# Project Proposal

## Audio Video Call with WebRTC(Web Real-Time Communication)


Group Members

K180350 - Ibrahim
K181190 - Harris Khan

# INTRODUCTION

**WebRTC** (Web Real-Time Communication) is a technology which enables Web applications and sites to capture and optionally stream audio and/or video media, as well as to exchange arbitrary data between browsers without requiring an intermediary. The set of standards that comprise WebRTC makes it possible to share data and perform teleconferencing peer-to-peer, without requiring that the user install plug-ins or any other third-party software.

WebRTC consists of several interrelated APIs and protocols which work together to achieve this. The documentation you'll find here will help you understand the fundamentals of WebRTC, how to set up and use both data and media connections, and more.

# SCOPE

The **Media Capture and Streams** API, often called the **Media Streams API** or **MediaStream API**, is an API related to WebRTC which provides support for streaming audio and video data. It provides the interfaces and methods for working with the streams and their constituent tracks, the constraints associated with data formats, the success and error callbacks when using the data asynchronously, and the events that are fired during the process.

# CONCEPTS AND USAGE

The API is based on the manipulation of a `MediaStream` object representing a flux of audio- or video-related data. See an example in Get the video.

A `MediaStream` consists of zero or more `MediaStreamTrack` objects, representing various audio or video **tracks**. Each `MediaStreamTrack` may have one or more **channels**. The channel represents the smallest unit of a media stream, such as an audio signal associated with a given speaker, like *left* or *right* in a stereo audio track.

`MediaStream` objects have a single **input** and a single **output**. A `MediaStream` object generated by `getUserMedia()` is called *local*, and has as its source input one of the user's cameras or microphones. A non-local `MediaStream` may be representing to a media element, like `<video>` or `<audio>`, a stream originating over the network, and obtained via the WebRTC `RTCPeerConnection` API, or a stream created using the Web Audio API `MediaStreamAudioSourceNode`.

The output of the `MediaStream` object is linked to a **consumer**. It can be a media element, like `<audio>` or `<video>`, the WebRTC `RTCPeerConnection` API or a Web Audio API `MediaStreamAudioSourceNode`.

# CONCLUSION

The WebRTC standard covers, on a high level, two different technologies: media capture devices and peer-to-peer connectivity.

Media capture devices include video cameras and microphones, but also screen capturing "devices". For cameras and microphones, we use navigator.mediaDevices.getUserMedia() to capture MediaStreams. For screen recording, we use navigator.mediaDevices.getDisplayMedia() instead.

The peer-to-peer connectivity is handled by the RTCPeerConnection interface. This is the central point for establishing and controlling the connection between two peers in WebRTC.