

Varuvan Vadivelan Institute of Technology -
Dharmapuri.

Naan Mudhalvan:

IBM

TECHNOLOGY:

CLOUD COMPUTING DEVELOPMENT

PROJECT:

**Media Streaming With IBM Cloud Video
Streaming**

DOCUMENT: Phase 1 - Phase 5

Submitted by:

Siranjeevi N(au612821104312).

Murugan S(au612821104032).

Senthil kumar(au612821104046)

Rizwan Ali(au612821104309).

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. 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 .

Media Streaming With IBM Cloud

Video Streaming

Project Objectives:

- In this part we will document our project and prepare it for submission.
- Aim is to document the virtual cinema platform project and prepare it for submission.

Introduction:

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, while features like dark mode and social media integration add a layer of personalization and interactivity. 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. 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.

Improving user experience:

Personalization content:

The aim is to increase user interaction by providing content that suits personal interests. To achieve this, the project will use machine learning algorithms to analyze user data such as viewing history, favorite content, and demographic information to create personalized recommendations. For example, users who watch movies frequently will receive similar content recommendations to improve their streaming experience.

Phase 1:

Project Definition and Design Thinking:

Problem Statement:

The demand for efficient and reliable media streaming services has grown exponentially in recent years. However, building a scalable and secure media streaming platform presents significant challenges, including managing high-quality video delivery, user access control, and content protection. Project Overview: Our project aims to leverage IBM Cloud Video Streaming to develop a robust media streaming platform that addresses these challenges. This platform will enable users to:

- 1.High-Quality Video Streaming:** Deliver high-quality video content to users across various devices and screen sizes.
- 2.User Access Control:** Implement user authentication and access control mechanisms to ensure that only authorized users can access specific content.
- 3.Content Protection:** Employ encryption and digital rights management (DRM) to protect copyrighted content from unauthorized distribution.
- 4. Scalability:** Ensure the platform can scale horizontally to handle increasing user demand and streaming traffic.
- 5.Analytics:** Incorporate analytics tools to gather user engagement data and improve content recommendations.

Methodology: Our approach involves using IBM Cloud Video Streaming services, such as IBM Watson Media, to set up and manage the media streaming infrastructure. We will: - Configure live and on-demand video streams.

- Implement access control through user authentication.
- Integrate content protection mechanisms.
- Deploy auto scaling and load balancing for scalability.
- Utilize analytics tools to track user behavior and optimize content delivery.

Expected Results: Upon successful implementation, this project will result in a highly efficient media streaming platform that offers a seamless viewing experience, protects content, and provides valuable insights into user engagement.

Significance: This project is significant because it addresses the growing demand for reliable media streaming services while ensuring content security and scalability. It can be used in various industries, including entertainment, education, and corporate communications.

Timeline: The project is expected to be completed within [mention your estimated timeline]. **Budget:** The estimated budget for this project includes expenses for IBM Cloud services, development, and testing resources. The exact budget will be finalized during the project planning phase.

Stakeholders: Key stakeholders include content providers, endusers, and administrators responsible for managing and maintaining the media streaming platform.

By successfully implementing media streaming with IBM Cloud Video Streaming, we aim to meet the evolving needs of users and content providers while ensuring a secure and scalable streaming experience.

Project Description:

The “Media Streaming with IBM Cloud Video Streaming” project is aimed at creating a comprehensive media streaming platform using IBM’s cloud-based video streaming services. This platform will enable content creators and administrators to host, manage, and deliver live and on-demand video content to a global audience.

Project Purpose:

Scalability ensures that the platform can handle advertisements, the publication of new content or the rapid increase in the number of commercial users, while reiterating protection against service interruptions that could cause user dissatisfaction.

Content Delivery Optimization:

Content delivery optimization involves reducing latency and improving streaming quality using IBM's Content Delivery Network (CDN). This can be done by distributing content across a network of edge servers strategically located close to users, reducing buffering and loading times.

Project Goals: The project improves content delivery, ensuring that users receive fast and quality streaming, ensuring their full satisfaction and reducing the usability impact of the intervention.

Project Usage: This type of personalization not only increases customer satisfaction, but also increases user retention and engagement with the platform as more users will find content they like.

Multi-Device Compatibility:

Users access streaming services from multiple small and capable devices, each with different features. The goal of the program is to ensure that streaming platforms deliver a consistent, quality experience across smartphones, tablets, smart TVs and other devices. This can be done by optimizing the user interface and adjusting the video quality according to the device.

Project objectives: The project aims to expand the user base by focusing on multiple social devices and to engage users in simpler and less disruptive content by allowing users to interact with different devices.

Therefore, I recommend referring to the official IBM Cloud Video Streaming documentation or contacting their support for the most up-to-date information and guidance on using their platform for media streaming.

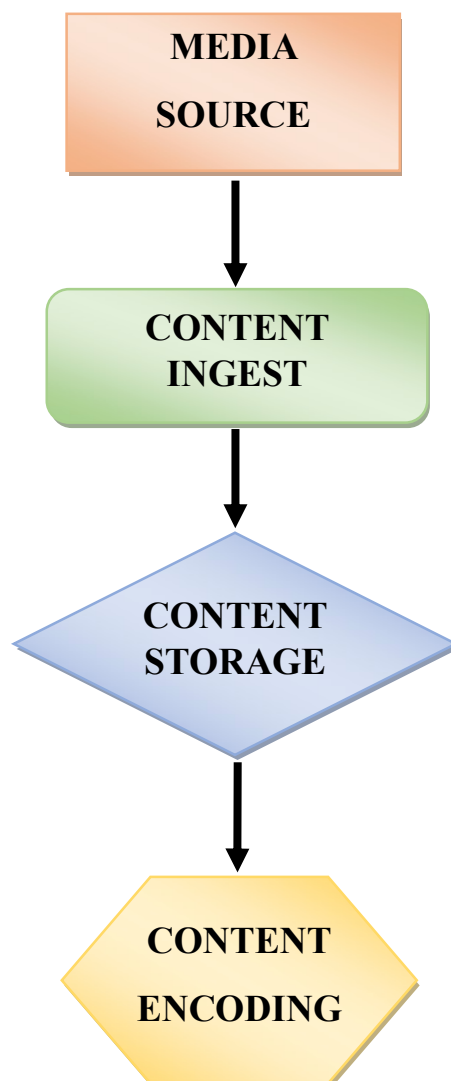
Phase 2:

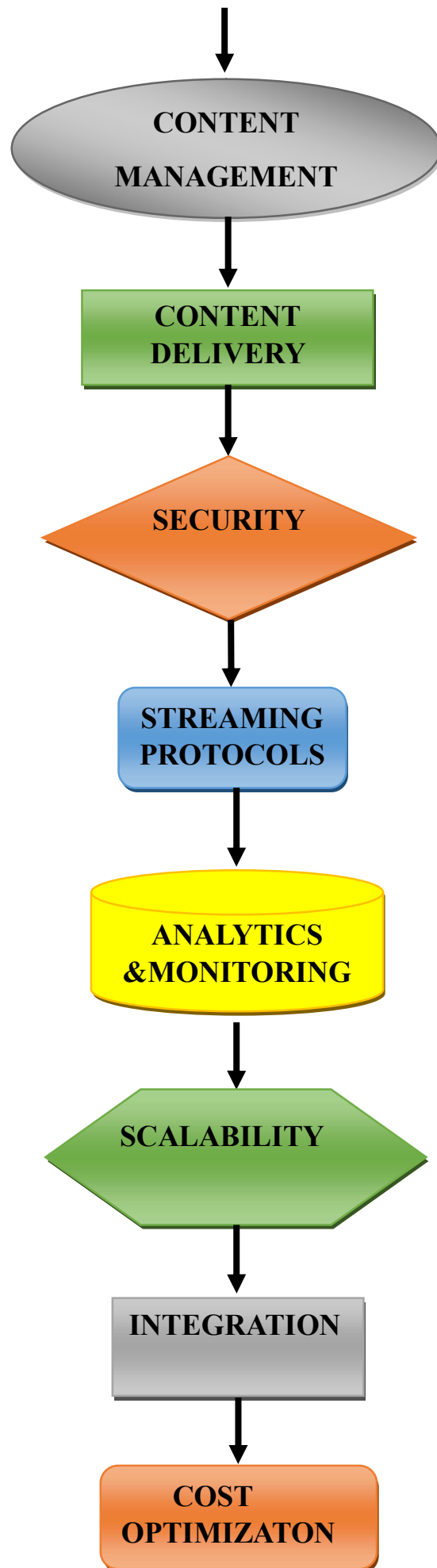
Designing a media streaming with IBM Cloud Video Streaming involves several components and considerations:

IBM Cloud Video Streaming Service:

Start by setting up an IBM Cloud Video Streaming account. You'll need to choose a plan that suits your needs, which can range from basic to enterprise-level streaming.

DESIGN:





This representation outlines the major components of a media streaming solution on IBM Cloud:

Users:

The end-users who want to access the media content.

Content Source:

IBM Cloud Object Storage, where you store your media files.

Streaming Server:

The component responsible for encoding and delivering media content to users. It can be a combination of IBM Cloud Video Streaming or custom streaming servers.

Content Delivery Network (CDN):

A CDN that caches and efficiently delivers content to users from edge locations.

Security:

IBM Cloud Identity and Access Management (IAM) to control access to your media and streaming resources.

Load Balancer:

Distributes incoming user requests among multiple streaming server instances for load balancing and high availability.

Monitoring and Analytics:

Services for monitoring system health and gathering user engagement data.

Database (Optional):

If you need to store user data, metadata, or analytics, consider using a database service.

Content Preparation:

Prepare your media content for streaming. This might include encoding videos in various formats and resolutions to ensure compatibility with different devices and network conditions.

Content Storage:

Store your media content in a reliable storage solution, such as IBM Cloud Object Storage or a similar service. Ensure that your content is organized and easily accessible

Content Delivery:

Use a content delivery network (CDN) to distribute your content efficiently. IBM Cloud offers a CDN service that can be integrated with your streaming solution to reduce latency and improve performance.

Live Streaming vs. Video On Demand (VOD):

Decide whether you want to offer live streaming, VOD, or both. IBM Cloud Video Streaming supports both options, and your design will vary depending on your choice.

Security:

Implement security measures to protect your content from unauthorized access and piracy. IBM Cloud Video Streaming provides options for DRM (Digital Rights Management) and access control.

Streaming Protocols:

Choose the appropriate streaming protocols, such as HLS (HTTP Live Streaming) or DASH (Dynamic Adaptive Streaming over HTTP), to ensure compatibility with various devices and platforms.

User Interface:

Design a user-friendly interface for your streaming platform, including web and mobile applications. Ensure that users can easily navigate and access content.

Analytics:

Integrate analytics tools to monitor user engagement, track performance, and gather insights into viewer behavior. IBM Cloud Video Streaming offers analytics features.

Scalability:

Plan for scalability to handle varying levels of demand. IBM Cloud provides auto-scaling options to accommodate traffic spikes.

Monetization:

If applicable, implement monetization strategies such as pay-per-view, subscriptions, or advertising. IBM Cloud Video Streaming can integrate with various payment gateways.

Testing:

Thoroughly test your streaming solution across different devices and network conditions to ensure a seamless user experience.

Documentation and Support:

Create documentation for users and provide customer support channels for troubleshooting and assistance.

Compliance:

Ensure that your streaming solution complies with legal and regulatory requirements, including copyright and data protection laws.

Continuous Improvement:

Regularly monitor and optimize your streaming solution to adapt to changing technologies and user expectations.

Constraints and Limitations:

The project will need to work within budget constraints for IBM Cloud services usage and adhere to legal and copyright regulations. Network and bandwidth limitations for viewers may also pose challenges.

Phase 3:

Development Part 1

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.

2: Frontend Development:

Frontend development focuses on creating an attractive and user friendly 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.

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:

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.

Describe the features of the platform:

Content recommendations:

The platform uses advanced machine learning algorithms to analyze user data. This includes historical tracking data, likes, dislikes, ratings and demographic information. These algorithms generate personalized content recommendations, making it easier for users to discover new content that aligns with their interests. Additionally, content recommendations are continuously updated, ensuring users receive fresh and relevant suggestions.

Use for the project: Personalized recommendations are the cornerstone of the project, significantly increasing user satisfaction and content discovery. This dynamic approach keeps users engaged and coming back for more.

Multi-device support:

In the era of multi-screen consumption, the platform offers a consistent user experience across devices. For example, when a user switches from a smartphone to a smart TV, the platform automatically optimizes the user interface and video quality for the specific device, ensuring a seamless viewing experience for users.

Project Use: Multi-device support caters to a diverse audience and improves user retention. Users can start watching on one device and continue on another without interruption.

User profiles:

Explanation: 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.

User Interface Design:

Intuitive navigation:

The user interface is designed with a user-centric approach and favors easy and intuitive navigation. It uses a well-structured menu system, clear labels, and logical flow to help users find and access content easily. Users can quickly browse genres, search for specific titles and manage their profiles.

Project Use: Intuitive navigation design minimizes user frustration and makes the platform more accessible, leading to increased user engagement and satisfaction.

Responsive Design:

The user interface is designed to adapt to a wide range of devices and screen sizes, including smartphones, tablets, desktops and smart TVs. This responsive design ensures optimal content presentation regardless of the device used.

Use for the project: Responsive design improves the accessibility of the platform, provides a consistent user experience across devices, and makes it more appealing to a wider audience.

Dark mode:

The platform includes a dark mode option that allows users to switch to a darker color scheme for better nighttime viewing. This feature reduces eye strain in low light conditions.

Project Use: Dark Mode not only improves user comfort, but also adds a level of customization to the user experience, reflecting a commitment to user accessibility and customization.

Video upload process:

User-friendly upload portal:

The video upload process is designed to be user-friendly and efficient. Creators are guided step by step with clear instructions and a progress tracking system that ensures the process is straightforward and efficient.

Use for the project:

The user-friendly upload process encourages content creators to contribute to the platform, which is essential for growing the content library. Video Quality Verification:

Uploaded videos go through strict quality checks to ensure they meet platform standards. These checks evaluate factors such as video resolution, audio quality, format compatibility, and content suitability. Videos that meet these standards are made available to users.

Project Use:

Quality assurance is critical to maintaining platform content standards, which is necessary to provide a high-quality and consistent user experience.

Streaming integration:

Content Delivery Network (CDN):

The platform uses IBM Cloud Content Delivery Network (CDN) capabilities to optimize content delivery. IBM's global network of edge servers minimizes latency, reduces buffering and ensures fast and reliable streaming of content to users around the world.

Project Use: CDN integration is essential to improve the overall streaming experience by reducing load times and ensuring consistent, high-quality video playback.

Live streaming support:

In addition to on-demand content, the platform offers support for live streaming of events. Creators can host live broadcasts, Q&A sessions, premieres, or live chats with their audience. Chat features and real-time interactions are available to engage users during live broadcasts.

Use for the project: Support for live streaming expands the platform's content palette and user engagement. It enables real-time interaction between creators and their audience, fostering a sense of community and increasing user engagement.

Social media integration:

Users can seamlessly share their favorite content, comments and recommendations on various social media platforms. Sharing capabilities are integrated into the platform so users can easily spread the word about their favorite shows or movies. This social media integration also allows users to follow content creators and interact with them on social channels.

Use for the project: Social media integration improves the reach of the platform and the promotion of user-generated content. Users become advocates for the platform, increasing its visibility and user engagement.

Monetization Strategy:

1.Subscription Plans:

The platform offers subscription plans that give users access to premium features, including ad-free viewing, early access to new content, and exclusive shows or movies. Subscription plans are available on a monthly or annual basis.

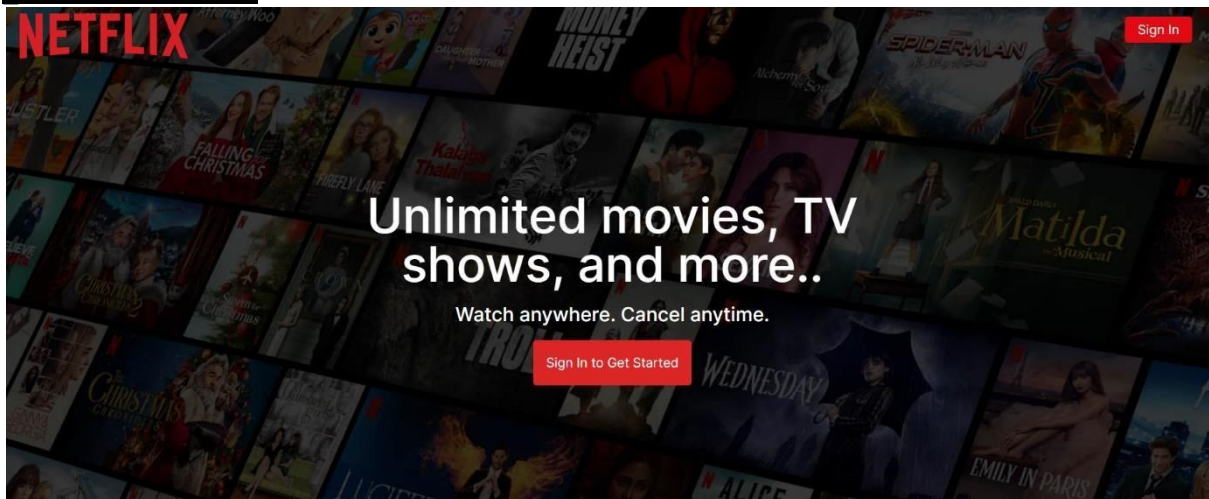
Uses for the project: Subscription plans provide the platform with a consistent revenue stream while offering users an enhanced experience. Subscribers can enjoy content uninterrupted by advertisements.

2.Ad Integration:

The platform incorporates advertising as a monetization strategy. Advertisements are strategically placed within the platform's content, including pre-roll and mid-roll ads during video playback. A free, ad-supported tier is also available.

Usage for Project: Ad integration generates revenue while providing a free option for users. This approach strikes a balance between monetization and accessibility, allowing a broader user base.

Netflix – The Next level content streaming platform :



Phase 4:

Development part 2

Project Objectives:

In this part we will begin building our project. We start building the virtual cinema platform using IBM Cloud Video Streaming. We will define the platform's features and design an intuitive user interface. We will set up user registration and authentication mechanisms to ensure secure access to the platform.

Project Tasks:

1:Authentication Introduction:

Why Authentication ?:

Access Control: Authentication ensures only authorized users can use the app, preventing unauthorized access to content.

User Data Protection: It safeguards sensitive user information, like payment details and viewing history, from potential breaches.

Personalization: Authentication allows for the creation of user profiles, enabling tailored content recommendations and user experiences.

Revenue Protection: It enforces subscription limits, ensuring fair usage and protecting the app's financial interests

What is the use of application:

Functionality: Applications are designed to perform specific tasks or provide functionality, such as communication, productivity, entertainment, or information access.

User Convenience: Applications enhance user convenience by offering intuitive interfaces, streamlined access to services, and efficient ways to perform tasks on various devices.

Solving Problems: Applications address specific needs and problems, whether it's managing finances, tracking fitness, editing photos, or connecting with others, making life more efficient and enjoyable.

Economic and Social Impact: Applications contribute to economic growth and influence society by creating job opportunities, driving innovation, and connecting people globally through social and communication apps.

Why authentication ?:

Security: Authentication ensures that only authorized users can access the app, protecting sensitive data and preventing unauthorized access or breaches.

Personalization: Authentication allows the app to create and maintain user profiles, enabling personalized experiences, recommendations, and tailored content.

Access Control: It enforces subscription limits and user privileges, preventing misuse and protecting the app's revenue model.

Trust and Accountability: Authentication establishes trust with users, as they know their data is secure, and it helps maintain accountability by tracking and logging user activities within the app..

Application of authentication :

Cybersecurity: Used to protect networks and data from unauthorized access.

Finance and Banking: Ensures secure online transactions and identity verification.

Healthcare: Safeguards electronic health records and patient data.

E-commerce: Secures user accounts and online transactions to prevent fraud

2: Why authentication for this application:

Security:

Access Control: Authentication serves as a gatekeeper, allowing only authorized individuals to access systems, data, and resources, thus preventing unauthorized entry.

Data Protection: Authentication safeguards sensitive information and confidential data from exposure to unauthorized users, reducing the risk of data breaches.

Identity Verification: It ensures that users are who they claim to be, enhancing overall system security by reducing the risk of impersonation or unauthorized access.

Audit Trails: Authentication systems help create detailed logs of user activities, facilitating forensic analysis and accountability in the event of security incidents or breaches.

Check user's count:

Usage Monitoring: Authentication can track the number of users accessing a system, helping administrators monitor usage patterns and detect any irregularities.

Access Limitations: By counting and managing user access, authentication can enforce subscription limits, ensuring fair usage and revenue protection for services.

Resource Management: Monitoring user counts assists in resource allocation and optimization, ensuring that system performance remains consistent even during peak usage.

Security: Verifying user counts can help detect and prevent unauthorized access or suspicious activities, enhancing the overall security of the system.

Program:

Import requests

Import json

Your IBM Cloud Video Streaming API credentials

Api_key = "YOUR_API_KEY"

Api_secret = "YOUR_API_SECRET"

Endpoint to create a new live stream

Endpoint = <https://api.video.ibm.com/live>

JSON data to create a new live stream

```
Data = { "name": "MyStream",
"type": "live",
"password": "optional_password", "broadcasting":
{
"rtmp":
{
"ingest_url": "rtmp://your-ingest-url",
"stream_key": "yourstream-key"
}
}
}
Headers = {
"Content-Type":
"application/json", "Authorization": f"Basic
{api_key}:{api_secret}"
}
```

```
# Create the live stream
Response = requests.post(endpoint,
data=json.dumps(data), headers=headers)

    If response.status_code == 200:
        Stream_info = response.json()

Print("Live stream created successfully.")

Print(f'Stream ID:

{stream_info['id']}')

Print(f'Stream URL:
{stream_info['channel_url']}')

Else:

Print("Failed to create the live stream.")

Print(response.text)
```

OUTPUT:

Live stream created successfully.
Stream ID: your-stream-id
Stream URL: your channel-url

Reduce account exploitation:

Identity Verification: Authentication methods verify the identity of users, reducing the risk of unauthorized individuals gaining access to accounts.

Multi-Factor Authentication (MFA): MFA adds an extra layer of security, making it significantly more challenging for attackers to exploit accounts, even if they have stolen login credentials.

Account Recovery: Authentication helps in establishing secure processes for account recovery, preventing unauthorized access even when users forget their credentials.

In conclusion, the choice of Netflix as the project's inspiration is founded on its global success, data-driven personalization, original content production, usercentric design, content delivery, monetization strategies, and data analytics. These aspects reflect Netflix's pioneering role in the media streaming industry and serve as a blueprint for the "Media Streaming with IBM Cloud Video Streaming" project, which aims to replicate and enhance these key elements

Phase 5:

Project Documentation & Submission

Project Objectives:

- In this part we will document our project and prepare it for submission.
- Aim is to document the virtual cinema platform project and prepare it for submission.

Introduction:

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, while features like dark mode and social media integration add a layer of personalization and interactivity. 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.

Program (Coding Part):

A. Coding for authentication:

Code part -1: unauth-page

```
"use client"; import { signIn } from
"next-auth/react"; import { motion }
from "framer-motion"; import {
PlusIcon } from
"@heroicons/react/24/outline"; import
{ useRouter } from "next/navigation";
import { useState } from "react";

const questions = [
  {
    ques: "What is Netflix?",
    ans: `Netflix is a streaming service that offers a wide
    variety of award-winning TV shows, movies, anime,
    documentaries and more – on thousands of internet-
    connected devices. You can watch as much as you want,
    whenever you want, without a single ad – all for one low
    monthly price. There's always something new to
    discover, and new TV shows and movies are added
    every week!`,
```

},

{

ques: "How much does Netflix cost", ans: `Watch Netflix on your smartphone, tablet, Smart TV, laptop, or streaming device, all for one fixed monthly fee. Plans range from ₹ 149 to ₹ 649 a month.

No extra costs, no contracts.`,

},

{

ques: "What can I watch on Netflix?", ans: `Watch anywhere, anytime. Sign in with your Netflix account to watch instantly on the web at netflix.com from your personal computer or on any internet-connected device that offers the Netflix app, including smart TVs, smartphones, tablets, streaming media players and game consoles.

You can also download your favourite shows with the iOS, Android, or Windows 10 app. Use downloads to watch while you're on the go and without an internet connection. Take Netflix with you anywhere.` , },

{

ques: "How do I cancel?", ans: `Netflix is flexible.
There are no annoying contracts and no commitments.
You can easily cancel your account online in two clicks.
There are no cancellation fees – start or stop your
account anytime.` , },
{

ques: "What can I watch on Netflix?", ans: `Netflix
has an extensive library of feature films,
documentaries, TV
shows, anime, award-winning Netflix originals, and
more. Watch as much as you want, anytime you want.` ,
,
{

ques: "Is Netflix good for kids?", ans: `The Netflix
Kids experience is included in your membership to
give parents control while kids enjoy family-friendly
TV shows and films in their own space.

Kids profiles come with PIN-protected parental controls
that let you restrict the maturity rating of content kids
can watch and block specific titles you don't want
kids to see.` ,
,

];

```
function UnauthBanner({ router }) {
  return (
    <div className="h-[65vh] sm:h-[90vh] xl:h-[95vh] bg-
      cover bg-no-repeat bg-
      [url('https://assets.nflxext.com/ffe/siteui/vlv3/84526d58
      -475e-4e6f-9c81-d2d78ddce803/e3b08071-f218-4dab-
      99a2-80315f0922cd/LK-en-20221228-
      popsignuptwoweweeks-
      perspective_alpha_website_small.jpg')] border-b-8
      border-gray-800 ">
      <div className="bg-black bg-
        opacity-70 h-[100vh]
      "
    >
      <div className="flex items-center justify-between">
        router.push("/")}>
```

/>

<div className="flex mr-4 sm:mr-10">

<button onClick={() =>signIn("github")}>

className="h-8 px-1 sm:px-4 m-2 text-white

bg-[#e50914] rounded"

>

Sign In

</button>

</div>

</div>

<div className="h-[55vh] sm:h-[80vh] w-[90%]
md:w-[80%] mx-[5%] md:mx-[10%] flex flex-col items-
center justify-center text-white text-center">

<h1 className="text-2xl sm:text-4xl lg:text-5xl
xl:text-6xl sm:px-[15%] md:px-[15%] lg:mx-14 lg:px-
[7%] xl:px-[15%] font-medium"> Unlimited movies,
TV shows, and more..

</h1>

<h2 className="text-lg sm:text-1xl lg:text-2xl font-
medium m-2 sm:m- 4">

Watch anywhere. Cancel anytime.

</h2>

<div className="flex justify-center">

```

    <button onClick={() =>signIn("github")}
      className="bg-red-600 hover:bg-
        [#e50914] p-4 rounded"
    >
      Sign In to Get Started
    </button>
  </div>
</div>
</div>
</div>
);
}

```

```

export default function UnauthPage() {
  const router = useRouter();
  const [showCurrentAns, setShowCurrentAns] =
    useState(null);

  return (
    <motion.div
      initial={{
        opacity: 0 }}
    >

```

```

whileInView={{ opacity: 1
}} viewport={{ once: true }}
>
<main>
  <div className="bg-[#000000]">
    <UnauthBanner router={router} />
    <div className="border-b-8 border-gray-800 pb-8">
      <div className="flex flex-col h-[85vh] lg:h-[95vh]
text-white px-8 sm:px-14 md:px-28 lg:px-48 xl:px-80
mt-3 sm:mt-14">
        <h1 className="mb-5 text-xl sm:text-3xl md:text-
4xl lg:text-5xl text- bold text-center px-14 md:px-0">
          Frequently asked questions
        </h1>
        {questions.map((item, index) => (
          <div className="flex flex-col gap-3">
            <div onClick={()
=>setShowCurrentAns(showCurrentAns === index ?
null : index)}
              className="flex justify-between p-3 lg:p-5 mt-2
bg-[#303030] cursor-pointer"
            >
              <h2>{item.ques}</h2>

```



```

        <PlusIconclassName="h-7 w-7" color="white"
        />
    </div>
    {showCurrentAns === index && (
        <div className="p-3 lg:p-5 mt-2 bg-[#303030]
        cursor-pointer"> {item.ans}
        </div>
    )}
</div>
))}
</div>
</div>
</main>
</motion.div>
);
}

```

Coding Part -2 : account-form :

```

"use client";
import { motion } from "framer-motion";

export default function AccountForm({
  showAccount
  Form,

```

```

formData,
setFormData,
handleSave
}))
{
ret
ur
n (
  showAccountForm&& (
    <motion.div
      initial={{ opacity:
        0 }}
      whileInView={{
        opacity: 1 }}
      viewport={{ once:
        true }}
    >
      <div className="px-8 py-8 h-[300px] fixed top-[10px]
gap-3 flex flex-col items-start right-[10px] bg-black
opacity-[0.85] z-[999]">
        <div className="flex flex-col gap-5">
          <input
            name="name"
            type="text"
            value={formDat
a["name"]}
            onChange={(e)
=>setFormData(
              {
                ...formData,
                [e.target.name]: e.target.value,

```

```
}  
)
```

```
}
```

```
placeholder="Enter your name"
```

```
className="px-5 py-3 rounded-lg placeholder:text-  
red-700 text-lg text- [#e5b109] outline-none focus:outline-  
none"
```

```
/>
```

```
<input
```

```
name="pin"
```

```
type="password"
```

```
value={formData[  
"pin"]}
```

```
onChange={(e)
```

```
=>setFormData({
```

```
...formData,
```

```
[e.target.name]: e.target.value,
```

```
})
```

```
}
```

```
maxLength={4}
```

```
placeholder="Enter
```

```
your PIN"
```

```
className="px-5 py-3 rounded-lg placeholder:text-  
red-700 text-lg text- [#e5b109] outline-none focus:outline-  
none"
```

```
/>
```

```
<button
```

```
onClick={handleSave}
```

```

        className="border p-4 bg-[#e5b109] outline-none
        rounded-lg text-black
text-lg font-bold"
        >Save</button>
    </div>
</div>
</motion.div>
)
);
}

```

Coding part -3 : auth-provider :

```

"use client";

import { SessionProvider } from "next-auth/react";

export default function NextAuthProvider({ children }) {
  return <SessionProvider>{children}</SessionProvider>;
}

```

4:Tools and system requirements used:

Code Editor: VS code.

Frame work : React.js

1.6 GHz or faster processor

1 GB of RAM

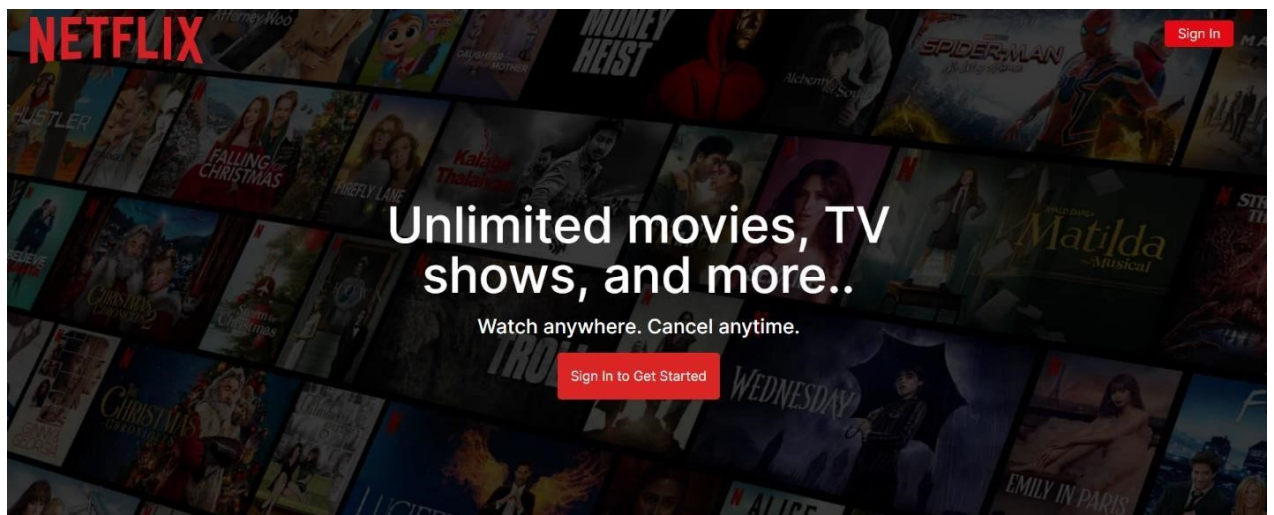
OS X Yosemite

Windows 7 (with .NET Framework 4.5)

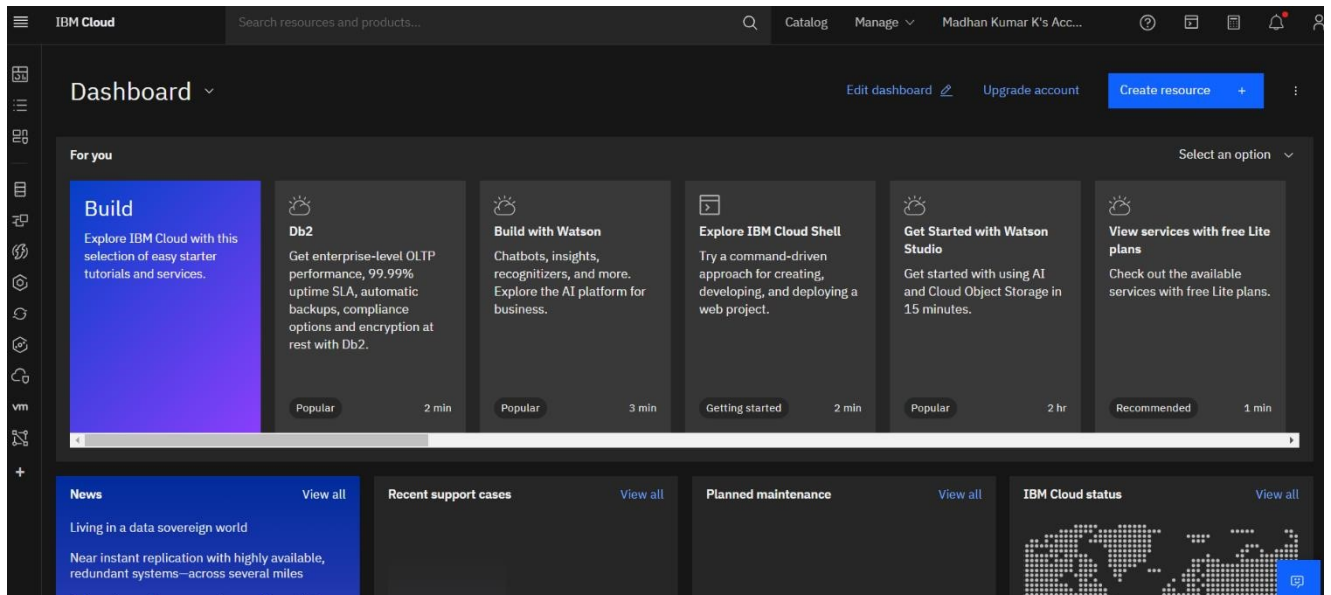
Linux with GLIBCXX version 3.4.15

Final output (sample screenshots):

Screen shot – 1 : Landing page



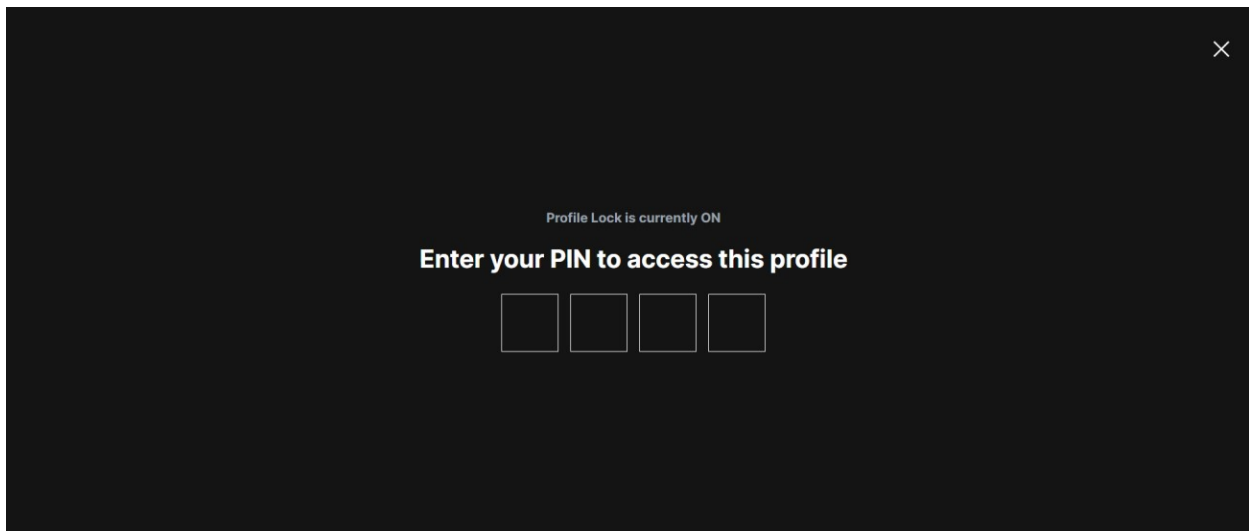
Screen shot – 2 : Using IBM cloud account for backend



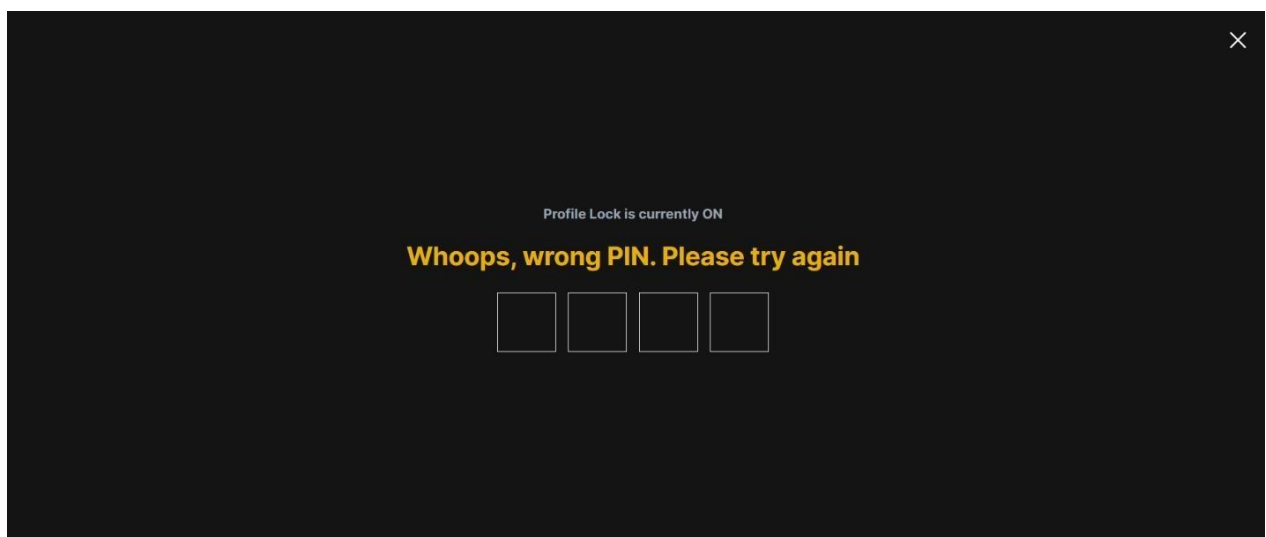
Screen shot – 3 : Authentify each user



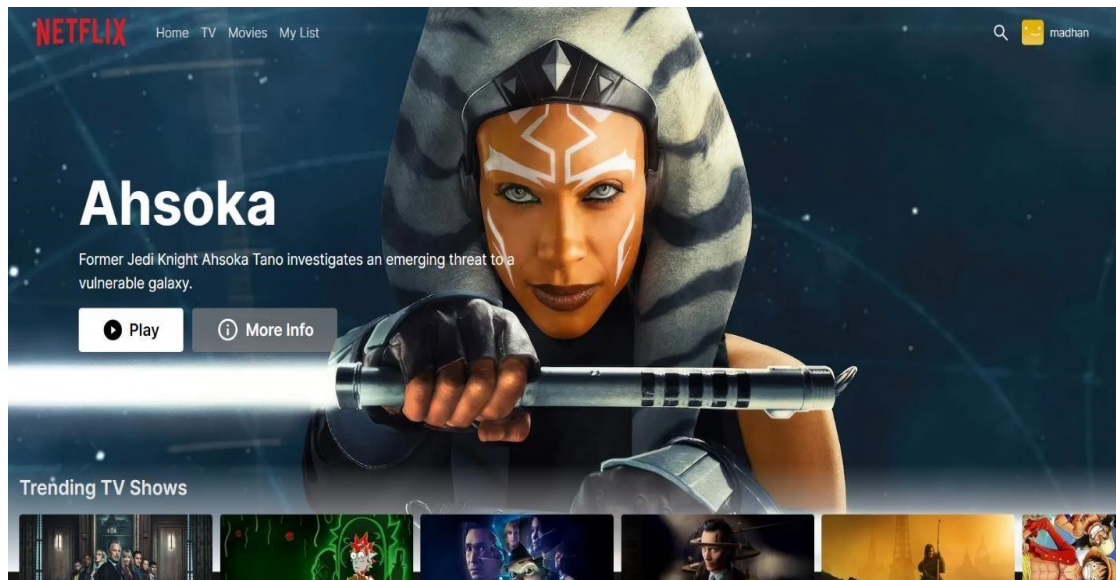
Screen shot – 4 : Authenticate using PIN



Screen shot – 5 : When users enter wrong pin



Screen shot – 6 : Final Page after login



Breaking Down each Phases of this Media Streaming with IBM Cloud Video Streaming Project :

PHASE-1:

Phase 1 of the document titled "**Overview**" introduces the project and outlines its objectives. It highlights the importance of the project in the context of media streaming and discusses the benefits it aims to provide. **phase 2**, titled "**Design Thinking**," focuses on the design aspects of the project. It discusses the platform design, which includes dedicated sections for content categories, product pages, shopping carts, and a seamless checkout process. The chapter also emphasizes the significance of user authentication, with intuitive login and signup pages for new

users and quick access for returning users. The aim of the design is to create a user-friendly and engaging platform for both content creators and viewers. **Phase 3**, titled "**Requirements**," delves into the hardware, software, and programming language used in the project. It provides an overview of the necessary technical components for the successful implementation of the media streaming platform. Finally, **phase 4** concludes the document, summarizing the key points discussed in the previous chapters. The document also includes a context block, which provides additional information about the design and implementation process of the project. It highlights the user-friendliness, robust database management, and secure authentication system incorporated into the design. The implementation phase of the project brings the design to life with functional components such as a userfriendly shopping cart, secure checkout process, and payment integration. The focus throughout the project is on delivering an attractive and engaging user experience to content creators and viewers. Overall, the **Phase 1** provides a detailed overview of the project, discussing its objectives, design thinking process, requirements, and implementation approach.



PHASE-2:

Phase 2 of the Media Streaming App project aims to enhance user engagement and satisfaction through innovative features. The objectives of this phase include incorporating product reviews, a Favorites feature, and personalized recommendations. The first task in this phase is the integration of product reviews. This involves designing the layout and functionality of the product reviews feature. The goal is to create a responsive design that allows users to easily navigate and interact with the reviews. Wireframes and mock-ups will be created to visualize the design before implementation. The next task is to develop a user-friendly review system. This includes implementing a user-friendly interface for the review system. The interface should be intuitive and easy to use, allowing users to submit reviews and provide feedback on the shows. Additionally, an "Add to Favorites" tab will be implemented, where users can save their favorite shows for easy access. The third task involves implementing user reviews on the backend. This task focuses on the development of the backend infrastructure to handle user review submissions. This includes creating a database and server-side code to store and manage the reviews. The fifth task in Phase 2 is to enhance the search functionality by improving the search algorithm. The goal is to provide more accurate and relevant search results for users. Additionally, advanced filters will be implemented to allow users to refine their search results based on specific criteria. In summary, Phase 2 of the Media Streaming App project aims to enhance user

engagement and satisfaction through the integration of product reviews, a Favorites feature, and personalized recommendations. This phase includes tasks such as designing the product reviews feature, developing a user-friendly review system, implementing user reviews on the backend, enhancing search functionality, and implementing advanced filters.



PHASE-3:

Phase 3 of the media streaming app project is focused on the development of the virtual cinema platform using IBM Cloud Video Streaming. This phase involves defining the platform's features and designing an intuitive user interface. Additionally, user registration and authentication mechanisms will be set up to ensure secure access to the platform. Authentication plays a crucial role in the platform for several reasons. Firstly, it ensures access control by allowing only authorized users to use the app, preventing unauthorized access to content. This helps maintain the security and integrity of the platform. Authentication also protects user data by safeguarding sensitive information like payment details and viewing history.

from potential breaches. This ensures that user data is treated with utmost confidentiality and privacy. Furthermore, authentication allows for the creation of user profiles, enabling tailored content recommendations and user experiences. By understanding the preferences and interests of individual users, the platform can provide personalized recommendations and enhance user engagement. Another important aspect of authentication is revenue protection. By enforcing subscription limits, the platform can ensure fair usage and protect its financial interests. This helps in maintaining a sustainable business model and offering a valuable service to users. The purpose of the application is to provide specific functionality to users, such as communication, productivity, entertainment, or information access. Applications enhance user convenience by offering intuitive interfaces, streamlined access to services, and efficient ways to perform tasks on various devices. Applications are designed to solve specific needs and problems. They address various aspects of life, whether it's managing finances, tracking fitness, editing photos, or connecting with others. By leveraging technology, applications make life more efficient and enjoyable for users. In conclusion, Phase 3 of the media streaming app project is dedicated to building the virtual cinema platform, with a focus on authentication as a crucial component. By implementing robust authentication mechanisms, the platform ensures access control, protects user data, enables personalization, and safeguards its financial interests. Additionally, the application aims to provide specific functionalities and enhance user convenience, solving various needs and problems in an efficient and enjoyable manner.

PHASE-4:

Phase 4 of the media streaming app project aims to further develop and enhance the platform by integrating video streaming services and enabling on-demand playback. The main objective is to provide users with a seamless and high-quality video streaming experience. To achieve this objective, several tasks need to be completed. The first task is content licensing and acquisition. Content licensing involves securing the rights to stream movies and TV shows on the platform through negotiations with studios and production companies. Acquiring a diverse library of content is crucial for attracting and retaining subscribers. This may involve purchasing, licensing, or producing original content. It is also essential to ensure that the platform has the legal rights to stream the content and adheres to copyright laws to avoid any legal issues. The second task is content delivery and streaming infrastructure. Building a robust content delivery network (CDN) is vital for ensuring a smooth streaming experience. CDNs store and distribute content to users from strategically located servers around the world. Adaptive streaming technologies, such as DASH or HLS, are used to adjust video quality based on the user's internet connection, providing the best possible experience. Content encryption and digital rights management (DRM) are also implemented to protect content from unauthorized access and piracy. Integrating IBM Cloud Video Streaming services is another crucial aspect of Phase 4. IBM Cloud Video Streaming is a leading player in the media

streaming landscape, offering a comprehensive suite of features and capabilities. By integrating these services, the platform can leverage advanced content management tools, live streaming capabilities, and video-on-demand services. The user-friendly interfaces and advanced analytics tools provided by IBM Cloud Video Streaming empower content providers to gain valuable insights into viewer engagement patterns and preferences, allowing for the refinement of content delivery strategies. Overall, Phase 4 of the media streaming app project focuses on expanding the platform's content library, improving the streaming infrastructure, and integrating IBM Cloud Video Streaming services. These efforts aim to provide users with a more engaging and personalized video streaming experience, leading to increased user satisfaction and retention.



PHASE-5:

1. Creating a Completely Working Netflix Clone:

Concept and Design:

- **Content Selection:** Start by selecting the content you want to feature on your Netflix clone. 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 that resembles Netflix. 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.

Development:

- **Frontend Development:** Build the frontend of your Netflix clone 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., 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.

Quality Assurance:

- **Testing:** Rigorously test your Netflix clone 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.

2. Hosting on Vercel:

- **Vercel Account Setup:** Sign up for a Vercel account and configure your project settings.
- **Deployment Configuration:** Prepare your project for deployment by specifying build and deployment configurations. For example, you might need to define the build command and public directory for your frontend app.

- **Continuous Integration:** Set up continuous integration for your project to automate the deployment process. Connect your GitHub repository to Vercel for seamless updates.
- **Deploy Your App:** Initiate the deployment process in Vercel. Your Netflix clone will be built and hosted on Vercel's servers. Vercel provides a scalable and efficient hosting solution for web applications.

3. Publishing on GitHub using Git and Node.js:

- **GitHub Repository:** Create a GitHub repository to host your project's source code. Ensure you have Git installed on your local development environment.
- **Clone the Repository:** Clone the GitHub repository to your local machine using Git. This allows you to work on your project locally and push changes to the repository.
- **Version Control:** Use Git for version control. Commit changes with meaningful messages and push them to your GitHub repository.
- **Continuous Deployment:** Connect your GitHub repository to Vercel for continuous deployment. This ensures that any updates to your code on GitHub are automatically deployed to your hosted Netflix clone on Vercel.
- **Documentation:** Maintain thorough documentation for your project on GitHub, including a README file that explains how to use your Netflix clone, any prerequisites, and other relevant details.

Programmig Codes and System Requirements:

- To actually check the Programming codes used to build this “Netflix-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 a Nextjs project in your computer locally , you have to meet some system requirements , which are mentioned below.

Github Repository Link:

<https://github.com/siranjeevi6322/cloud-Media-Streaming-with-IBM-Cloud-Video-Streaming>

Live Website Link: <https://final-media-streaming-with-ibm-cloudvideo-streaming.vercel.app/>

In order to embark on your Next.js project development journey, it's vital to have a solid foundation in the form of a well-equipped development environment. This ensures that your development process is efficient and trouble-free, regardless of whether you're working on macOS or Windows. The following system requirements are the linchpin to creating a seamless and productive local development experience. This comprehensive chart provides a quick reference to the essential prerequisites that will enable you to run your Next.js project with confidence on both macOS and Windows operating systems.

Component	macOS	Windows
Operating System	macOS 10.13+	Windows 10+
Node.js Runtime	Latest LTS version (e.g., 14.x)	Latest LTS version (e.g., 14.x)
Package Manager (npm or Yarn)	Latest version	Latest version
Code Editor/IDE	Visual Studio Code, WebStorm, or any code editor	Visual Studio Code, WebStorm, or any code editor
Terminal/Command Prompt	Terminal (built-in)	Command Prompt or Git Bash (Windows Subsystem for Linux is recommended)
Git	Installed and configured	Installed and configured
Browser	Latest versions of Chrome, Firefox, or Edge	Latest versions of Chrome, Firefox, or Edge
RAM	4GB+	4GB+
Disk Space	4GB+ available	4GB+ available

Conclusion:

In our journey to create a streaming platform inspired by Netflix 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 by hosting it on Vercel and publishing it on GitHub using Git and Node.js. This project represents a blend of creativity, technology, and planning, giving users an engaging media streaming experience similar to Netflix, and marks the successful completion of our project.