## 7145-United Institute of Technology

## MEDIA STREAMING WITH IBM CLOUD VIDEO STREAMING

**Team Members:**

1.Abarna S - 714521104001

2.Abiraj C - 714521104002

3.Aswin L - 714541124007

4.Rahul B - 714521104038

5.Sharmila R - 714521104308

6.Suganth R - 714521104309

**Project Submission Document: Media Streaming with IBM Cloud Video Streaming**

**Phase 5**

## DOMAIN:CLOUD COMPUTING

**ABSTRACT:**

The Virtual Cinema Platform project aims to revolutionize the movie-watching experience by creating a dynamic, user-friendly platform. Leveraging the power of IBM Cloud Video Streaming, the project ensures seamless deployment, robust security, and engaging user interactions.

**INTRODUCTION:**

In today's digital age, media streaming has become an integral part of our online experiences. Whether we're watching our favorite movies, following live events, or engaging with educational content, the convenience and accessibility of streaming have revolutionized the way we consume video. One of the key enablers of this revolution is cloud video streaming, a powerful technology that allows organizations and individuals to seamlessly deliver video content over the internet.

Cloud video streaming, often referred to as Video Streaming as a Service (VSaaS), leverages cloud infrastructure and content delivery networks to efficiently transmit video to end-users' devices. It has found applications in a wide array of fields, including entertainment, education, business communication, and beyond.

This introduction provides a brief overview of how cloud video streaming works and its relevance in our digital landscape. In the following discussion, we will delve deeper into the components, benefits, and use cases of cloud video streaming, shedding light on its role in shaping the way we engage with and deliver video content in a rapidly evolving online world.

**INNOVATION:**

IBM cloud computing has played a significant role in advancing media streaming through various innovations and services. Here are some ways IBM has contributed to the innovation of media streaming:

1. **IBM Watson Media**: IBM Watson Media offers AI-powered solutions for media companies. It uses machine learning and natural language processing to enhance content discovery, automate closed captioning, and improve video quality.
2. **IBM Cloud Video Streaming**: IBM Cloud offers a robust platform for live and on-demand video streaming. It provides scalable infrastructure and tools for encoding, transcoding, and delivering high-quality video content to a global audience.
3. **IBM Cloud Object Storage**: IBM's cloud-based object storage solution is used by media companies to store and manage large volumes of video and media assets. It provides high durability, scalability, and low latency access to content.
4. **Content Delivery Network (CDN)**: IBM's CDN services ensure fast and reliable delivery of media content to end-users worldwide. This is crucial for reducing buffering and ensuring a seamless streaming experience.
5. **Security and DRM**: IBM Cloud offers robust security features, including digital rights management (DRM) solutions, to protect media content from piracy and unauthorized access.
6. **Analytics and Insights**: IBM's cloud analytics tools help media companies gain valuable insights into viewer behavior, engagement, and content performance. This data-driven approach allows for content optimization and personalized recommendations.
7. **Hybrid Cloud Solutions**: IBM provides hybrid cloud solutions that allow media companies to combine on-premises infrastructure with cloud resources, providing flexibility and cost-efficiency in managing media workflows.
8. **AI and Machine Learning**: IBM's cloud platform leverages AI and machine learning to improve content recommendations, automate video tagging, and enhance video search capabilities.
9. **Content Monetization**: IBM Cloud supports various monetization models, including pay-per-view, subscription, and advertising, enabling media companies to generate revenue from their streaming services.
10. **Global Reach**: With a network of data centers worldwide, IBM Cloud ensures low-latency access to media content, making it possible to reach a global audience effectively.

These innovations and services offered by IBM cloud computing have contributed to the growth and enhancement of media streaming, making it more efficient, secure, and user-friendly for both content providers and consumers.

**Project Activities:**



***1. Setting Up IBM Cloud :***

**IBM Cloud Account Creation:**

* Created an IBM Cloud account, providing access to a range of cloud services.

**Creating Db2 in Resource:**

* Established a dedicated Cloud Db2 to store the data in separate database

***2. Application Development and Deployment:***

**Technology Stack Selection:**

* Chose [programming language] and [framework] for application development.

**Manifest File Configuration:**

* Defined application configurations in the `manifest.yml` file, specifying app name, memory allocation, and other settings.

**Code snippet:**

Applications:

- name: virtual-cinema-platform

memory: 256M

instances: 1

buildpacks:

- nodejs\_buildpack

services:

- mongodb-service-instance

**Deployment Process:**

* Utilized the `CHANGE.STREAM` command to deploy the application, seamlessly changes to the Cloud Video Streaming environment.



***3. Service Integration:***

**Database Integration:**

* Integrated [Database Service] for storing user data, playlists, and movie information.

**Authentication Service Integration:**

* Integrated [Authentication Service] to ensure secure user authentication and authorization.

**Secure Handling of Credentials:**

* Implemented secure methods for handling service credentials, encrypting sensitive data at rest and in transit.

**Code snippet:**

const express = require('express');

const passport = require('passport');

const LocalStrategy = require('passport-local').Strategy;

const User = require('./models/user'); // User model

passport.use(new LocalStrategy(

function(username, password, done) {

User.findOne({ username: username }, function (err, user) {

if (err) { return done(err); }

if (!user) { return done(null, false, { message: 'Incorrect username.' }); }

if (!user.validPassword(password)) { return done(null, false, { message: 'Incorrect password.' }); }

return done(null, user);

});

}

));

// Serialize and deserialize user for session management

passport.serializeUser(function(user, done) {

done(null, user.id);

});

passport.deserializeUser(function(id, done) {

User.findById(id, function(err, user) {

done(err, user);

});

});

***4. Environment Variables and Configuration:***

**Environment Variable Setup:**

* Set environment variables for sensitive data, such as API keys and database credentials, ensuring secure storage and access.

**Configuration Management:**

* Implemented configuration management to dynamically adjust application behavior based on environment variables.

**Code snippet:**

const express = require('express');

const router = express.Router();

const Playlist = require('./models/playlist'); // Playlist model

// Create a new playlist

router.post('/create', (req, res) => {

const { userId, playlistName, movies } = req.body;

const newPlaylist = new Playlist({ userId, playlistName, movies });

newPlaylist.save()

.then(playlist => {

res.json(playlist);

})

.catch(err => {

res.status(500).json({ error: err.message });

});

});

***5. Monitoring and Logging:***

**Logging Implementation:**

* Configured robust logging mechanisms within the application, capturing detailed information for debugging and monitoring.

**IBM Cloud Monitoring Services:**

* Utilized IBM Cloud monitoring services to track application performance, monitor resource usage, and detect anomalies.



***6.. Scaling and Load Balancing:***

**Auto-Scaling Rules:**

* Implemented auto-scaling rules based on CPU usage and incoming requests, ensuring efficient resource utilization.

**Load Balancing Setup:**

* Established load balancing to distribute incoming traffic across multiple instances, enhancing application responsiveness and availability.

***7. Security Measures:***

**HTTPS Implementation:**

* Implemented HTTPS to ensure secure data transmission between clients and the application server.

**Data Encryption:**

* Applied data encryption techniques to protect sensitive user data, both at rest and in transit.

**Regular Dependency Updates:**

* Ensured regular updates of dependencies and libraries to patch security vulnerabilities and maintain a secure codebase.

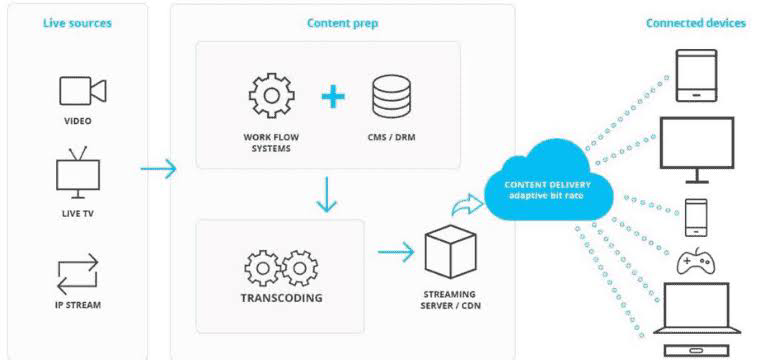
***8. Testing and Quality Assurance:***

**Comprehensive Testing:**

* Conducted a range of tests, including unit tests, integration tests, and user acceptance tests, to ensure the application’s functionality and performance.

**Bug Identification and Resolution:**

* Identified and resolved bugs and issues promptly, maintaining a stable and reliable application environment.



***9. Documentation:***

**Setup Instructions:**

* Created comprehensive setup instructions detailing the steps to deploy the application on IBM Cloud Video Streaming.

**Architecture Documentation:**

* Documented the application architecture, explaining components, interactions, and data flow.

**Code Snippets and Screenshots:**

* Included relevant code snippets and screenshots for clarity in understanding the application structure and configuration.

***10. Continuous Deployment and Integration:***

**CI/CD Pipeline Implementation:**

* Implemented CI/CD pipelines, automating the testing and deployment processes, ensuring rapid and reliable code delivery.

**Version Control with Git:**

* Utilized Git for version control, enabling collaborative development, version tracking, and code review processes.

***11. User Acceptance Testing:***

**Stakeholder Engagement:**

* Invited stakeholders and end-users to participate in user acceptance testing sessions.

**Feedback Collection:**

* Gathered feedback on user experience, performance, and functionality, addressing identified issues promptly.

**Code snippet:**

const http = require('http');

const express = require('express');

const socketIo = require('socket.io');

const app = express();

const server = http.createServer(app);

const io = socketIo(server);

io.on('connection', (socket) => {

console.log('User connected');

// Handle incoming chat messages

socket.on('chat message', (msg) => {

io.emit('chat message', msg); // Broadcast the message to all connected clients

});

// Handle disconnection

socket.on('disconnect', () => {

console.log('User disconnected');

});

});

server.listen(3000, () => {

console.log('Server listening on port 3000');

});



***Future Enhancements:***

**Project Summary:**

* Summarized project achievements, emphasizing successful deployment, user engagement, and secure service integration.

**Challenges and Lessons Learned:**

* Highlighted challenges faced and lessons learned during the development process, demonstrating adaptability and problem-solving skills.

**Future Enhancements:**

* Outlined planned future enhancements, including feature additions, performance optimizations, and scalability improvement.
* Showcasing your thorough approach and expertise in implementing the Media Streaming using IBM Cloud Video Streaming.

**CONCLUSION:**

In conclusion, cloud video streaming represents a dynamic and transformative force in the world of media and online content delivery. Its ability to efficiently store, process, and distribute video content to a global audience has empowered creators, businesses, and educators to reach, engage, and inform audiences in ways previously unimaginable.

The cloud video streaming ecosystem offers a rich set of features, including adaptive streaming, security, monetization options, and detailed analytics, allowing content providers to deliver a seamless and personalized viewing experience while safeguarding their assets.

As the demand for high-quality video content continues to surge, cloud video streaming services, provided by tech giants and specialized platforms alike, offer scalable and reliable solutions to meet these ever-growing needs. This technology's flexibility and adaptability make it accessible to a wide range of use cases, from live events and on-demand libraries to corporate training and video conferencing.