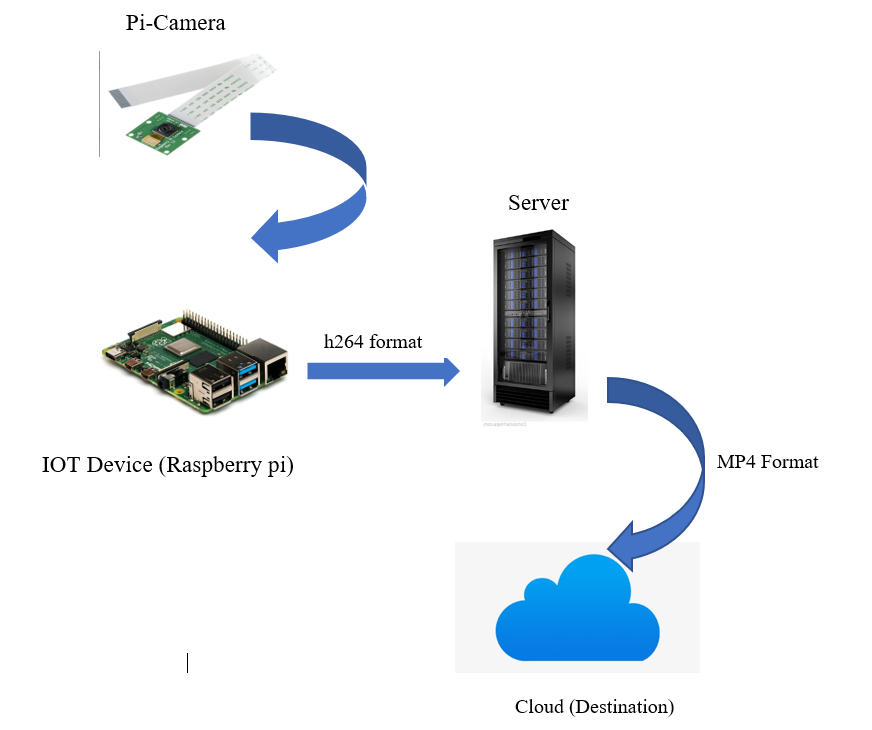
**Abstract**

Security has become an important issue everywhere. Nowadays security is necessary as the possibilities of intrusion are increasing day by day. Presently live video storing and accessing as per the requirements has become a part of the security system. A live video storing security system is a flexible home and public place control and monitoring system remotely using Android-based smartphones application or at the Desktop using the web. This system provides essential security at home or in a remote place. The system captures information and transmits the live video streams to applications using storage mechanisms. This storage application used to have the main role of accessing the videos anywhere, everywhere is achieved. This system operates and controls live video streams and records them for future playback. The main aim is to provide a better security system than the present level of security at home offices or even for a body in motion.

**Introduction**

This paper focuses primarily on reforming the security system by providing live video streaming The Proposed system is going to provide live video hosting services purely on the IoT devices. It shows how live video streaming and predefined video are processed by IoT devices (IP webcam, mobile, tablets, etc.) such that videos from the IoT devices are obtained. The client system, working on the edge of the network takes data from IoT devices. Once the video data is obtained from IoT devices, it is encoded and pushed into the MUX. This helps to achieve much more efficient with respect to performance



**Fig: - Architecture of the proposed system.**

For Uploading a video to the Mux platform, we need a unique independent URL, so for that first uploading this video to firebase storage. From there we get a unique URL. We are Storing that public URL in a JSON file.

The Mux API will return an authenticated URL as well as an ID specific to that Direct Upload so you can check the status later via the API.

Using the request. post method in python, uploading the video to Mux. This action will return a number that indicates a status code. The status code should be 200 if the video is uploaded successfully.

Finally, you get a URL for that video, and using the URL user can access the video remotely.

**Background**

The IP camera has been in existence over the last decade. Video streaming has been used effectively for surveillance purposes. However, in advanced technologies like mobile, IoT, and computing devices, streaming has been an integral part of life. People are using this technology to stream local events to reach more people due to the emergence of IoT and the reduction in the cost of the devices. Hence it has been another popular platform for video streaming. However, due to the limited processing capabilities of the IoT delivering streaming large user data and bigger bandwidth requirement is immensely difficult. To overcome this problem, we can offer a realistic QoS guaranteed video streaming, irrespective of the number of users. In this paper, architecture for video streaming is proposed where data is encoded with MJPEG format at IoT and then it is transferred to a more powerful client system running on the local network. The client system mixes audio and video and encodes this entire pattern into industry-standard H.264 with a very high compression ratio and with minimum loss of quality. The encoded data is mitigated to your storage device which ensures high reliability, efficiency, and QoS to stream clients.

In recent years, there has been an increase in video surveillance systems in public and private environments due to a heightened sense of security. The next generation of surveillance systems will be able to annotate video and locally coordinate the tracking of objects while multiplexing hundreds of video streams in real-time. Video surveillance has been evolving significantly over the years and is becoming a vital tool for many organizations for safety and security applications. Video surveillance systems play an increasingly important role to maintain social security. It has been widely used in many fields, such as finance, public security, banking, and home. Traditional video surveillance can generally achieve close distance monitoring, by using the PC as a monitor host and monitor host-connected monitor camera with coaxial cable.

Initially, it was dominated by analog cameras connected using coax cables. For cost and performance reasons, there was a switch to digital switching systems and now IP-based delivery of data. Detection and tracking of moving objects are important tasks for computer vision, particularly for visual-based surveillance systems. Video surveillance applications, most times imply paying attention to a wide area, so omnidirectional cameras or mobile cameras are generally used. In this system, we use the Raspberry pi 4 chip as the microprocessor. Video data is captured from a USB camera or Raspberry pi camera, compressed into MPEG format, and transferred to the 4G network; then, the monitor client will receive the compressed data frame to restructure and recompose video images. A wireless video monitor systems provide a practical solution for remote wireless monitoring at a low cost and you can even access the video even from your smartphone.

The video is stored on the online open store platform you can also call it cloud. Implementation of the proposed system is divided into 2 phases.

Phase 1 - Recording of Video at the IoT Device and converting the recorded h264 video format to mp4 format.

Phase 2 - Sending the Encoded video stream to MUX Cloud Storage.

Mux Video is an API that enables developers to build unique live and on-demand video experiences. Mux Video combines video encoding, storage (origin serving), and delivery into a single platform. The Mux Video approach is unique. Most of the world follows one of two main alternatives: either build a video streaming platform yourself on top of a single-function API or hand over control of everything to a third-party OVP. Use an OVP if you want to outsource your whole video stack to a third party.

For Uploading a video to the Mux platform, we need a unique independent URL, so for that first uploading this video to firebase storage. From there we get a unique URL. We are Storing that public URL in a JSON file.

The Mux API will return an authenticated URL as well as an ID specific to that Direct Upload so you can check the status later via the API.

Using the request. post method in python, uploading the video to Mux. This action will return a number that indicates a status code. The status code should be 200 if the video is uploaded successfully.

**Technologies Used**

1. **Raspberry pi 4**



**Fig 4.1: -Raspberry Pi 4**

The Raspberry Pi is a series of small single-board computers developed in the United Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer science in schools and in developing countries. It does not include peripherals (such as keyboards, mice, and cases).

The Raspberry Pi is a computer about the size of a credit card, that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing

It’s capable of doing everything you’d expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games. Raspberry pi is used for the hardware modules. Raspberry pi is a credit card size Linux computer used for simple programming. The board is designed with an intention of providing computer education to remote schools where PCs are not very commonly used.

This includes a 700 MHz ARM11 processor; 128 or 256 MB of memory (RAM); there is also a memory card slot, audio/video outputs to connect to the TV, and a USB port for keyboard, mouse, and so forth.

These include the central processing unit (processor), which handles the main workload; the graphics processing unit (GPU), which accelerates the process of producing the complicated graphics you see on your screen; and the random-access memory (RAM) which acts as somewhere for the CPU to keep the information that it is working on.

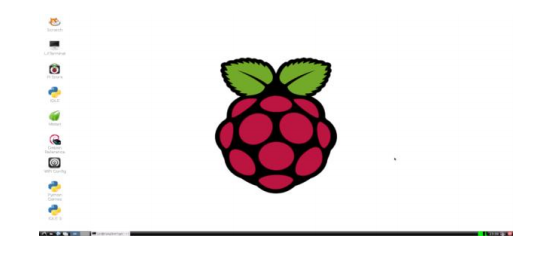
1. **pi-Camera**



**Fig 4.2: -pi Camera Module**

The pi camera is used as a web camera for capturing video. It is normally used in image processing, machine learning or in surveillance projects.

1. **Raspbian Operating System**



**Fig 4.3: - Raspbian Operating System**

It's an Official Operating System of Raspberry Pi. Raspbian is a Debian-based computer operating system for Raspberry Pi. There are several versions of Raspbian including Raspbian Stretch and Raspbian Jessie. Since 2015 it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the family of Raspberry Pi single-board computers. Raspbian was created by Mike Thompson and Peter Green as an independent project. The initial build was completed in June 2012. The operating system is still under active development. Raspbian is highly optimized for the Raspberry Pi line's low-performance ARM CPUs.

1. **Firebase**



**Fig 4.4: - Firebase**

Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android, and Web apps. Firebase provides tools for tracking analytics, reporting and fixing app crashes, and creating marketing and product experiment.

Firebase Cloud Storage is a service that developers can use to store and download files generated directly by clients.

No server-side code is needed. It uses Google Cloud Storage buckets to store the files, allowing accessibility from both Google Cloud and Firebase. These buckets are formed within a hierarchical structure.

It is seamless as it integrates with Firebase Authentication so you can organize uploaded files based on each user and apply access controls if needed.

Also, it scales automatically so there’s no worry about moving to another provider when stored data gets too large.

Accessing also becomes fast and easy as it generates unique URLs for a file which makes it easy to use anywhere, everywhere.

1. **MUX (Online Streaming Storing platform)**



**Fig 4.5: - MUX**

MUX is an open-source platform providing multiple facilities like storing and streaming videos typically behaving as cloud storage.

Mux Video is an API that enables developers to build unique live and on-demand video experiences.

It provides features like streaming a video online, or streaming live video, storing it, direct uploads and pull-based events, and many more. It allows easy interaction with API and stuff like that.

So, basically mux is a platform providing multiple options for video enhancement.

**Description and Working of Project**

The Implementation of the proposed system is divided into 2 phases.

Phase 1 - Recording of Video at the IoT Device and converting the recorded h264 video format to mp4 format.

Phase 2 - Sending the Encoded video stream to MUX Cloud Storage.



**Phase 1:** Set up an IoT device in the first phase, the process of recording video from the pi-camera to IoT devices is carried out.

The recorded video is in h264 format. The h264 format is the encoded format of the video. This format is not readable by Media players. Therefore, we are converting this h264 format to mp4 format using the MP4Box library in python.

INPUT: Video Frames (h264 format)

OUTPUT: Video in mp4 format

**Phase 2:** Sending the Encoded video to MUX.

Mux Video is an API that enables developers to build unique live and on-demand video experiences. Mux Video combines video encoding, storage (origin serving), and delivery into a single platform. The Mux Video approach is unique. Most of the world follows one of two main alternatives: either build a video streaming platform yourself on top of single-function APIs, or hand over control of everything to a third-party OVP. Use an OVP if you want to outsource your whole video stack to a third party.

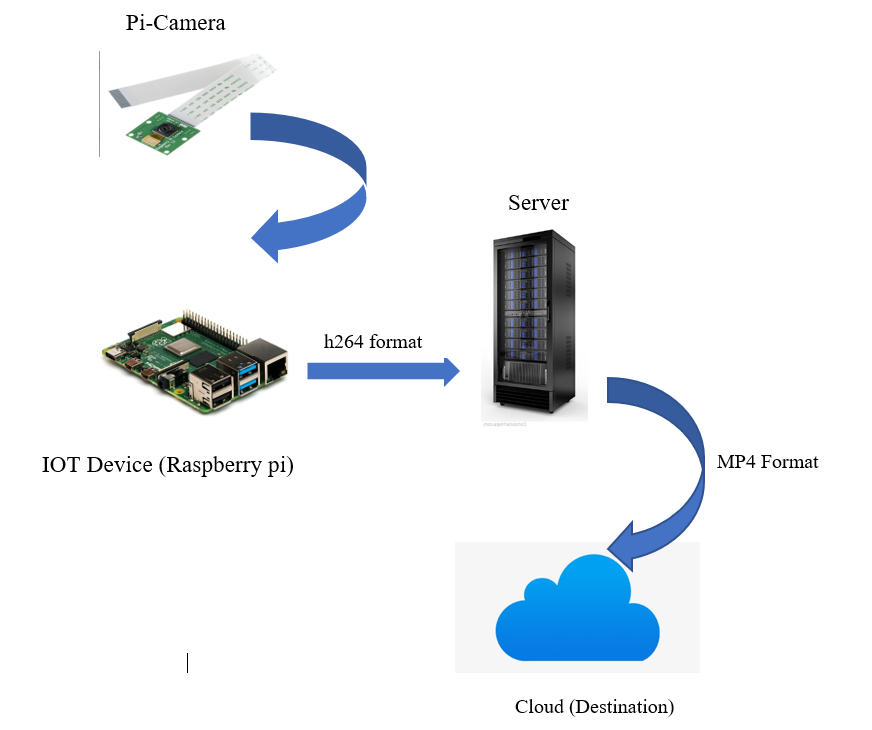
For Uploading a video to the Mux platform, we need a unique independent URL, so for that first uploading this video to firebase storage. From there we get a unique URL We are Storing that public URL in a JSON file.

The Mux API will return an authenticated URL as well as an ID specific to that Direct Upload so you can check the status later via the API.

Using the request.post method in python, uploading the video to Mux. This action will return a number that indicates a status code. The status code should be 200 if the video is uploaded successfully.

Finally, the User gets a URL for that video, using that URL user can access the video remotely.

**System Architecture**



**Fig - Architecture of the proposed system.**

The System working on the edge of the network takes data(video) from IoT devices. Once the video data is obtained from IoT devices, it is encoded and pushed into the MUX. This helps to achieve much more efficient with respect to performance.

After recording video from pi-camera to IoT devices is carried out.

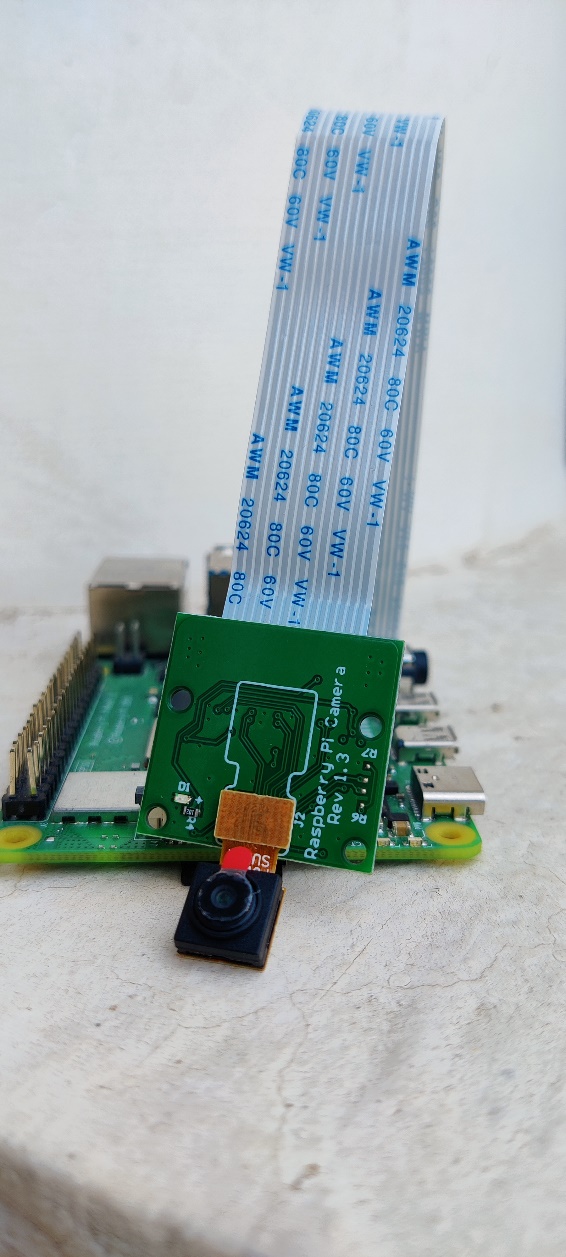
The recorded video is in h264 format then h264 format is the encoded format of video. This format is not readable by Media players so now converting this h264 format to mp4 format using the MP4Box library in python.

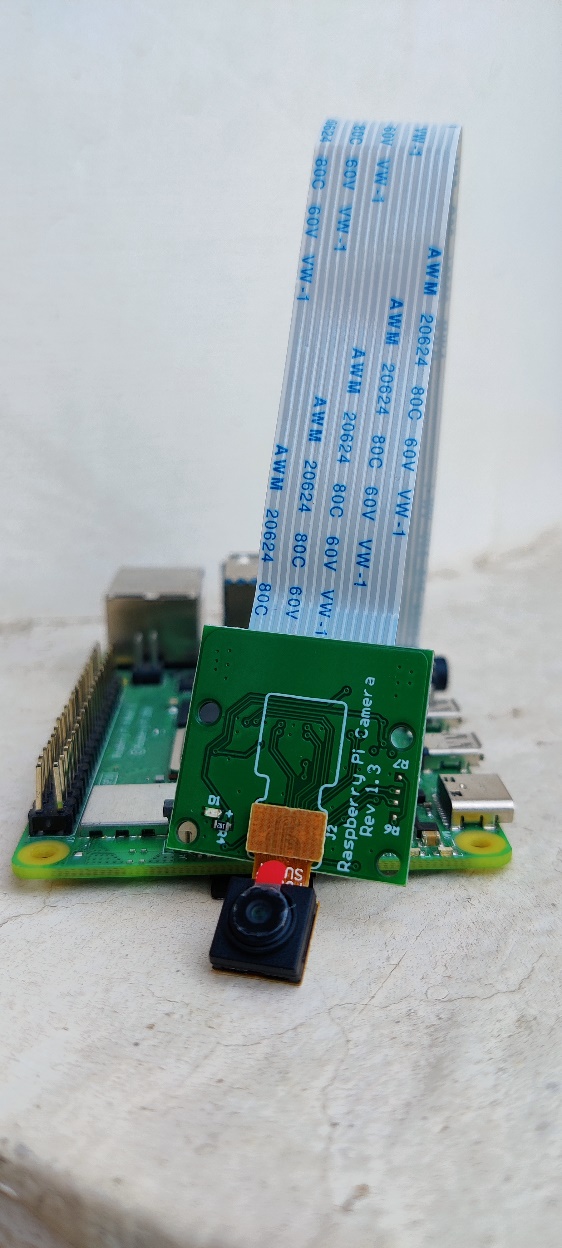
After Formatting the video is uploaded to the Mux platform, we need the unique independent URL, so for that first uploading this video to firebase storage from there we get a unique URL. We are Storing that public URL in a JSON file.

Using the request. post method in python, uploading the video to Mux. This action will return a number that indicates a status code. The status code should be 200 if the video is uploaded successfully and stored in MUX.

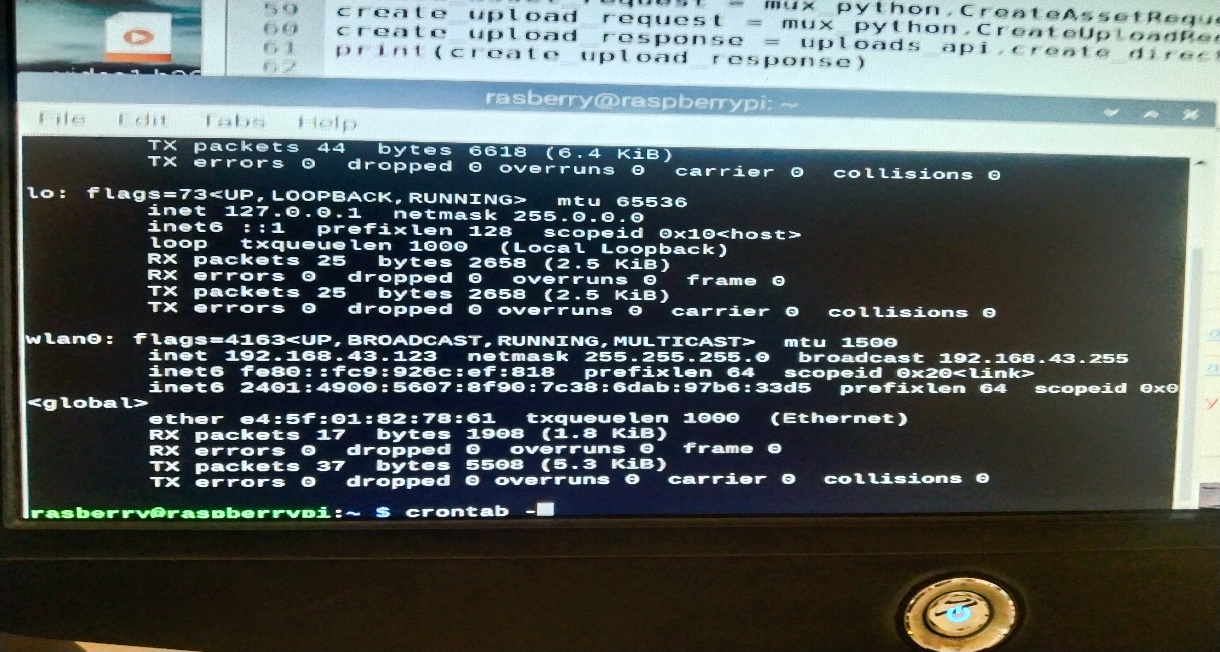
Finally, you get a URL for that video, and using the URL user can access the video remotely anywhere through the URL

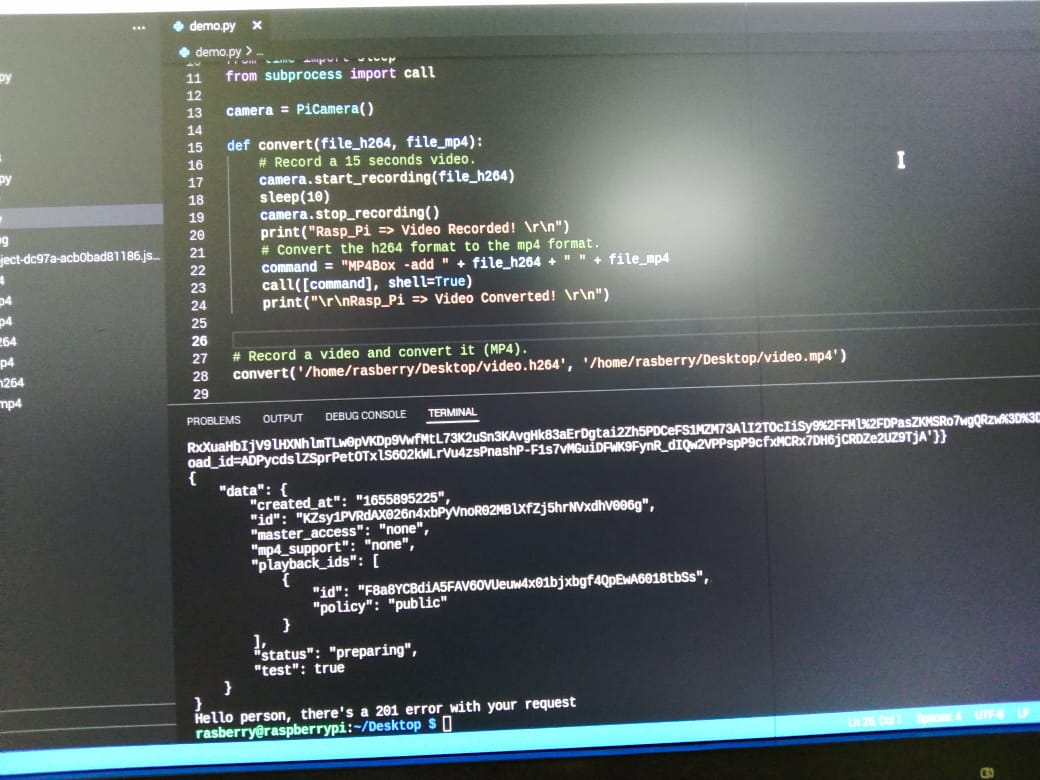
**Screenshots & Results**

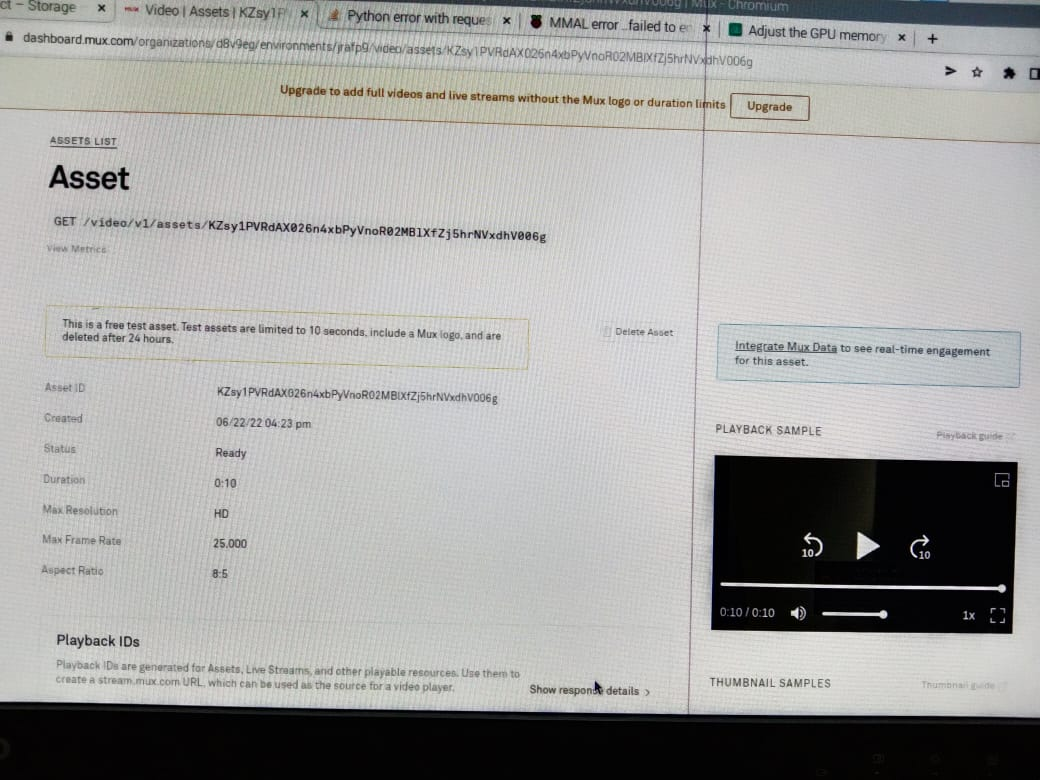


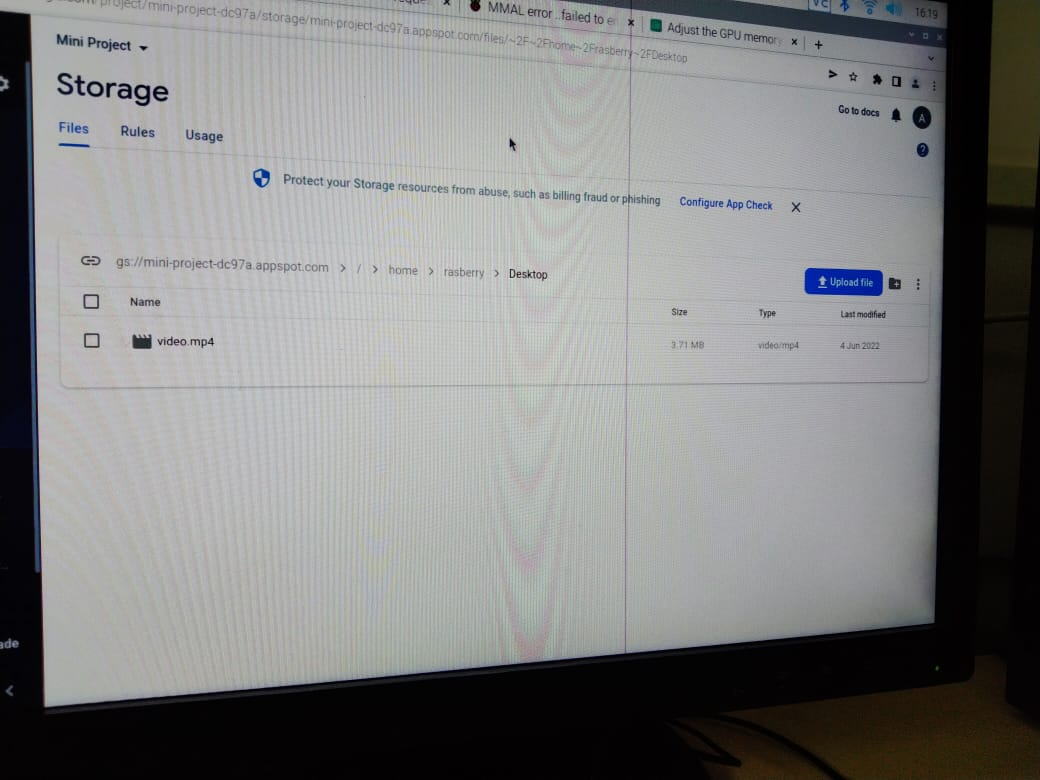














**Advantages and Disadvantages or Applications**

Advantages-

• They could be devices that you own personally and carry with you or keep in your home.

• Security is provided with accessible facilities anywhere and everywhere.

• This constant connection will keep us on the grid constantly, making the real world and

the virtual one inseparable; we will converse both offline and online, making us

accessible no matter the location or the signal status.

• Using firebase storage provides with more authentication.

Dis-Advantages-

• Network Problem.

**Applications**

* ZOOM Cloud Meetings
* Electricity Traffic Lights
* YouTube Live
* Larix Broadcaster
* Airmix
* BroadcastMe
* Twitch
* TikTok

**Future scope**

Live streaming has seen a lot of progressive progress in a relatively short amount of time. Facebook Live to has been growing rapidly and most experts and marketers speculate that this is likely to be a big deal in the coming months and years.

**All cities will be smart** – With more than one-half of the world’s population living in cities innovative new IoT solutions, such as smart parking, connected waste, and traffic management, hold great promise for combatting the major challenges of rapid urbanization. We are unlikely to see many Jetson-like smart cities of the future appearing overnight. However, like in the past with the adoption of revolutionary technologies such as sewers, electricity, traffic lights, and the Internet, mayors will slowly implement IoT solutions to save money, shape the future and make their cities better places to live.

**Overall Project Cost**

|  |  |  |
| --- | --- | --- |
| **Components** | **Quantity** | **Amount** |
| Raspberry pi | 1 | 4000 |
| Pi-Camera | 1 | 390 |
| SD Card | 1 | 500 |
| HDMI Cable | 1 | 420 |
|  | Total | **5310** |

**Conclusion**

In view of the above, it may be concluded that this project will provide security of vehicles from thieves or transport organizations can be able to monitor the location and status of their own vehicles. It will also decrease the crime rates somewhere as anybody who tries to perform such malpractices will think about it before doing it. This device can be fixed with any important or costly items. Since this device uses advanced technology i.e., using IoT application with Raspberry pi 4 board, this makes our proposed project efficient, and definitely, it will be beneficial to our society. It will also decrease the crime rates somewhere as anybody who tries to perform such malpractices will think about it before doing it.

**References**

1. International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 23 Issue 6 –OCTOBER 2016 (SPECIAL ISSUE).
2. Makeuseofit.com/raspberrypi
3. <https://www.youtube.com/watch?v=r5pXWky_3T0>
4. <https://www.dacast.com/blog/best-live-video-streaming-apps-mobile/>
5. <https://www.pngkit.com/view/u2w7r5a9o0q8a9o0_img-cloud-blue-internet-cloud/>
6. https://firebase.google.com/