



**ANNA UNIVERSITY REGIONAL  
CAMPUS COIMBATORE**

# **MEDIA STREAMING USING IBM CLOUD STREAMING**

Submitted in partial fulfilment of the degree of

**BACHELOR OF ENGINEERING in  
ELECTRONICS AND COMMUNICATION  
ENGINEERING  
(BATCH 2021-2025)**

**Submitted by:**

<b>Navin V</b>	<b>-au710021106022</b>
<b>Ravi Vishnu A</b>	<b>-au710021106030</b>
<b>Tharunika K</b>	<b>-au710021106037</b>
<b>Jayasurya T</b>	<b>-aut710021106306</b>
<b>Anusri S</b>	<b>-aut710021106303</b>

## **ABSTRACT**

The landscape of media streaming is undergoing a profound transformation, and IBM Cloud Video Streaming emerges as a leading player in this dynamic environment. This project delves deeply into the realm of media streaming, with a specific focus on IBM's cloud service. IBM Cloud Video Streaming provides an extensive suite of features and capabilities, offering organizations the means to seamlessly deliver, manage, and protect high-quality video content to audiences around the globe.

Within this exploration, we thoroughly examine the key components and functionalities of IBM Cloud Video Streaming, encompassing comprehensive content management tools, live streaming capabilities, and the dynamic realm of video-on-demand services. We shed light on the platform's user-friendly interfaces and advanced analytics tools, empowering content providers to gain invaluable insights into viewer engagement patterns and preferences, thus allowing for the refinement of content delivery strategies. Furthermore, we highlight the paramount importance of the robust security measures and content protection mechanisms implemented by IBM Cloud Video Streaming. These measures ensure that sensitive content remains safeguarded in an increasingly interconnected digital landscape, offering peace of mind to content creators and distributors. Through a comprehensive analysis of real-world case studies and practical implementations, this project underscores how IBM Cloud Video Streaming can be a transformative force for organizations seeking to deliver captivating media content efficiently and securely, ultimately enhancing their digital presence and engagement with audiences worldwide.

## TABLE OF CONTENTS

<b>I</b>	<b>INTRODUCTION</b>	<b>4</b>
<b>II</b>	<b>DEVELOPMENT PHASE</b>	<b>5</b>
<b>III</b>	<b>FEATURES OF THE PLATFORM</b>	<b>6</b>
<b>IV</b>	<b>BREAKING DOWN EACH PHASES OF THIS MEDIA STREAMING WITH IBM CLOUD VIDEO STREAMING PROJECT</b>	<b>11</b>
<b>V</b>	<b>PROGRAMMIG CODES AND SYSTEM REQUIREMENTS</b>	<b>14</b>
<b>VI</b>	<b>PLATFORM DESIGN</b>	<b>15</b>
<b>VII</b>	<b>CONCLUSION</b>	<b>17</b>

# **I.INTRODUCTION**

The Problem statement of the project is to create a virtual cinema platform using IBM Cloud Video Streaming. Where users can upload and stream movies and videos on-demand. Define the virtual cinema platform, designing the user interface, integrating IBM Cloud Video Streaming services, enabling on-demand video playback, and ensuring a seamless and immersive cinematic experience.

Media Streaming with IBM Cloud Video Streaming represents a cutting-edge approach to delivering high-quality, on-demand video content to a global audience. With the ambition to create a Netflix-style application, IBM Cloud Video Streaming offers a robust platform that integrates a suite of tools and services tailored to the complex demands of modern media consumption. At its core, this technology leverages the power of IBM's cloud infrastructure to ensure seamless streaming experiences, from anywhere and on any device, be it a smartphone, smart TV, or computer. IBM Cloud Video Streaming provides an array of features that are paramount for building a Netflix-inspired application. These encompass personalized content recommendations, multi-device support, and sophisticated user profiles, ensuring that users can discover, access, and enjoy content tailored to their preferences. The platform's intuitive user interface design and responsive layout guarantee an enjoyable and user-centric experience. Further, content creators benefit from an efficient video upload process, with quality verification checks, metadata enhancements, and thumbnail selections to maximize content discoverability. Streaming integration through IBM's Content Delivery Network and support for live events empowers content diversity and

fosters real-time interactions. A variety of monetization strategies, such as subscription plans and ad integration, provide revenue channels while offering users options for ad-free viewing. Lastly, robust data analytics tools are employed to track user engagement and content performance, thereby enabling data-driven decisions and continuous platform enhancements. In summary, Media Streaming with IBM Cloud Video Streaming stands as a formidable foundation for crafting a Netflix-like application that caters to the evolving expectations of today's media consumers.

## II.DEVELOPMENT PHASE

### 1. **Streaming Platform Development:**

#### **Backend Infrastructure:**

The backend infrastructure includes setting up server clusters, databases, and integrating content delivery networks (CDN). It ensures that the system can handle concurrent user requests efficiently and provides low-latency content delivery.

Usage for Project: A robust backend infrastructure ensures the reliability and performance of the streaming platform, contributing to a seamless and high-quality user experience.

Frontend Development: Explanation: Frontend development focuses on creating an attractive and userfriendly user interface. It integrates the recommendation algorithms into the frontend, allowing users to receive personalized content suggestions. Functions such as search, user profile and integration are also used.

Project Goal: The frontend is a portal through which users can access streaming services; Therefore, a well-designed and effective front-end is crucial for users' engagement and satisfaction.

## **2. Quality Assurance and Testing:**

### **Load Testing:**

Load testing requires a system to simulate high loads to evaluate performance in the busiest ambulance. It identifies conflicts and ensures that the platform processes large streams without interruption.

Mission Goal: Load testing to ensure the platform remains reliable and operates at high levels of performance even during high workloads.

### **User Testing:**

**Description:** User testing involves the interaction of real users with the platform to verify set usability issues, issues, or performance issues. Feedback from users is collected and used to make necessary improvements to the platform.

Purpose: User testing helps identify and resolve issues that may negatively impact user experience to ensure that the streaming platform is user-friendly and free of interference.

## **III.FEATURES OF THE PLATFORM**

### **User profiles:**

Users can create and manage individual profiles within a single account. Each profile independently tracks user preferences, browsing history, and content recommendations. This feature not only improves the personalization of content, but also allows the creation of family or shared accounts.

Use for the project: User profiles are essential for providing highly personalized content recommendations. They also make it easy to organize content for multiple users on a single account.

### **Home page:**

The place to inform the users of the benefits they will find, show them the capabilities, mesmerize them by the looks and promises of entertainment. The Home section highlights movies, TV shows, sports programs, and other categories of content, and is customized according to each user's preferences.

### **Registration/Login:**

An indispensable interactive element for a media streaming service. Provide a variety of ways for the users to register/log-in:

- Facebook login
- Google login
- Mobile number for OTP verification
- Password recovery option

### **User profile:**

It's good to have somewhere to feel at home in. After a home page, a user profile is virtually this dashboard to navigate through the entire journey, not to mention an excellent place for you to get insights (analytics-wise).

### **Settings:**

Letting users control the level of their privacy is a good tactic. Allow users to toggle between privacy options for the audience and social platforms to broadcast to.

### **Search a video:**

Once the platform has grown, it could be increasingly hard to find the content. We need a strict system of categories and an extended system of indicators for users to filter through. Keep filters for searching by location, broadcast topic, date, length of the video, genres, language, and most-watched, etc. Auto-correct and autocomplete will quicken the procedure, especially for a mobile user.

### **Video library:**

Create ‘shelves’ for viewers to treasure their best video experiences – ‘viewed’ history, to watch later, favorites, likes, shares and such. The ‘expiring’ category (Hulu) indicates the episodes that might be prioritized on your watch-list.

### **Video player & play experiences:**

Video players are the first point of contact with the audience, so delivering the best possible experience is key. Viewers can create a familiar look and feel by customizing the following elements: player logo, watermark, a quality selector (crucial for a mobile viewer on a limited data plan), info screen, share & embed options, related videos,



playlist configuration, download button. Mind the integrations: analytics, monetization.

### **Subscription & interaction:**

Users should be able to subscribe to channels of their choice. Allow them to interact with the content and community in a variety of ways:

- Subscribe to channels
- Like/dislike videos
- Comment on videos
- Review and rate
- Share links to free video on websites
- Share the screen
- Live chat

### **Payments:**

On-site payments are indispensable for user retention. Users can pay by PayPal or by their debit/credit card. The two payment models are:

- Pay to get access to gated content (pay-per-view)
- Pay monthly subscription from the page
- This can be technically implemented by integrating PayPal via [Braintree](#) or [Stripe](#).

### **Video upload and ingestion:**

Help transfer video from their multiple devices onto our site with video upload. Video ingestion allows the receiver of the video to take it in and process it for further use. Use the studio in a variety of ways:

- Upload files from the desktop and mobile devices
- Capture videos from the webcam
- Create screen recordings in real-time
- Launch live web events
- Create video presentations with synchronized video and slides
- Import directly from YouTube at the click of a button
- Edit the recording

### **Video management:**

Managing a media library, even a large one, should be simple and straightforward. Content creators will benefit from tools to organize videos and metadata, easily search media, edit the information, moderate, create playlists, etc.

### **Analytics:**

Provide the creators with a comprehensive tracking system to measure the engagement.

## **IV.BREAKING DOWN EACH PHASES OF THIS MEDIA STREAMING WITH IBM CLOUD VIDEO STREAMING PROJECT**

### **PHASE 1:**

**Problem Statement** of the document consists of our understanding about the problem statement. **Proposed solution** focuses on the design aspects of the project which includes features and programming sources that we are going to use for developing the project. **Architectural design** consists of the model of our design thinking that shows how our envisions about the cloud based media streaming web application will work.

### **PHASE 2:**

**Basic tools** used consists of the basic programming tools that we used for the initial level development of our web application. **Web app features** includes the innovative ideas that we have implemented in this project. It highlights the user-friendliness, robust database management, and secure authentication system incorporated into the design. **Block diagram** portrays the way of procedure that we are going to follow during development of this project. **Model prototype** is the prototype of our media streaming web application that consist of the basic innovative features that we have involved.

### **PHASE 3:**

**IBM cloud video streaming** explains the steps that we have done to develop the media streaming platform. **User applications** speaks about the user interface enhancement that have been made which includes proper registration/login that includes Facebook login, google login, mobile number for OTP verification and password recovery option . It also includes payment features for subscription, video management and analytics .

### **PHASE 4:**

**Steps for video storing in cloud** includes the techniques for storing the media in IBM cloud .It involves the requirements for the media storing. It involves the Db2 setup , creating bucket in cloud object storage and the procedure to containerize the application using Docker. In this phase the complete media streaming web application have been developed and its page is uploaded in the ppt.

### **PHASE 5:**

#### **1. Creating a Completely Working Media streaming web application: Concept and Design:**

**Content Selection:** Start by selecting the content you want to feature on your streaming page. Consider obtaining the necessary licenses for content usage or, alternatively, creating your own original content.

**User Interface Design:** Design an intuitive and user-friendly interface. Focus on creating a responsive design that adapts to different screen sizes and devices.

**User Authentication:** Implement a user authentication system, allowing users to create accounts, log in, and personalize their profiles.

**Content Recommendation:** Develop algorithms for content recommendation based on user behavior, such as watch history and ratings.

## 2. Development:

**Frontend Development:** Build the frontend of your media streaming web app using technologies like HTML, CSS, and JavaScript. Consider using modern frontend frameworks like React or Vue.js for a dynamic user interface.

**Backend Development:** Develop a robust backend using a server-side language like Node.js, Python, or Ruby. Implement features like user management, content delivery, and content recommendation.

**Database Management:** Choose a database system (e.g., SQLite, MySQL, MongoDB) to store user profiles, content metadata, and viewing history. - Video Streaming: Implement a video streaming solution, ensuring smooth playback of content. Consider using video encoding and streaming services like FFmpeg and AWS Elemental MediaConvert.

**Payment Processing:** If you plan to monetize your platform, integrate payment processing systems like Stripe for subscription billing.

### **3. Quality Assurance:**

**Testing:** Rigorously test your web app to ensure all features work as expected. Verify the video streaming quality, user authentication, and content recommendation algorithms. - Bug Fixes: Address and resolve any bugs or issues that arise during testing.

**Performance Optimization:** Optimize the platform's performance for fast loading times and smooth streaming

## **V.Programmig Codes and System Requirements:**

### **Programming Languages used:**

- HTML,CSS,Javascript
- Python with Flask

### **Libraries used:**

- An IBM Cloud account.
- Docker and Kubernetes knowledge.
- IBM Cloud CLI and Kubernetes CLI installed.
- A media streaming application or service that you want to containerize.

## Database:

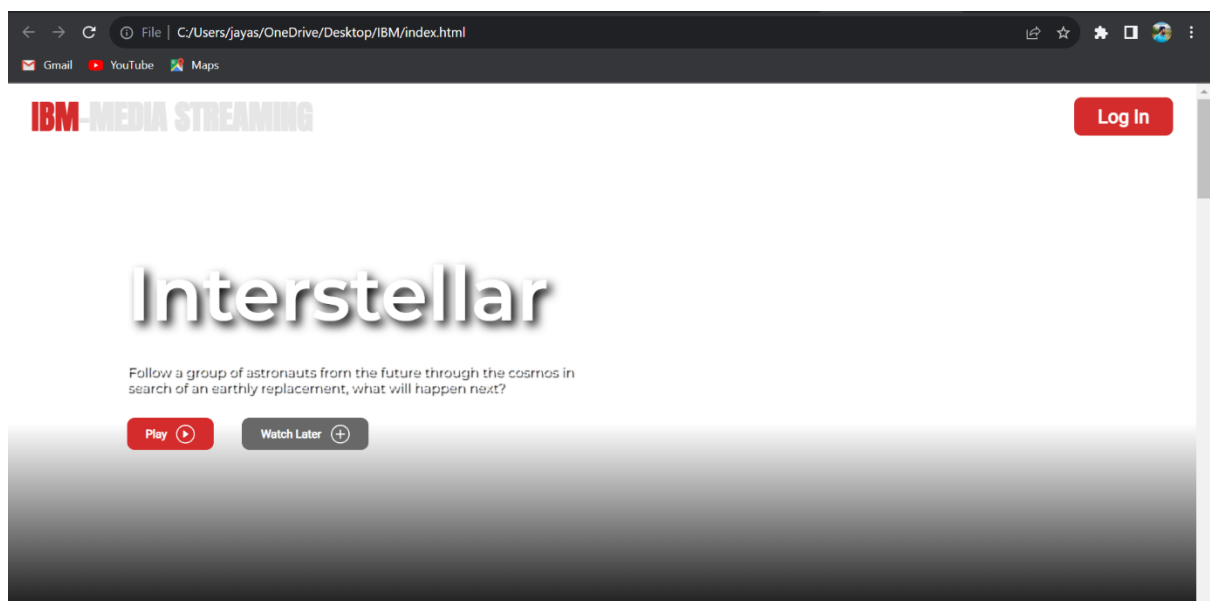
- IBM Cloud DB2
- IBM Cloud kubernities

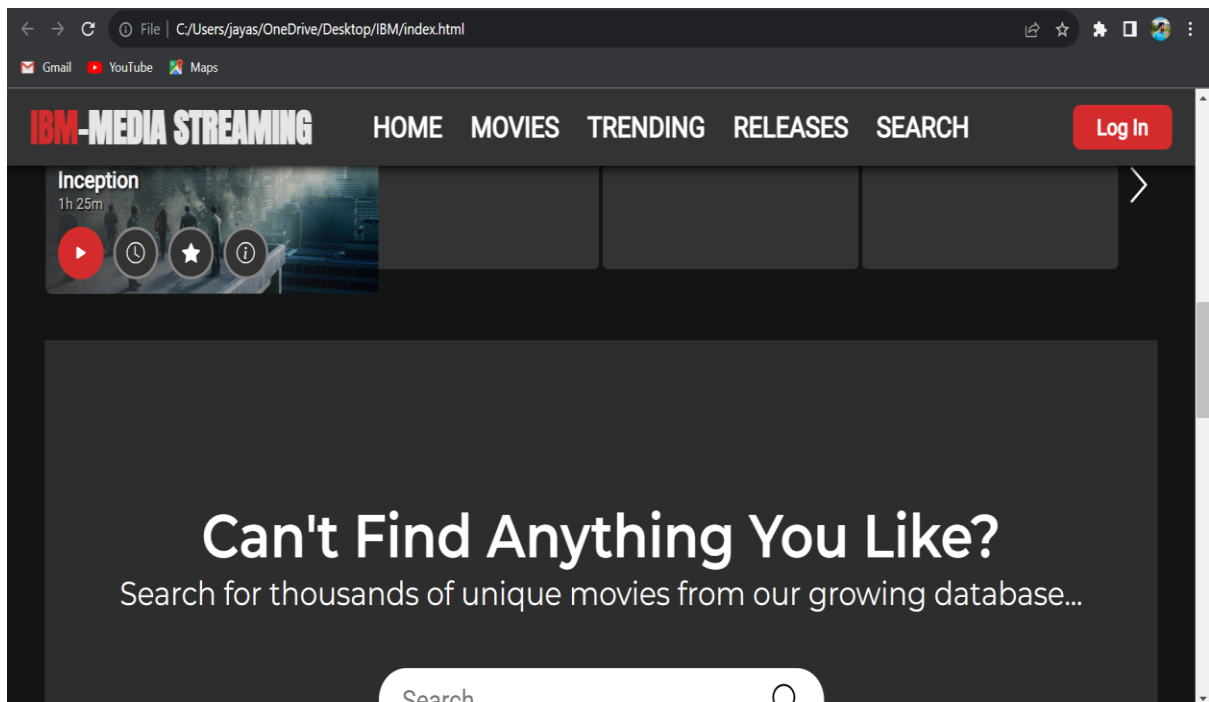
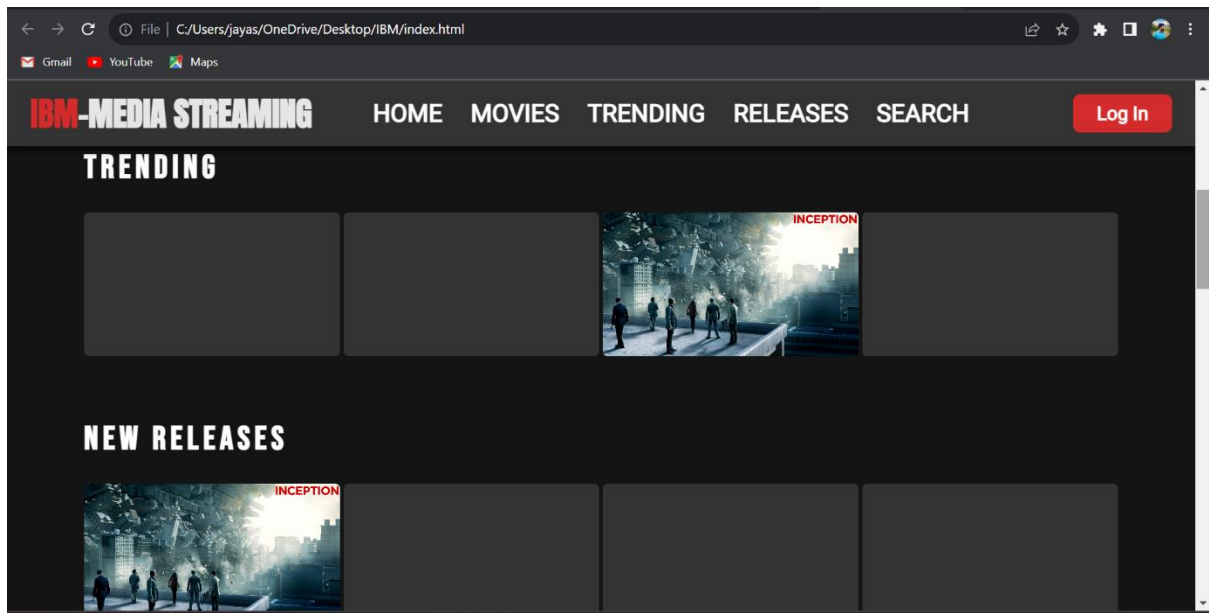
To actually check the Programming codes used to build this “Media Streaming with IBM Cloud Video Streaming” please feel free to check the below mentioned github repository .

- To check the live deployment of the website check the below mentioned link
- To run the backend code you need some system requirements which is explained in the README.md file

Github Repository Link: <https://github.com/T-JAYASUTYA/Cloud-Computing.git>

## VI.PLATFORM DESIGN







## **VI.CONCLUSION**

In our journey to create a streaming platform inspired by Netflix, Amazon Prime Video through the "Media Streaming with IBM Cloud Video Streaming" project, we've gone through several phases. We began by setting our project goals and then moved into planning, design, and development, ensuring we included features like content selection and user-friendly design. Finally, in the last phase, we made our project accessible to the world and publishing it on GitHub. This project represents a blend of creativity, technology, and planning, giving users an engaging media streaming experience similar to other trending media streaming web applications, and marks the successful completion of our project.