT

We acknowledge the support and help provided by M.S. Ramaiah Institute of Technology, Bengaluru. We are also thankful to Karnataka State Council for Science and Technology, Indian Institute of Science Campus, Bengaluru for funding this project under the 37th series of Student Project Programme.

REFERENCES

[1] http://educationportal.com/articles/The_Differences_Between_Online_and_Traditional_Classroom_Educations.html
[2] http://techpegasus.blogspot.in/2012/04/disadvantages-of-traditional-classroom.html
[3] http://seacstudentweb.org/top-benefits-of-online-education.php
[4] http://www.html5rocks.com/en/tutorials/webrtc/basics/
[5] http://en.wikipedia.org/wiki/Real-time_Transport_Protocol
[6] http://en.wikipedia.org/wiki/STUN
[7] https://developer.mozilla.org/enUS/docs/Web/Guide/API/WebRTC/WebRTC_architecture
[8] http://en.wikipedia.org/wiki/World_Wide_Web_Consortium
[9] http://en.wikipedia.org/wiki/IETF
[10] en.wikipedia.org/wiki/Session_Description_Protocol
[11] http://en.wikipedia.org/wiki/Secure_Real-time_Transport_Protocol
[12] http://www.html5rocks.com/en/tutorials/webrtc/basics/#toc-more
[13] www.codecademy.com/
[14] http://www.cdacmumbai.in/design/corporate_site/override/pdf-doc/virtual-classroom.pdf
[15] http://www.w3.org/TR/webrtc
[16] http://www.w3.org/TR/webrtc/#rtcpeerconnection-interface.