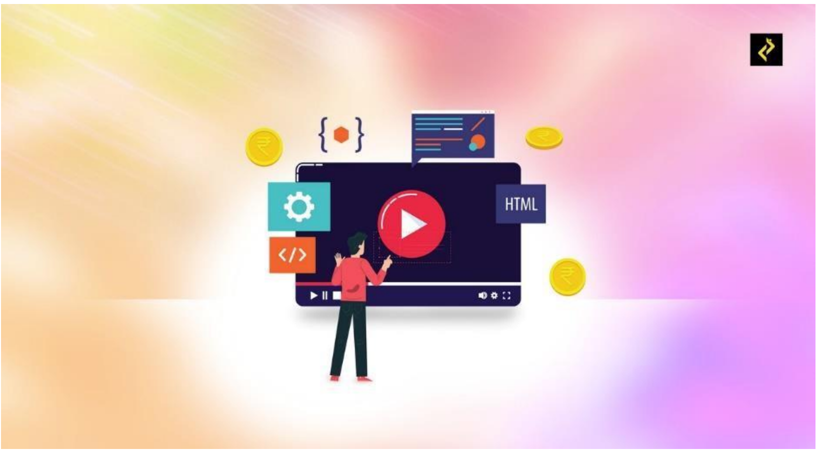
Media Streaming with IBM Cloud Video Streaming



Submitted by

H. Aaliya Samira

P. Anitha

S. Deepa

S. Devisri

R. Dharanisri

PHASE 3 SUBMISSION DOCUMENT

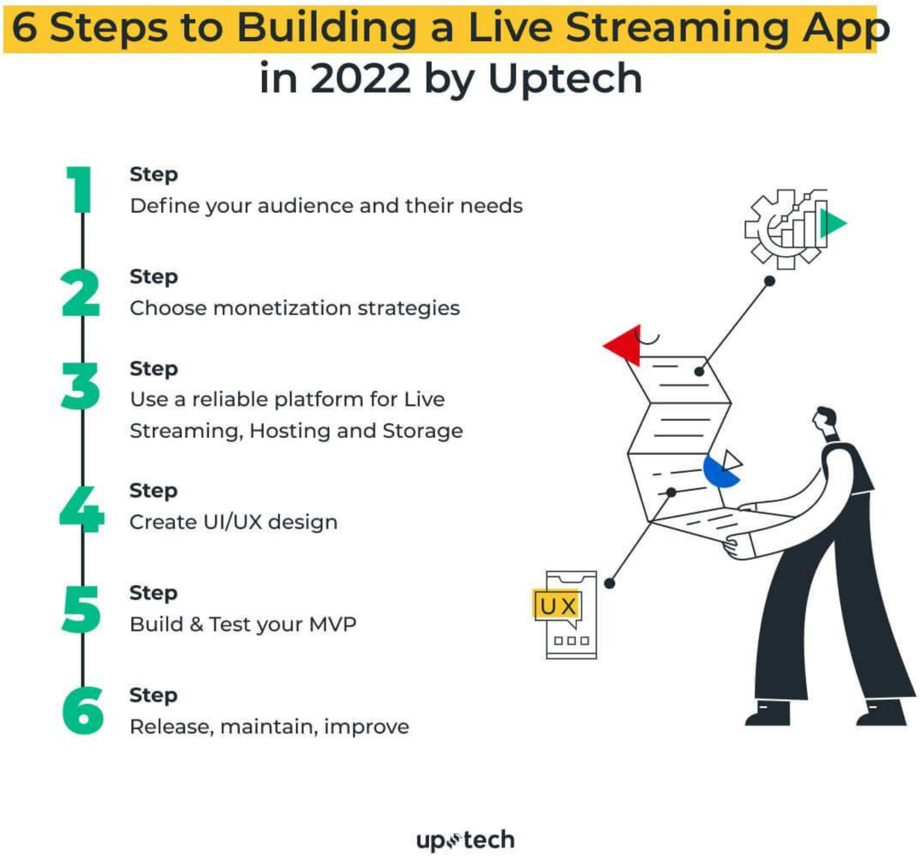
Phase 3: Development part1

Topic: Create a virtual cinema platform using IBM Cloud Video Streaming. Upload and stream your favourite movies and videos on demand.

Problem Statement:

- Create a virtual cinema platform using IBM Cloud Video Streaming. Upload and stream your favourite movies and videos on-demand.
- Share joy of movie nights with friends and family, no matter where they are located.
- Elevate movie-watching experience with streaming and high-quality video playback for a truly immersive cinematic experience.

Steps to create media streaming /video streaming applications



6 Steps to Building a Live Streaming App in 2022 by Uptech

1 Step
Define your audience and their needs

2 Step
Choose monetization strategies

3 Step
Use a reliable platform for Live Streaming, Hosting and Storage

4 Step
Create UI/UX design

5 Step
Build & Test your MVP

6 Step
Release, maintain, improve

uptech

Design Media streaming platform layout and create user authentication information coding

Planning and Design:

- The virtual cinema platform, including user registration, video upload, and on-demand streaming.
- Users to upload movies and videos to the platform.
- Focus on providing a seamless and immersive movie-watching experience with high-quality video playback.
- IBM Cloud Video Streaming services to enable smooth video playback and streaming.

1. **Cloud Hosting:**

   - Choose a cloud provider like AWS, Azure, or GCP.

2. **Media Server:**

   - Set up a media server for streaming content.

3. **Content Delivery Network (CDN):**

   - Utilize a CDN for efficient content distribution.

4. **Authentication Service:**

   - Implement user authentication with services like OAuth or custom solutions.

5. **Content Storage:**

   - Store video content in a cloud-based storage solution.

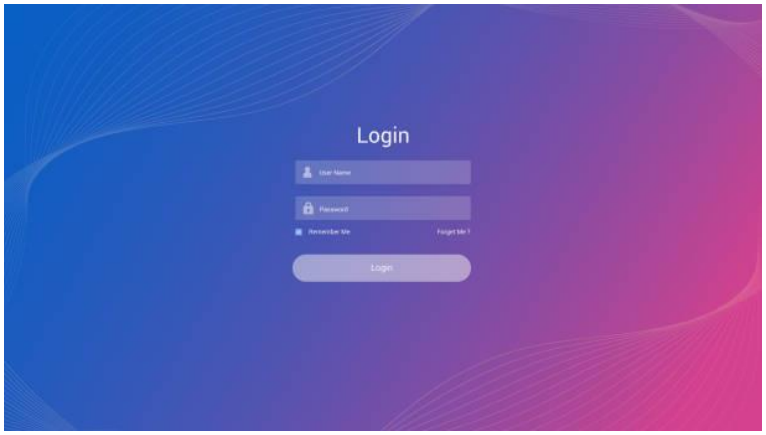
6. **Database:**
   - Maintain a database for user profiles and access control.
7. **Encryption:**
   - Ensure data encryption for secure streaming.
8. **Scalability:**
   - Design for scalability to handle varying loads.
9. **Content Transcoding**:
   - Support multiple formats and resolutions for compatibility.

Set Up Your Development Environment:

1. **Cloud Environment**: Sign up for a cloud platform (e.g., AWS, Azure, or GCP) and set up your development environment with the necessary resources and access permissions.
2. **Media Server Configuration**: Install and configure a media server, such as Wowzah or Amazon Elemental, for video streaming, and integrate it with your cloud environment.
3. **Development Tools**: Install and configure development tools(e.g., SDKs, IDEs) for your chosen cloud platform and programming language to building the authentication and streaming components of your application**.**

Secure Your Application:

- Implement the security process for using HTTPS, securing APIs, and protecting user data.



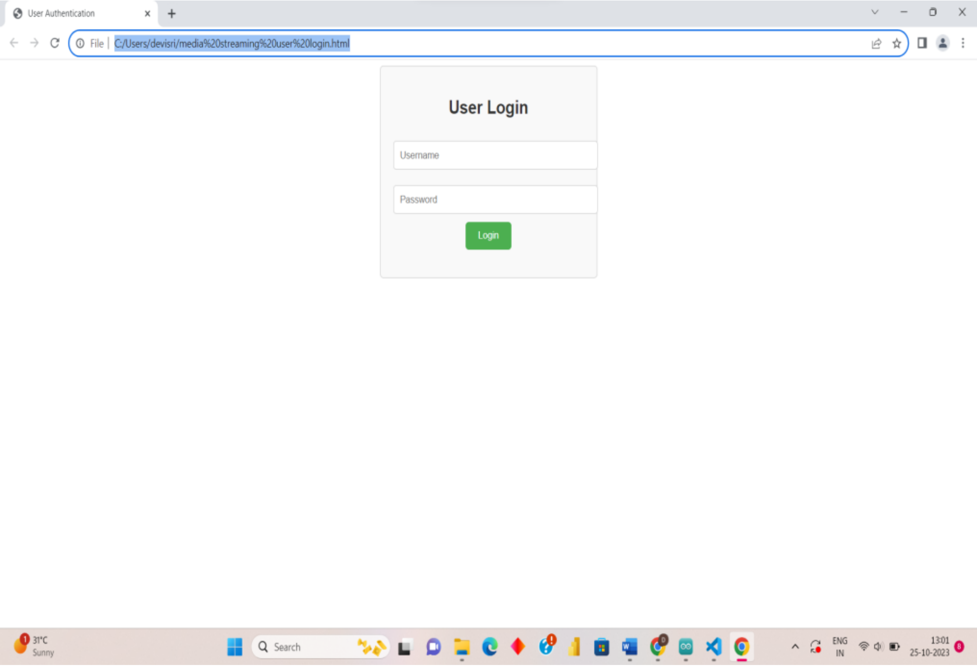
Program for user authentication:

```
<!DOCTYPE html>

<html>
<head>
  <title>User Authentication</title>
  <link rel="stylesheet" type="text/css" href="styles.css" />
  <style>
    body {
      font-family: Arial, sans-serif;
    }

    container {
      text-align: center;
      max-width: 300px;
      margin: 0 auto;
      padding: 20px;
      border: 1px solid #ccc;
      border-radius: 5px;
      background-color: #f9f9f9;
    }

    h2 {
      color: #333;
    }

    input[type="text"],
    input[type="password"] {
      width: 100%;
      padding: 10px;
      margin: 10px 0;
      border: 1px solid #ccc;
      border-radius: 5px;
    }

    button {
      background-color: #4caf50;
      color: #fff;
      padding: 10px 20px;
      border: none;
      border-radius: 5px;
      cursor: pointer;
    }

    error-message {
      color: red;
    }
  </style>
</head>

<body>
  <div class="container">
    <h2>User Login</h2>
    <form id="login-form">
      <input type="text" id="username" placeholder="Username" required />
      <input type="password" id="password" placeholder="Password" required />
      <button type="submit">Login</button>
    </form>
    <p id="error-message" class="error-message"></p>
  </div>
  <script>
```

```
document.getElementById("login-form").addEventListener("submit", function (e) {
    e.preventDefault();

    // Get input values
    const username = document.getElementById("username").value;
    const password = document.getElementById("password").value;

    // Simulate authentication (replace with a real authentication check)
    if (username === "user" && password === "password") {
        // Successful login
        document.getElementById("error-message").textContent = "Login successful!";
    } else {
        // Failed login
        document.getElementById("error-message").textContent = "Incorrect username or password.";
    }
});

    </script>
  </body>
</html>
```

Output:

USER AUTHENTICATION



Conclusion:

cloud-based media streaming for virtual cinema platforms offers a seamless and scalable solution for on-demand video content. The user authentication process ensures secure access to content, enhancing the overall experience for viewers. This technology has revolutionized the way we consume media, making it more accessible and user-friendly than ever before.