

Team Member

960621104021: Gold Lidiya S

Phase 2 Submission Document

Introduction

In today's digital age, the demand for online video content has reached unprecedented heights. Content creators and organizations alike seek a reliable and scalable solution to deliver both live and on-demand video content to their audiences. However, beyond mere content delivery, there is a growing desire to transform the passive act of watching movies into an engaging and interactive experience that fosters community, discussion, and personalization.

This document presents an innovative solution that leverages the capabilities of IBM Cloud Video Streaming to not only address the core requirements of media streaming but also elevate the movie-watching experience to new heights. By incorporating features such as user-generated playlists and real-time chat, our solution aims to empower content creators and organizations to provide an immersive, social, and participatory environment for their viewers.

As we delve into the details of this solution, we will explore the key features and benefits of IBM Cloud Video Streaming and outline an implementation plan that includes these interactive elements. By doing so, we envision a media streaming platform that transcends conventional content delivery, creating a space where viewers become active

participants, forging connections with fellow enthusiasts, and enhancing their overall movie-watching journey.

Design and Innovation Solution for Media Streaming with IBM Cloud Video Streaming

Problem Statement

The growing demand for online video content has created a need for a robust and scalable media streaming solution. Organizations and content creators require a reliable platform to deliver both live and on-demand video content to their audiences while maintaining control, security, and monetization options. Solution Concept

Solution Concept

We propose leveraging IBM Cloud Video Streaming to build a comprehensive media streaming solution. IBM Cloud Video Streaming offers a wide range of features and tools for content upload, live streaming, on-demand video delivery, monetization, audience engagement, and security.

Content for project phase2

Consider incorporating features like user-generated playlists or real-time chat for a more engaging movie-watching experience.

Data Source

A good source for media streaming solution with features like user-generated playlists and real-time chat would need Video Content Library, User Profiles and Authentication, Playlists Database, Real-Time Chat Messages, Analytics and User Behavior Data, Content

Metadata and Thumbnails, Monetization and Transaction Data, Moderation and Compliance Data, Geolocation Data

We propose leveraging IBM Cloud Video Streaming to build a comprehensive media streaming solution. IBM Cloud Video Streaming offers a wide range of features and tools for content upload, live streaming, on-demand video delivery, monetization, audience engagement, and security.

Key Features and Benefits

- **Content Upload:** IBM Cloud Video Streaming allows easy and flexible content upload, supporting various formats and codecs.
- **Live Streaming:** Set up live streaming using secure RTMP and HLS endpoints, making it suitable for events, webinars, and real-time broadcasts.
- **On-Demand Video:** Create on-demand video assets from uploaded content, enabling viewers to access content at their convenience.
- **Monetization:** Implement pay-per-view, subscription, or advertising models to generate revenue from video content.
- **Audience Engagement:** Interact with the audience through live chat and comments, enhancing the viewer experience.
- **Security and Access Control:** Protect content with DRM, geoblocking, and token-based authentication, ensuring content security.
- **Analytics and Reporting:** Gain insights into viewer behavior, stream performance, and content popularity through real-time analytics and reporting tools.

- **8. Integration and Customization:** Utilize APIs and SDKs for seamless integration with custom applications and websites. Customize the player and user interface to match branding.
- **9. Scalability and Reliability:** Benefit from IBM's cloud infrastructure, ensuring scalability and reliability to accommodate large audiences and peak traffic.

Implementation Plan

1. **Account Setup:** Sign up for an IBM Cloud account and access the IBM Cloud Video Streaming service through the IBM Cloud Console.

2. **Content Management:**

- ❖ Upload existing video content to the platform.
- ❖ Organize and categorize content for easy retrieval.

3. **Live Streaming:**

- ❖ Configure streaming software or hardware to connect to IBM Cloud Video Streaming.
- ❖ Test live streaming with a smaller audience before large-scale events.

4. **On-Demand Video:**

- ❖ Create on-demand video assets from uploaded content.
- ❖ Customize metadata, thumbnails, and descriptions.

5. **Monetization:**

- ❖ Implement the desired monetization model (PPV, subscription, or advertising).
- ❖ Configure pricing, access controls, and payment gateways.

6. **Audience Engagement:**

- ❖ Encourage audience interaction through live chat and comments.
- ❖ Monitor and moderate user-generated content.

7. Security and Access Control:

- ❖ Configure security settings, including DRM, geoblocking, and authentication.
- ❖ Define user roles and permissions.

8. Analytics and Reporting:

- ❖ Monitor stream and video performance in real-time.
- ❖ Use data to optimize content delivery and user engagement.

9. Integration and Customization:

- ❖ Explore integration possibilities with other applications or websites.
- ❖ Customize the player and user interface to align with branding.

10. Support and Documentation:

- ❖ Refer to IBM Cloud Video Streaming documentation for guidance and best practices.
- ❖ Access IBM's customer support for assistance as needed.

Architecture and Design

System Architecture

Our media streaming platform is designed to deliver an immersive and interactive movie-watching experience to users. The system architecture has been carefully crafted to support key functionalities, including user-generated playlists and real-time chat.

Components Overview

Frontend: The user-facing interface built using React.js, providing an engaging and responsive experience.

Backend: The server-side logic powered by Node.js and Express.js, responsible for user management, content delivery, and real-time chat functionality.

Database: PostgreSQL serves as our relational database management system, storing user profiles, content metadata, playlists, chat messages, and more.

Real-Time Communication: WebSockets enable real-time chat, ensuring instant interactions among users.

IBM Cloud Video Streaming Integration: IBM Cloud Video Streaming seamlessly integrates with our platform to deliver high-quality video content to users.

Data Flow

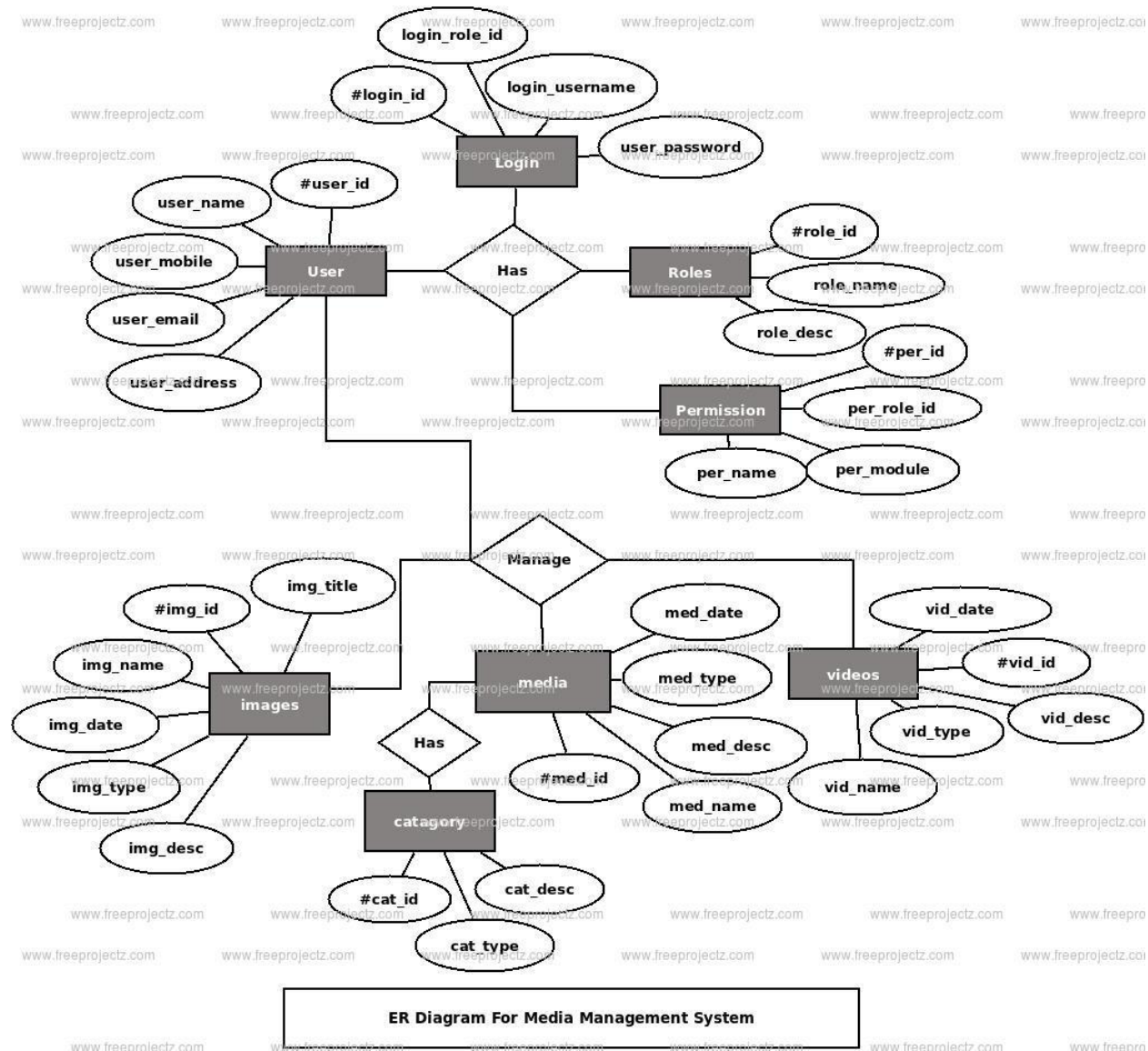
Our system's data flow is as follows:

- User interactions with the frontend trigger requests to the backend.
- The backend communicates with the PostgreSQL database to retrieve user data, content metadata, playlists, and chat messages.
- Real-time chat messages are transmitted via WebSockets for instant communication among users.
- IBM Cloud Video Streaming serves video content to users, with the frontend handling video playback.

Database Design

A robust database design is essential for storing and managing user data, content metadata, playlists, and chat messages efficiently.

Entity-Relationship Diagram (ERD)



Users Table: Stores user profiles, including usernames, email addresses, authentication tokens, and preferences.

Content Table: Contains metadata about video content, including titles, descriptions, genres, durations, and cover images.

Playlists Table: Stores user-generated playlists, including references to content IDs and playlist metadata (names, descriptions, timestamps).

Chat Messages Table: Records chat messages, capturing sender IDs, message content, timestamps, and chat room associations.

API and Interface Design

A well-defined API and user interface (UI) are essential for seamless communication between frontend and backend components.

RESTful API

User Management: Endpoints for user registration, authentication, and profile management.

Content Discovery: API routes for retrieving content metadata, personalized recommendations, and content details.

Playlist Management: API endpoints for creating, editing, and sharing user-generated playlists.

Real-Time Chat: WebSocket endpoints for real-time chat functionality, including joining chat rooms, sending and receiving messages, and chat room management.

User Interface

Homepage: Provides content recommendations and access to user-generated playlists.

Video Player: Offers an intuitive interface for video playback, including controls and video descriptions.

Playlist Management: Allows users to create, edit, and manage their playlists.

Chat Interface: Real-time chat rooms with user avatars, message history, and the ability to send messages.

Security Design

Ensuring the security of user data, authentication, and content protection is a top priority for our platform.

Security Measures

Authentication: User authentication is handled securely using JWT (JSON Web Tokens), ensuring that only authenticated users can access the platform.

Encryption: Sensitive user data and communication are encrypted to protect against unauthorized access.

Content Security: DRM (Digital Rights Management) is employed to protect premium content from unauthorized distribution.

SOURCE CODE

Index.html

```
<!DOCTYPE html>  
<html lang="en">  
  <head>
```

```
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Document</title>
<style>
  body {
    margin: 40px auto;
    max-width: 650px;
    line-height: 1.6;
    font-size: 18px;
    font-family: "Courier New", Courier, monospace;
    color: #444;
    padding: 0 10px;
  }
  h2 {
    line-height: 1.2;
  }
</style>
</head>

<body>
  <h2>HTTP Video Streaming</h2>
  <p>This video is 61MB and is being streamed instead of downloaded.</p>
  <p>
    Feel free to seek through the video and it only loads the part you want to
    watch
  </p>
  <video id="videoPlayer" width="650" controls muted="muted" autoplay>
```

```
<source src="/video" type="video/mp4" />
</video>
<i>Big Buck Bunny</i>
</body>
</html>
```

Index.js

```
const express = require("express");
const app = express();
const fs = require("fs");

app.get("/", function (req, res) {
  res.sendFile(__dirname + "/index.html");
});

app.get("/video", function (req, res) {
  // Ensure there is a range given for the video
  const range = req.headers.range;
  if (!range) {
    res.status(400).send("Requires Range header");
  }

  // get video stats (about 61MB)
  const videoPath = "bigbuck.mp4";
  const videoSize = fs.statSync(videoPath).size;
```

```
// Parse Range
// Example: "bytes=32324-"
const CHUNK_SIZE = 10 ** 6; // 1MB
const start = Number(range.replace(/\D/g, ""));
const end = Math.min(start + CHUNK_SIZE, videoSize - 1);

// Create headers
const contentLength = end - start + 1;
const headers = {
  "Content-Range": `bytes ${start}-${end}/${videoSize}`,
  "Accept-Ranges": "bytes",
  "Content-Length": contentLength,
  "Content-Type": "video/mp4",
};

// HTTP Status 206 for Partial Content
res.writeHead(206, headers);

// create video read stream for this particular chunk
const videoStream = fs.createReadStream(videoPath, { start, end });

// Stream the video chunk to the client
videoStream.pipe(res);
});

app.listen(8000, function () {
  console.log("Listening on port 8000!");
});
```

```
});
```

Package.json

```
{  
  "name": "http-video-stream",  
  "version": "1.0.0",  
  "description": "",  
  "main": "index.js",  
  "scripts": {  
    "start": "nodemon index.js"  
  },  
  "author": "",  
  "license": "ISC",  
  "dependencies": {  
    "express": "^4.17.1",  
    "nodemon": "^2.0.20"  
  }  
}
```

Conclusion

By leveraging IBM Cloud Video Streaming, organizations and content creators can build a powerful and flexible media streaming solution that meets the demands of today's online audience. This solution offers a comprehensive set of features for content management, monetization, audience engagement, and security, all backed by the scalability and reliability of IBM's cloud infrastructure.