 **GRT INSTITUTE OF ENGINEERING AND**

**TECHNOLOGY, TIRUTTANI - 631209**

**Approved by AICTE, New Delhi Affiliated to Anna University, Chennai**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**PROJECT TITLE**

***Media Streaming with IBM Cloud Video Streaming***

**College code:1103**

**DHANUSH KUMAR K V**

3rd yr, 5th sem

Reg no.:110321104006

dhanushkumar.kv97865@gmail.com

**Introduction :**

In the era of digital transformation, where information and entertainment are at our fingertips, video streaming has emerged as a powerhouse in the realm of media consumption. From

on-demand movies and TV series to live sports events and user-generated content, video streaming has revolutionized the way we access and engage with visual media. This introduction sets the stage for a deeper exploration of video streaming, shedding light on its significance, evolution, and impact on our daily lives. Video streaming is the process of transmitting video and audio content in real-time or on-demand over the internet. Unlike traditional methods of content delivery, where media files needed to be fully downloaded before playback, streaming allows for the immediate and continuous delivery of data. This transformative technology has not only altered the way we consume media but has also driven substantial changes in the entertainment industry, technology landscape, and societal behavior.

**PROJECT DESIGN :**

Designing a project for media streaming with IBM Cloud Video Streaming involves a series of steps to ensure a reliable, scalable, and feature-rich streaming platform. Below is a project design outline that covers various aspects of setting up a media streaming service using IBM Cloud Video Streaming.

Project Title: Building a Scalable Media Streaming Service with IBM Cloud Video Streaming

**Project Design:**

Project Scope and Objectives:

**i)** Define the scope of the project, including the types of media to be streamed (e.g., live events, on-demand videos).

**ii)** Set clear objectives, such as the number of concurrent viewers, quality of service, and any specific features required (e.g., monetization, analytics).

**Platform Selection:**

i)Choose IBM Cloud Video Streaming as the primary streaming platform for its feature scalability, and reliability.

**Content Preparation:**

**i)** Gather and prepare the media content to be streamed (videos, live broadcasts).

**ii)** Ensure content is appropriately encoded and optimized for streaming.

**IBM Cloud Video Streaming Setup:**

**i)** Create an IBM Cloud account if not already done.

**ii)** Set up an IBM Cloud Video Streaming account.

**iii)** Configure video channels and define your streaming settings.

**Content Ingestion:**

**i)** Integrate video sources (cameras, encoders, prerecorded videos) to feed into IBM Cloud Video Streaming.

**ii)** Implement necessary security measures for content protection.

**Monetization (if required):**

**i)** If monetization is a goal, implement paywalls, subscription models, or ad integration.

**ii)** Configure billing and payment processing.

**Content Delivery:**

i) Leverage IBM's Content Delivery Network (CDN) for efficient content delivery to viewers.

ii) Optimize CDN settings for low latency and high-quality streaming.

User Interface (UI) and User Experience:

i) Develop a user-friendly interface for viewers to access and interact with the streaming content.

ii) Ensure responsive design for various devices and platforms.

Analytics and Monitoring:

i) Implement analytics tools to monitor user engagement, viewership statistics, and system performance.

ii) Set up alerts for any issues or anomalies.

Scalability and Redundancy:

i) Design the architecture to handle scaling gracefully as the number of viewers increases.

ii) Implement redundancy and failover mechanisms for high availability.

Security and Access Control:

i) Implement authentication and authorization mechanisms to restrict access to the content.

ii) Employ encryption for secure data transmission.

Regulatory Compliance:

i) Ensure compliance with copyright laws, content licensing, and other legal requirements.

ii) Address data privacy and GDPR considerations.

Testing and Quality Assurance:

i) Perform extensive testing of the streaming service under various scenarios (peak loads, different devices, slow connections).

ii) Address any performance bottlenecks and fine-tune the setup.

Documentation and Training:

i) Document the entire setup and configuration for future reference.

ii) Provide training to the operational team for maintaining and troubleshooting the platform.

Launch and Marketing:

i)Plan a launch strategy and marketing campaign to promote the streaming service to the target audience.

Maintenance and Support:

i)Establish a support system to handle user inquiries, issues, and technical support. ii)Regularly update and maintain the streaming platform with the latest features and security patches.

Continuous Improvement:

i)Continuously gather user feedback and data analytics to make improvements and optimize the platform.

Cost Management:

i)Monitor and optimize costs associated with cloud resources and streaming services.

By following this project design, you can create a robust media streaming service using IBM Cloud Video Streaming that meets your objectives, offers a seamless user experience, and ensures scalability and reliability for your viewers.

**Features:**

IBM Cloud Video Streaming provides a variety of features for media streaming to help businesses and individuals deliver high-quality video content to their audiences. Below are some of the key features offered by IBM Cloud Video Streaming:

1. Live Streaming: Stream live events, conferences, webinars, and more in real-time to reach a global audience.

2. Video On Demand (VOD): Store and deliver video content for on-demand viewing, allowing users to access videos at their convenience.

3. Customizable Video Player: Customize the video player's appearance to match your branding and user experience.

4. Adaptive Bitrate Streaming (ABR): Automatically adjust the video quality based on the viewer's internet connection, ensuring a smooth playback experience.

5. Monetization: Implement various monetization options, including pay-per-view, subscriptions, and advertising to generate revenue from your content.

6. Security and DRM: Protect your content with features like token-based authentication, IP whitelisting, and Digital Rights Management (DRM) for secure content delivery.

7. Chat and Social Interaction: Engage with your audience through live chat, comments, and social media integration.

8. Analytics and Reporting: Gain insights into viewer behavior with detailed analytics, helping you optimize your content and delivery strategy.

9. Content Management: Organize and manage your media library with features for tagging, categorization, and metadata.

10. Global Content Delivery: Leverage a Content Delivery Network (CDN) to ensure high-quality and fast content delivery to viewers worldwide.

11. Mobile Streaming: Stream video content to mobile devices, ensuring a seamless mobile viewing experience.

12. Multi-Platform Support: IBM Cloud Video Streaming supports various platforms and devices, including web browsers, mobile apps, smart TVs, and more.

13. Multi-Bitrate Streaming: Deliver your content at multiple bitrates to accommodate users with varying internet speeds and devices.

14. High-Quality Video Encoding: Provide high-resolution and high-definition video streams for optimal viewing quality.

15. APIs and SDKs: Access APIs and software development kits (SDKs) for integration with your own applications and websites.

Please note that the availability and specific features may vary depending on your subscription plan and the version of IBM Cloud Video Streaming you are using. It's essential to review IBM's official documentation and offerings to understand the full range of features and capabilities available to you.

**Code**:

import requests

import json

# IBM Cloud Video Streaming API endpoint and credentials

API\_BASE\_URL = "https://api.video.ibm.com"

API\_KEY = "YOUR\_API\_KEY"

API\_SECRET = "YOUR\_API\_SECRET"

# Function to create a new live stream

def create\_live\_stream():

url = f"{API\_BASE\_URL}/channels"

headers = {

"Authorization": f"Bearer {API\_KEY}:{API\_SECRET}",

"Content-Type": "application/json",

}

data = {

"name": "MyLiveStream",

"source\_type": "rtmp",

}

response = requests.post(url, headers=headers, data=json.dumps(data))

if response.status\_code == 201:

stream\_data = response.json()

return stream\_data

else:

print("Failed to create a live stream.")

return None

# Get the details of the live stream

def get\_live\_stream\_details(stream\_id):

url = f"{API\_BASE\_URL}/channels/{stream\_id}"

headers = {

"Authorization": f"Bearer {API\_KEY}:{API\_SECRET}",

}

response = requests.get(url, headers=headers)

if response.status\_code == 200:

stream\_details = response.json()

return stream\_details

else:

print("Failed to get stream details.")

return None

if \_\_name\_\_ == "\_\_main\_\_":

# Create a live stream

live\_stream\_data = create\_live\_stream()

if live\_stream\_data:

stream\_id = live\_stream\_data["id"]

print(f"Live stream created with ID: {stream\_id}")

# Get the stream details

stream\_details = get\_live\_stream\_details(stream\_id)

if stream\_details:

rtmp\_url = stream\_details["source\_output"]["rtmp"][0]["url"]

stream\_key = stream\_details["source\_output"]["rtmp"][0]["stream\_key"]

print(f"RTMP URL: {rtmp\_url}")

print(f"Stream Key: {stream\_key}")

**output:**

Live stream created with ID: YOUR\_STREAM\_ID

RTMP URL: rtmp://YOUR\_RTMP\_URL

Stream Key: YOUR\_STREAM\_KEY