EXPOSYS DATA LABS

ONLINE INTERNSHIP ON

VIDEO CHATTING WEBSITE

Submitted by: MOHAMED AWEEZ AKRAM (1HK17CS079)
MOHAMMED ADNAN K (1HK17CS085)
NITHISH KUMAR R (1HK17CS107)
NAVEEN N (1HK17CS100)

EXPOSYS DATA LABS

P.M.R Residency, Ground Floor, No-5/3 Sy. No.10/6-1 Opp Nithyotsava Wedding Hall, Doddaballapur Main Road, Singanayakanahalli, Yelahanka, Bengaluru, Karnataka 560064.

ABSTRACT

ChatAway is a website where you can connect with people across the world and have a one-on-one video chat with users. The technology used here is Web Real-Time Communication also known as WebRTC. The idea behind the development of this project is to use the existing technology to bring people and ideas together in terms of face-to-face interaction despite of the geographical barriers. Our project serves as an example of Web Development or Web Technology. In ChatAway, we will have two users who communicate with each other. When you login, the website asks for permission to use the webcam and microphone. The client then initiates a connection with the server. After which the client is able to send a request for video/voice call to the desired online user and receive the same.

TABLE OF CONTENTS

INTRODUCTION

The demand for live face-to-face interaction between people is increasing day by day. A video chat website is a platform that allows you to interact in live with people, which is our primary inspiration for this project. The goal of our project is to build an online video chatting website that enables users to join real-time streaming video chat rooms where users can share their video with another user. Users can send instant messages and share their live web cam data with another user in the chat room.

Some of the existing video chat websites are CooMeet, CamSurf, Gotalk, Omegle, etc. Our online video chat website offers similar functionality like these social networking sites, and it is simple to execute and doesn't have to rely on any third-party sites. Users can directly enter webcam chat rooms after signing up. Also, this video chat website doesn't require any additional installation on the client side. By just accessing our website, users can have a video connection from both the ends.

The main technology used in this project is the Django Framework. Django is a python framework used for the development and deployment of cross platform, rich Internet websites based on existing web technologies.

EXISTING METHODOLOGY

Most of the existing famous methodologies are developed as software applications. Unlike those applications, we have developed this as a complete website so that the user can access through any web browser. Since most of the communication at present happens through e-mails or chat applications, our website ensures that users can have communication with each other through live face-to-face interaction along with sharing of data. Thus, we make sure that user can save an adequate amount of time through virtual communication than being in a physical communication.

The video chat methodology is emerging since the rise in the emergence of virtual communication. Thus, the technology implemented here is WebRTC or Web Real-Time Communication which delivers an efficient way of communication through video/voice call enabled chat API. The popularity of WebRTC has made sure that browsers deliver peer-to-peer communication through video and audio calls in real-time. WebRTC was built in a great accordance with the developers, which provides the ability to deliver high pixel-rate audio and video chats. In simple words, every WebRTC endpoint would have a unique address that it could exchange with other peers in order to communicate directly.

The protocol used in developing this website is the HTTP/HTTPS protocol. Other than this we use SIP which is a signaling protocol used by VoIP and video conferencing systems. To enable communication between a WebRTC website and a SIP client such as a video conferencing system, WebRTC needs a proxy server to mediate signaling.

PROPOSED METHODOLOGY WITH ARCHITECTURE

ChatAway website has been developed with an idea to make video conferencing as easy as possible. It should be quite easy for the users to use our website across the world for one-on-one video/voice call. It specializes in providing video chat and voice calls between computers, tablets, smartphones over the Internet. It also provides instant messaging services. You can see each other via webcam, hear and communicate through a microphone.

At present video chatting has become quite popular. It has become a daily need for people who stay far away from each other. In the Internet, there are a lot of both paid and free video chat websites. Our service is also completely free to use with a focus on safety, privacy and security.

It is pretty simple to use our website:

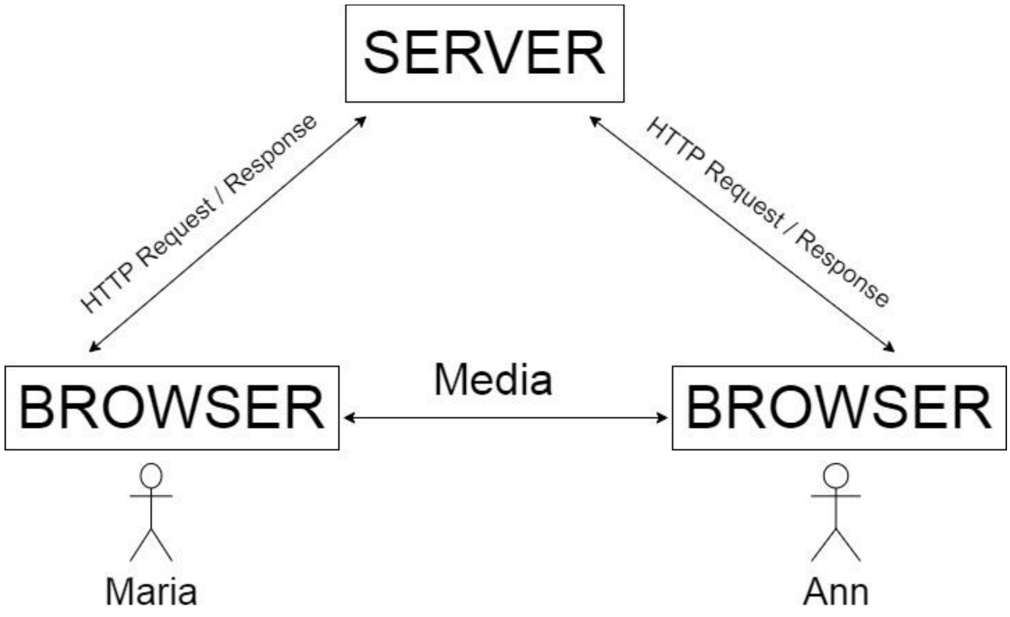
1. Enter the URL.
2. Signup (if you don't have an account).
3. Login and you will be able to see your friends who are online.
4. Click on your friends' name and the chat dialog box opens up and you can start messaging him/her instantly.
5. There is an option to start the audio/video calling in the top right corner of the chat box and that's it there you go!

In this project, we have used different tools like Django, Bootstrap 4, WebRTC. With WebRTC, you can add real-time communication capabilities to your application that works on top of an open standard. It supports video, voice, and generic data to be sent between peers, allowing developers to build powerful voice and video-communication solutions. The technology is available on all modern browsers as well as on native clients for all major platforms.

The technologies behind WebRTC are implemented as an open web standard and available as regular JavaScript APIs in all major browsers.

Bootstrap is a free and open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS and (optionally) JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components.

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. It makes our website a bit more secure and also has the predefined fuctionalities which helps new user while signing up to create a secure password which makes an individuals account much safer. Ultimately Django is ridiculously fast, reassuringly secure, exceedingly scalable.



METHODOLOGY

The user interacts with the website using a GUI or simply known as Graphical User Interface which is basically needed for the user to initiate the communication from the sender to the receiver. So, let's discuss the process encountered when the user visits the website.

- The GUI operates in the form of a chat box along with the video display of the user.
- When the user gives permission for the camera and microphone, they can start searching for the desired user in the list.
- The list will contain all the users who are present online.
- The user can select any particular receiver for initiating the communication.
- If the receiver accepts the request, they can start communicating with each other.
- When either of the user want to end the communication, they can press the 'end call' button.

Approach: The website is developed using most popular Web Framework known as Django. The object-oriented method of it serves better for storage of a particular entity.

Functional Requirements:

The primary requirement of our video chatting website is to provide a mechanism for virtual online face-to-face communication on a small scale. It will also provide a feature for the users to exchange messages through the use of a chat box.

Non-Functional Requirements:

Hardware Requirements:

- Operating System: Windows family, macOS, Linux, etc.
- Browser: Chrome, Firefox, Opera, Brave, etc.
- RAM: Minimum 2 GB.
- Processor: Intel Core (1.90 GHz)

Software Requirements:

- Languages used: HTML, Javascript, CSS, Python
- FrontEnd: Django Framework, Bootstrap 4
- BackEnd: SQLite
- Technology: PeerJS

IMPLEMENTATION

ChatAway is implemented using Django Web Framework. Bootstrap, HTML, CSS are used for the front-end and SQLite is used for the backend. Django is a framework for creating web applications. Django provides us with libraries, tools, etc. that help us to build a web application faster and with less code.

Prerequisites:

- Python 3.0 or newer.
- A code editor like VSCode (preferably), Sublime Text, etc.
- Web browsers like Chrome, Brave, Firefox, Microsoft Edge, etc.

Installation and Setup:

Step 1 - Install and configure Python 3.0 or above and pip 19.0 or above

Install the latest Python version from here.

Verify the version of Python using the command:

- python --version

Also verify the version of pip using the command:

- pip --version

Step 2 - Install and configure virtual environment

First, install the virtual environment wrapper. (in command prompt)

- pip install virtualenvwrapper-win

Then create the virtual environment where Django is to be installed.

- mkvirtualenv environment_name

Step 3 - Install and configure Django

To install Django, execute the command below:

- pip install django

Verify the version of Django using the command:

- django-admin --version

Execution:

Step 1 - Open the folder named ‘video_chat_app’ in VSCode or any other code editor.

Step 2 - Activate the virtual environment in CMD or terminal of VSCode using the command:

- workon environment_name

Step 3 - Run the python server using the command:

- python manage.py runserver

After the server runs, follow the link 127.0.0.1:8000 to the website.

Client connecting to the server:

```
function startPeerClient(username) {

// TODO - Set title

cookie.set('username', username);

peerClient.connectToServerWithId(username);

}
```

Updating online users list every 5 seconds:

```
updateOnlieUsers : function (users) {

var list = $('.onlinepeers')

list.empty()

if(users.length == 0) {

var usr = '<li>Looks like no one is online</li>'

list.append(usr);

return
}

for (var i = 0; i < users.length; i++) {

var usr = '<li class="peeruser">'+ users[i] + '</li>'

list.append(usr);

}

}
```

For streaming from the user side and client side, video call page:

```
setTheirVideo : function (stream) {

var video = document.getElementById('their-video');

if (typeof video.srcObject == "object") {

video.srcObject = stream;}

else

{

 video.src = URL.createObjectURL(stream);

}

};

setMyVideo : function (stream) {

 // $('#my-video').prop('src', stream);

var video = document.getElementById('my-video');

if (typeof video.srcObject == "object")

{

video.srcObject = stream;}
else

{

video.src = URL.createObjectURL(stream);

}

}
```

Login:

```
<form method="POST" action="">

{% csrf_token %}

<div class="input-group mb-3">

<div class="input-group-append">

<span class="input-group-text">

<i class="fas fa user"></i></span>

</div>

<input type="text" name="username" placeholder="Username..." class="form-control">

</div>

<div class="input-group mb-2">

<div class="input-group-append">

<span class="input-group-text">

<i class="fas fa-key"></i></span>

</div>
```

```
<input type="password" name="password" placeholder="Password..." class="form-control" >

</div>

<div class="d-flex justify-content-center mt-3 login_container">

<input class="btn login_btn" type="submit" value="Login">

</div>

</form>
```

Registration:

```
<form method="POST" action="">

{% csrf_token %}

<div class="input-group mb-3">

<div class="input-group-append">

<span class="input-group-text">

<i class="fas fa-user"></i></span>

</div>

{{form.username}}

</div>

<div class="input-group mb-2">

<div class="input-group-append">

<span class="input-group-text">

<i class="fas fa-envelope-square"></i></span></div>

{{form.email}}
```

```
</div>

<div class="input-group mb-2">

<div class="input-group-append">

<span class="input-group-text"><i class="fas fa-key"></i></span>

</div>

{{form.password1}}

</div>

<div class="input-group mb-2">

<div class="input-group-append">

<span class="input-group-text">

<i class="fas fa-key"></i></span></div>

{{form.password2}}

</div>

<div class="d-flex justify-content-center mt-3

login_container">

<input class="btn login_btn" type="submit" value="Register Account">

</div>

</form>
```

CONCLUSION

ChatAway is developed for video chatting and it has been implemented according to the latest technology and requirements. Thus, it ensures that this system will be quite useful whenever we need a remote connection from different places around the world for a hassle-free communication. It needs valid credentials of the user for authorization and verification purposes, so that the user can register and login to his account for initiating the communication. The website is developed and implemented using WebRTC which is the latest among technologies for communication through mobile applications and web browsers. Hence, we would like to conclude that any user who wants to make a video call instantly can make use of our website.