RESEARCH ARTICLE OPEN ACCESS

RASSEMBLER- LET’S MEET AN ONLINE VIRTUAL MEETING PROJECT

A.Pawan Krishna reddy [1], Pathakota prasanth reddy [2], Madagala Lakshman Sai [3],Thodeti Rajashekar [4], Tammineni Gopal Reddy [5], Parampreet kaur [6]

[1],[2],[3],[4],[5],[6] School of Computer Science and Engineering, Lovely Professional University,Jalandhar,Punjab - India

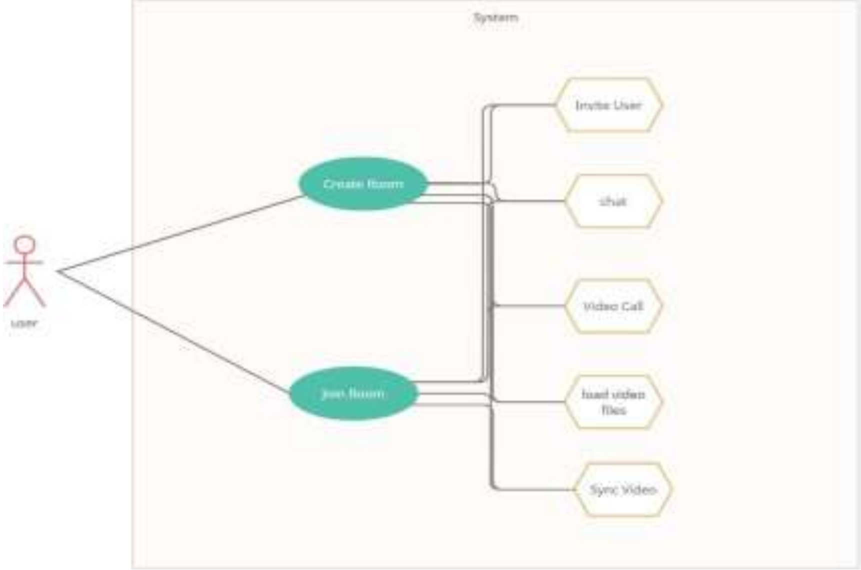
ABSTRACT

Rassembler let’s meet is an online video file synchronization tool, which enables users to enjoy the company of their friends in the "online cinema" setting, which provides a way for people to watch TV or movies to get. The best social networking website allow users to watch popular videos and live chat with friends online. The user can make a video call while watching videos. By using this website, friends can chat and they can share videos and enjoy them watching together. In Rassembler let’s meet you can stream videos from multiple platforms. As always, there are already websites and apps that serve the same purpose. However, they have different goals and have not left us completely satisfied. Many OTT platforms does not have the content to view together. Other websites does not allow vimeo player, mx player, mp5 video, but in our website we are providing these facilities, and also user can control following things such as they can make the video pause, rewind, forward and play.

Keywords: Virtual Theater, Video Synchronization, Sync play, watch together, video chat.

I. INTRODUCTION

Rassembler let’s meet is a real online video synchronization platform. You can enjoy any video available online with friends who may not be near you. The best social networking website allow users to watch popular videos and chat live with friends online. With the help of this website, you can enjoy any video while chatting with your friend and you can stream videos from multiple platforms that you can enjoy with your partners. The user can make a video call while watching videos. Virtual Theater currently supports YouTube, Daily Motion, Vimeo, and especially any .mp4 /. web online with HTML5 Player. Many OTT platforms do not have the option to view content together. Some websites like this do not offer the video call feature. This project will provide better security to avoid some of the web attacks that are currently occurring on most web applications. First the User needs to create a room by entering his or her name and room id. After entering the room, you find a room with a video player. With chat option and video call. in the video player area, the user needs to enter the video link and upload the URL. The user has the option to copy the room link and can share it with others to join the room. when another user joins the same room, it shows all users present in the room. We can watch the video in synchronization and chat with each other.



II. LITERATURE REVIEW

[1] A research paper on “Are we synced? Synchronization requirements for online video viewing in collaboration”by David Geerts and Oskar van Deventer state that Synchronization between sites is critical to enabling remote sharing. However, test data for acceptable syncing levels are scarce.This paper discusses the synchronization requirements for watching online videos together - a collection of popular services that evoke a shared feeling of watching TV together by providing communication tools while watching. It learns the visibility and annoyance of visual video sync, as well as the impact on users' feelings of being together, in voice chat and text chat.The test results of 36 participants show that when using voice chat, users notice a faster syncing difference, become more irritated and feel more connected than when using a text chat. However, users with high-text chat functions recognize syncing similarities similar to those using voice chat.

[2] A research paper on A History and Future Web API by Jacek Kopecky states that widely distributed information systems have customer structures, such as the Web itself. In this article, we review the emergence of a visual interface for client-distributed programs, from pre-programming and RPC pre-Web RPCs, to RESTful Web APIs. We emphasize the general importance of ignoring the client-server interface in Web applications, and looking at historical and current systems to discuss the technical roles of "Web Service" and Service-Oriented Architectures. Looking to the future, we point to four indicators where we can see moving Web APIs, including hypermedia input and semantics.

[3] A research paper on Web Real-Time Communication Surveys Based on Web Socket by Qigang Liu, Xiangyang Sun says Regarding the limitations of real-time web communications solutions such as voting, long-term voting, flash plug-in, suggest using Web Socket new upcoming technology in the field of real-time web communications, introduces features of Web Socket technology, analyzes differences between Web Socket protocol and HTTP protocol, provides a way to use Web Socket on both the customer and server side, proving that Web Socket can significantly reduce network traffic and delays with testing, it has given rise to promising future use of Web Socket in real-time web communications.

[4] A research paper on Collection of the video and its use at the beginning by K. Rop1 and NK. Bett2 says Over the past decade, video conferencing (VC) has become very popular and reliable as a tool to bridge the gap where travel is not an option, not possible or unwanted. Video conference uses audio and video communication to bring people together in different places. The insights needed to make videoconferencing and its application become one of the major topics researched by various educational institutions and entrepreneurs. In this paper, a presentation of video conferences is presented with an emphasis on its use in distance learning.

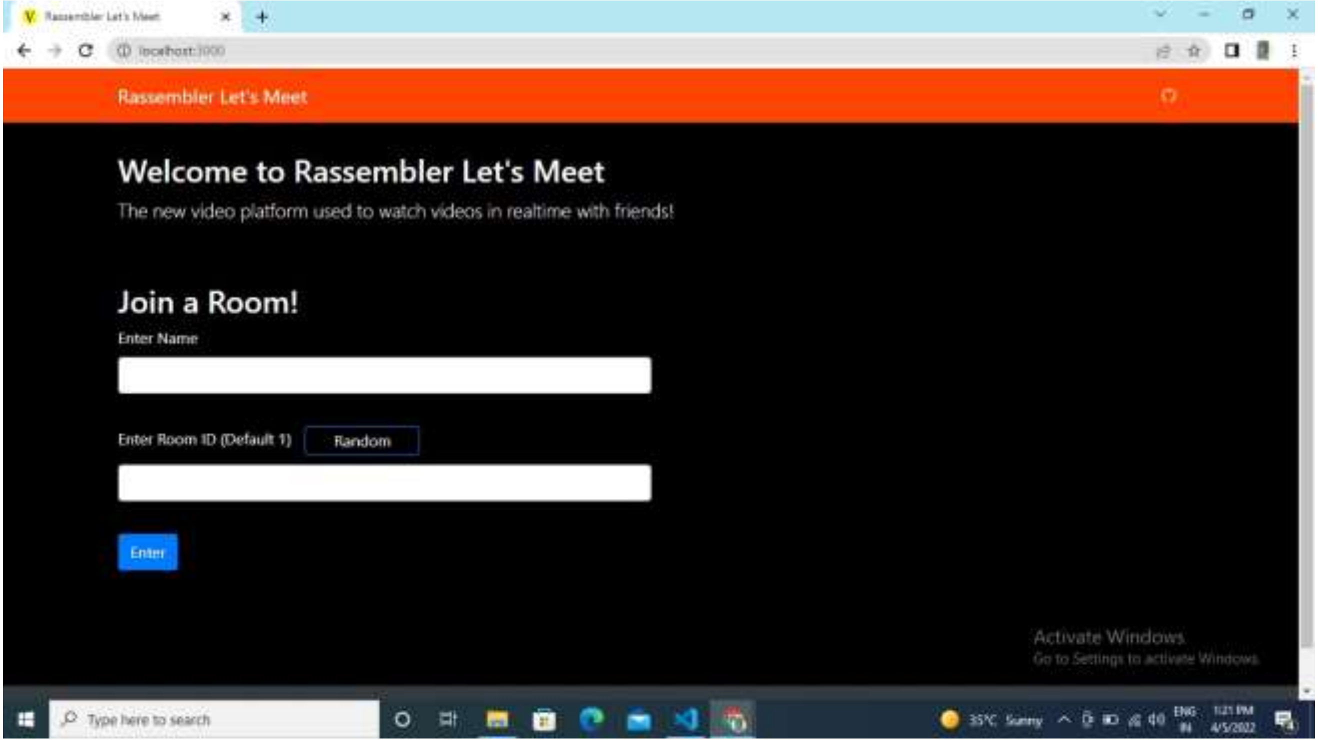
[5] A research paper on Text Chat in Action by Jacki O'Neill and David Martin states that Synchronous texting has become known as an important communication tool in the workplace but some studies in group text writing suggest that its structures can lead to collaborative collaboration. We address this issue with a detailed examination of the text of a text discussion by showing how participants manage their connections by considering multiple threads, turn-by-turn and topic changes. It outlines common practices that participants use to create and manage related communications. These practices stem from the opportunity-taking system, which helps to directly address the conflict. We conclude by looking at the effects of this on the design and planning and management of interactions of various kinds.

[6] A research paper on Testing the security of the Node.js platform by Andres Ojamaa and Karl Düüna claim that Node.js is a network-based application platform for developers forcing developers to use asynchronous programming connectors in I / O applications. The traditional language for application development in this platform is JavaScript. Although still very young the platform has attracted a significant community of developers and gained support from the industry. The Node.js community is generally very focused on the decline of the platform. However, little research has been done on how platform-making decisions affect the security of its applications. This paper explains the many security pitfalls that can be used when using the Node.js platform and the JavaScript server. We also describe the two risks identified and provide recommendations for the development and maintenance of secure and robust web applications on the Node.js platform.

III. METHODOLOGY

Create Rooms

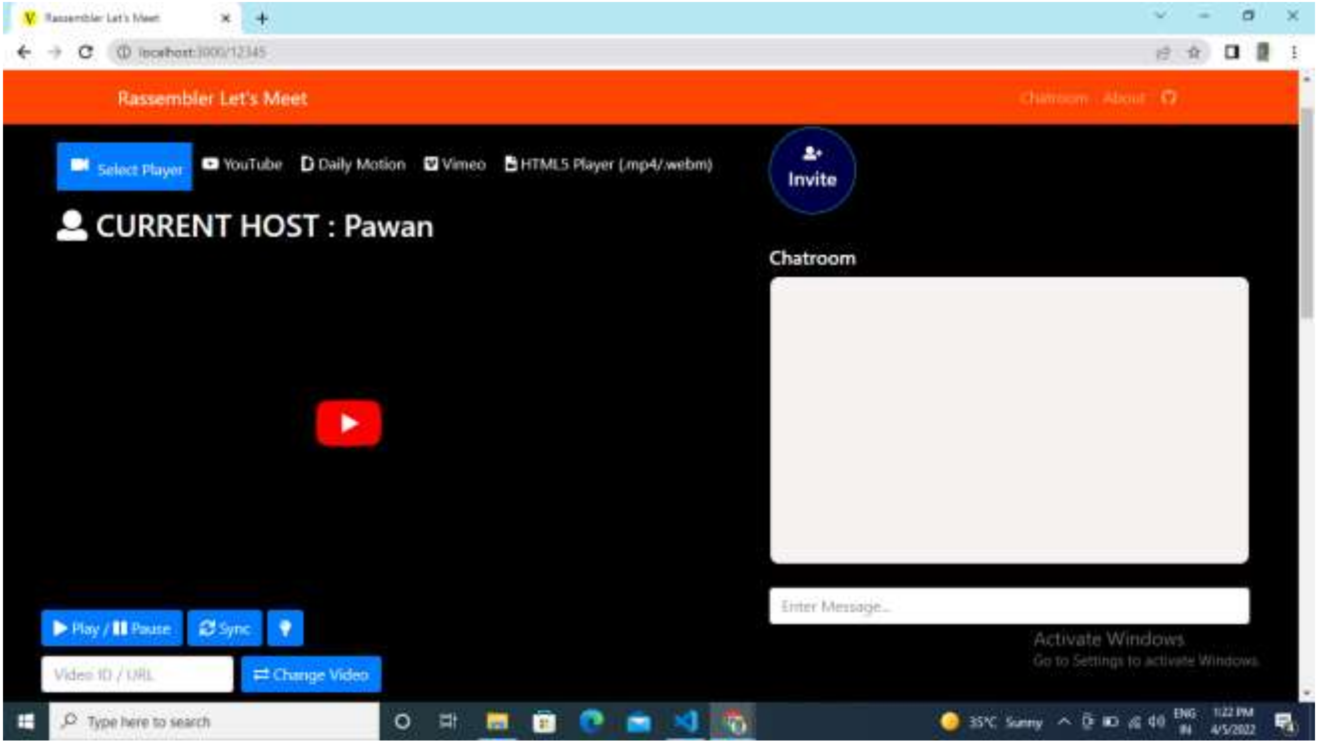
All functionality depends on web sockets, especially Socket.IO. When a client connects to a server, a socket is created. The user then enters the name and room number. Input is sent back to the server, and creates / joins a room for that name with Socket.IO. Any user can connect to the room and interact with users there. Socket.IO services can only be disconnected from certain rooms. In this way users in a particular room can call a function and have only one that touches their room. This provides the basis for performance.



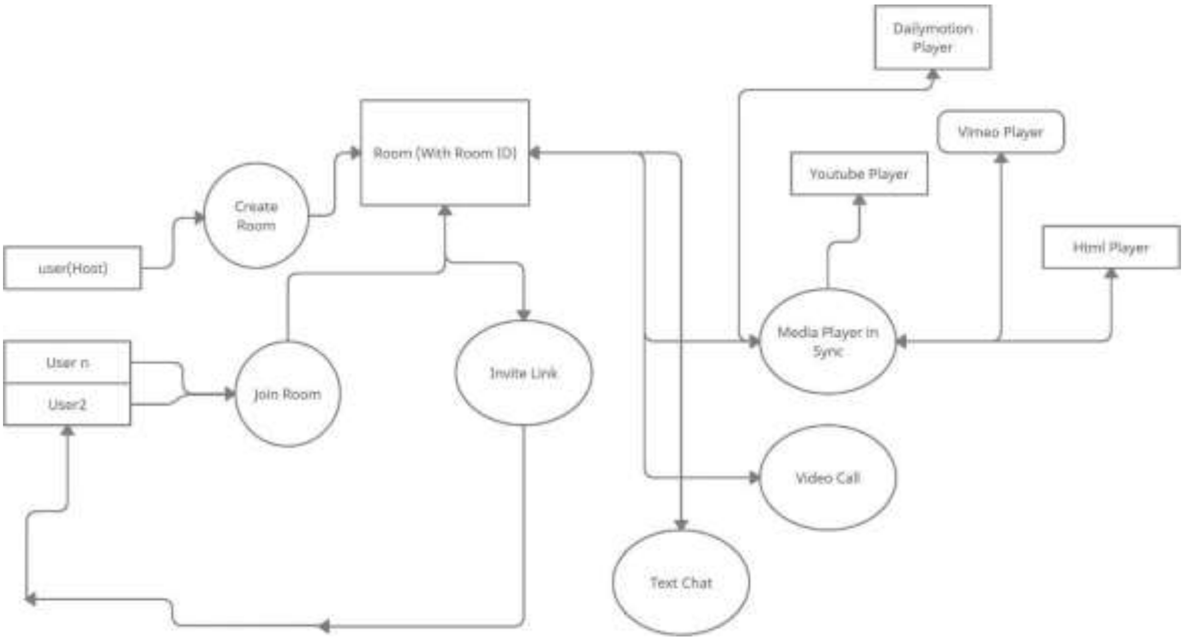
Welcome to Rassembler Let's Meet

The new video platform used to watch videos in realtime with friends!

Join a Room!



CURRENT HOST : Pawan



Functionality

Sync operation simply controls the video player, and calls for the same functions for each socket in the room. For example, if one calls for play, it will call for play for all connected bases. When someone calls a sync, it will retrieve the current time from that user only and send the details to all the bases. It will use that data and deliver it to everyone in a timely manner.

Hosts

At first it was nice to have unoccupied rooms, but I quickly realized that people want automatic syncing rather than hitting the sync button over and over again. For example, when you join an existing room, you want to jump on content rather than worry about syncing. To do this I made a host socket to be marked when the room was built. This selector was able to send all important video details to any new connectors. Socket.IO rooms have excellent flexibility that can hold certain room details.

io. sockets. adapter. rooms ['room -' + roomnum]. host = socket.id

As well as capturing the hosting details of this item and owning the current video, player, and connected customers.

Video players

At first, the YouTube API was the simplest and most popular platform.Additionally, every video player has a different API and works differently.I started playing around with the Daily Motion API, and I had to completely rearrange how I show the player. What I did was just start all the players and ruin / hide the player when needed. While this operation seemed to have a lot of bugs.This was probably the hardest part so far. It is because of all the asynchronous wires that damage other foundations. I finally found a way to add all the possible data to the home and bedroom item. Any new sockets joining or syncing will have all the data already uploaded and synced automatically. Even if you play a YouTube video and change players, your continuation of the original YouTube video will be saved and you can return to it at any time. This sets the basis for many video players in the future. I hope to use it soon! One of my favorite features would be to be able to stream videos to any link, but that might not be my strength yet.

Messaging System

The messaging system is simple messaging service where users can communicate using text. There is a key section on file for a messaging system that makes texting and instant messaging possible in real time. Socket.io is easy on the client and a real time messaging server. User messages wants to send to recipient sent to the server there is socket.io it actually comes in handy. Just after the server is updated with a new message, socket.io will work in real time and send you messages according to the message again where the message should be sent. Messages are standard text messages. All these messages are stored on the browser itself. Socket.io will be able to access the file data using a piece of code. It integrates and allows socket.io to do its job.

Video Call

The website also provides an interface allowing users to communicate via video calls. This is the case for the file messaging interface, which uses io and webrtc socket to use. Like real time messaging module, this video-call module is also based on io extension. The code should be spelled correctly and some space should be provided on the website where video clips occur. According to privacy issue, video calls will not exist recorded and not saved in any file website modules.

Implementation of the project

We started our project using Node JS, Express, Socket IO and Tok Box API. Node JS made it easy to manage the back end of the program, but our backend also consist the YouTube data API, Dailymotion API and Vimeo API integrated into Express JS and then added to the Node JS server. We used a lot of frontend technology to do this project including Html, CSS, JavaScript, jQuery and bootstrap.

Limitations and future work

Many questions are raised in our studies. First, although we have found one way to reduce the feelings of frustration in combining conversation and video, there are many more. One can use a clear interface design, for example, by setting the start time of a conversation at the most convenient times by linking a chat with a videowindow.The main focus after launch is to expand the user base and get more involved and many people in our project, as our entertainment-based project provides a great platform to enjoy the best videos viewed on the website in sync with friends. SEO and project marketing are the main concerns about the use of posts. To do website and popular is very important in our project, most are no. of users, more we will lead. To maintain the website and its use, one of our team members is reviewing the website on a daily basis, members of our team are also writers, so any issues or issues arising from the website or software is being tested and repaired there. Being users again Authors have helped us identify our software from the user, and this helps us make better changes to the software.

IV. CONCLUSION

New peer-to-peer video streaming technology promises to fundamentally change the way we support the media. The experience will no longer be artificial, or limited to the social boundaries set by the world itself. Watching video online enables us to engage actively as we engage with video, but active engagement comes at a cost. Attention is a limited service, and we have shown that while chatting and video calling and watching at the same time are fun, and there are social benefits, it is also distracting. Logging in can help reduce this distraction, or it may not be appropriate for all types of content. Therefore, future research is needed to gain a better understanding of the viewing experience, and to understand what factors contribute to a more effective and collaborative experience.

REFERENCES

[1]. "David Geerts and Oskar van Deventer.Are we synchronizing? Synchronization requirements for collaborative online video viewing, Procedures for the International Conference on Human Resources in Computing Systems, CHI 2011, Vancouver,BC,Canada,May7-12,2011"

[2]. "Jacek Kopecky,"A History and Futureof WebAPI, January 2014it - Information Technology, University of Portsmouth, UNITED KINGDOM"

[3]. "Qigang Liu and Xiangyang Sun, „Web Real-Time Communication Surveys Based on Web Socket" January 2012International Journal of Communications, Network and System Sciences 05(12):797-801

[4]. "K. Rop1 and NK. Bett2, Collection of the video and its use at the beginning, Annual Interdisciplinary Conference, The Catholic University of Eastern Africa, Nairobi Kenya. Volume: 1"

[5]. "Text Chat in Action by Jacki O'Neill and David Martin, Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work, GROUP 2003, Sanibel Island, Florida, USA, November 9- 12, 2003"

[6]. "Testing the security of the Node.js platform by Andres Ojamaa and Karl Düüna,2012 International Conference for Internet Technology and Secured Transactions, IEEE London UK"