

# **Media Streaming with IBM Cloud Video Streaming**

**TEAM MEMBER**

**312621243023:Shalini.J**

**Phase-5 Document submission**

**Project Title: Cloud Media Streaming with IBM Cloud Video Streaming**

## **Problem Statement**

**Objective:** Develop a seamless and reliable cloud-based media streaming platform using IBM Cloud Video Streaming.

## **Problem identified:**

In today's digital age, the demand for high-quality, uninterrupted media streaming experiences is higher than ever. Whether it's for entertainment, education, or business purposes, users expect smooth and reliable playback across various devices and platforms. However, setting up a robust and scalable cloud media streaming solution can be a complex task, involving considerations for video encoding, content delivery, and user experience optimization.

## **Introduction:**

In response to this growing need, our project, "Cloud Media Streaming with IBM Cloud Video Streaming," aims to create a cutting-edge platform for delivering high-quality video content over the internet. Leveraging the capabilities of IBM Cloud Video Streaming, we will design a solution that ensures seamless playback, regardless of the viewer's location or device.

The project starts by addressing the core issue: providing a reliable and scalable infrastructure for media streaming. This involves tasks such as video transcoding, adaptive bitrate streaming, content delivery network (CDN) integration, and user interface design. We will utilize the powerful features of IBM Cloud Video Streaming to handle these tasks efficiently.

Our project focuses on a well-structured process that spans from content upload and encoding to user interface customization and integration with third-party services. We will also implement features like analytics and user authentication to enhance the overall streaming experience.

In the following sections, we will delve into the intricacies of our project, describing the methods, tools, and technologies we utilize to build a robust cloud media streaming platform. Our goal is to address the increasing demand for high-quality video content delivery, empowering content creators and businesses to reach their audience with a seamless streaming experience.

**Data:** The primary data for this project will consist of various video files in different formats and resolutions. Additionally, we will collect user engagement metrics to analyze viewer behavior and optimize the streaming experience. This data will be instrumental in fine-tuning our platform for optimal performance.

## **LITERATURE SURVEY**

### **1." Cost Minimization of Cloud Services for On-Demand Video Streaming", Mahmoud Darwich [2022]**

This research explores the utilization of cloud technology for video stream processing, leveraging the benefits of virtual machines and cost-effective storage servers. To cater to diverse user devices, multiple video formats are typically prepared, posing a challenge in terms of storage costs. This study presents an approach to optimize cloud storage by strategically determining

which videos, in what formats, should be stored, thereby minimizing overall cloud service expenses. Promising results were obtained, demonstrating effectiveness with a growing number of frequently accessed videos and increasing view counts. The proposed method resulted in a noteworthy reduction of up to 22% in cloud service costs.

## **2.” Cost-Efficient Storage for On-Demand Video**

### **Streaming on Cloud ”, Ishihara [2021]**

This paper addresses the challenge of video transcoding, where videos are converted into various formats to accommodate different user devices. This process demands significant storage and resources. With the rise of cloud technology, video streaming companies have shifted to processing videos in the cloud. Traditionally, multiple pre-transcoded video formats are stored and streamed, leading to high storage costs. This paper proposes a solution for cost-effective video storage in the hierarchical cloud storage. The method optimizes video pre-transcoding decisions based on suitable cloud storage to minimize costs. Experimental results demonstrate the effectiveness of the approach, particularly in scenarios with a high percentage of frequently accessed videos, leading to potential cost reduction of up to 40%.

## **3.” PERFORMANCE ANALYSIS OF VIDEO ON-DEMAND AND LIVE VIDEO**

### **STREAMING USING CLOUD BASED SERVICES - 2021”, UJASH PATEL [2022]**

The paper explores Cyber-Physical Systems (CPS) and their integration with video streaming for applications like smart grids and health monitoring. It emphasizes the importance of Quality of Experience (QoE), cost, and bandwidth impact on cloud-based video analysis for Video-On-Demand Streaming (VoDS) and Live Video Streaming (LVS). Content Delivery Networks (CDNs) are discussed as crucial for achieving optimal user experience across various cloud providers

#### **4.” Improving Hierarchy Storage for Video Streaming in Cloud ”, Yasser Ismail [2021]**

To address the cost implications of storing various video stream formats, this research focuses on utilizing cloud technology for efficient storage. Traditionally, storing multiple formats incurred high expenses. By adopting cloud services, video stream companies aim to mitigate costs. However, storing all streams in a single cloud type escalates costs, worsened by changing access patterns. To optimize storage costs, the paper introduces a method that leverages hierarchical cloud storage. The algorithm identifies frequently accessed video segments and stores them in the appropriate cloud storage type. Experiments demonstrate a promising 18.75% reduction in cloud storage costs through this approach.

#### **5.” Point Cloud Video Streaming: Challenges and Solutions ”, Weishan Zhang [2021]**

This article addresses the emerging field of volumetric video, essential for VR/AR/MR experiences and well-suited for advanced wireless communication like 5G. It emphasizes the need for efficient volumetric video streaming and focuses on point cloud video as a popular way to represent volumetric media. The article introduces point cloud video technology and its applications, outlines challenges and solutions in point cloud video streaming, and discusses encoding, tiling, viewing angle prediction, decoding, quality assessment, and transmission optimization. A preliminary MPEG DASH-based point cloud video streaming prototype is explained with simulation results, and future research directions are highlighted for high-quality point cloud video streaming.

## **DESIGN THINKING**

Design Thinking Approach for Media Streaming with IBM Cloud Video Streaming

**Empathize:**

Gain insights into users' needs and expectations, focusing on viewers and content creators.

**Actions:**

- Conduct user surveys or interviews to understand preferences and pain points.
- Analyze market trends and competitors to identify key success factors.
- Seek input and feedback from experts in content delivery, user experience design, and cloud infrastructure.

**Define:**

Set clear objectives and success criteria.

**Objectives:**

- Develop a robust, scalable cloud-based platform using IBM Cloud Video Streaming.
- Ensure high-quality video playback and provide a user-friendly interface.
- Achieve high user satisfaction metrics, including increased user engagement, minimal buffering, and positive feedback.

**Ideate:**

Brainstorm innovative approaches leveraging IBM Cloud Video Streaming.

**Actions:**

- Explore IBM Cloud Video Streaming capabilities and integration with HTML, CSS, PHP, and JS for hosting and streaming content.
- Consider various streaming protocols and codecs to deliver high-quality video content through the platform.
- Brainstorm features like content recommendation algorithms, user-generated content integration, and social sharing functionalities within the framework of IBM Cloud Video Streaming.

**Prototype:**

Visualize the platform using HTML, CSS, PHP, JS with IBM Cloud Video Streaming.

**Actions:**

- Set up an IBM Cloud environment to test content upload, storage, and streaming capabilities.
- Develop a user interface prototype using HTML, CSS, PHP, and JS, integrating with IBM Cloud Video Streaming.
- Conduct usability testing with a small group of users to validate the initial design concepts and user experience.

**Test:**

Evaluate technical and user experience performance with IBM Cloud Video Streaming.

**Actions:**

- Test the IBM Cloud Video Streaming infrastructure for scalability, security, and performance under various loads.
- Gather user feedback on the prototype's usability, navigation, and overall satisfaction, specifically in conjunction with IBM Cloud Video Streaming.
- Use metrics such as video buffering rates, load times, and user engagement to assess the prototype's effectiveness with IBM Cloud Video Streaming.

**Implement:**

Build and deploy the platform using HTML, CSS, PHP, and JS in conjunction with IBM Cloud Video Streaming.

**Actions:**

- Integrate content upload, storage, and streaming components with IBM Cloud Video Streaming capabilities.
- Implement user authentication and authorization mechanisms for secure access, leveraging IBM Cloud Video Streaming features.
- Conduct comprehensive testing to ensure seamless content delivery and user interactions with IBM Cloud Video Streaming.

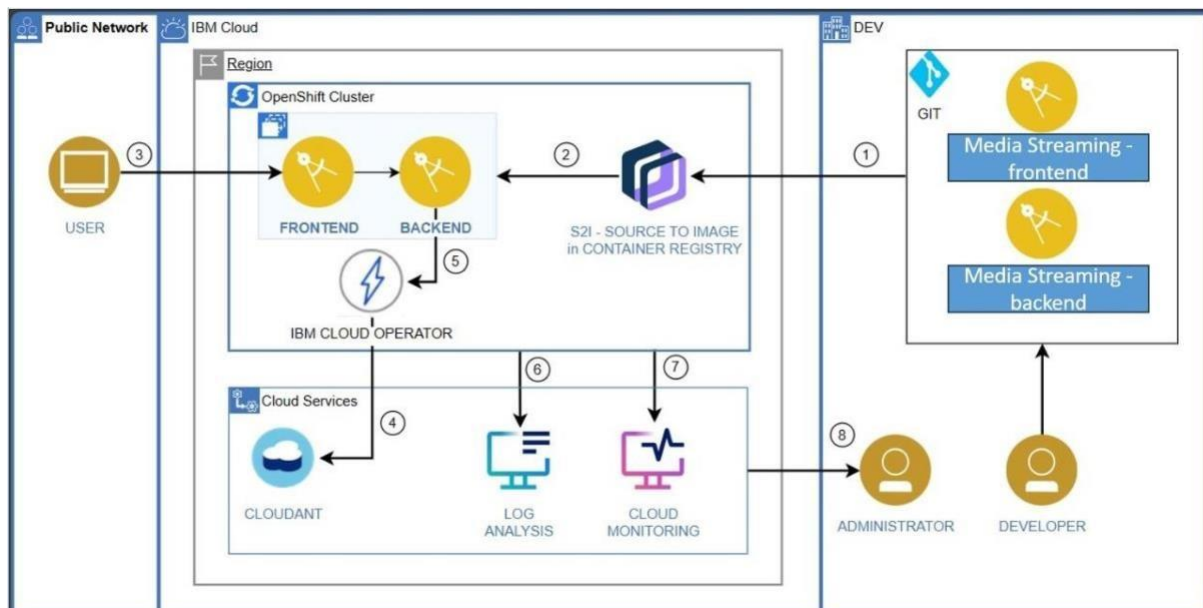
**Iterate:**

Continuously gather feedback and make improvements, in collaboration with IBM Cloud Video Streaming.

**Actions:**

- Monitor system performance, user engagement, and content popularity to identify areas for improvement, particularly leveraging IBM Cloud Video Streaming capabilities.
- Address user feedback and implement enhancements, such as refining recommendation algorithms or adding social sharing features, while integrating with IBM Cloud Video Streaming.
- Stay updated with emerging technologies and trends in media streaming to incorporate innovations into the platform, in collaboration with IBM Cloud Video Streaming.

**TECHNOLOGY ARCHITECTURE**



## Design Thinking Approach for Media Streaming with IBM Cloud Video Streaming

### 1. Content Acquisition and Storage:

- Content Sources: Utilize various sources to acquire video content, including user uploads, licensed content, and streaming services.
- Cloud Storage: Store video files in IBM Cloud Object Storage for efficient retrieval and distribution.

### 2. Content Preprocessing:

- Quality Assurance: Ensure video content meets specified quality standards, addressing issues like resolution, aspect ratio, and compression artifacts.
- Transcoding: Convert videos into multiple formats and resolutions to accommodate various user devices and network conditions.

### 3. Platform Development and Integration:

- IBM Cloud Video Streaming: Leverage IBM Cloud Video Streaming for hosting and managing video content.



- API Integration: Integrate IBM Cloud Video Streaming APIs for seamless content upload, retrieval, and streaming.

#### **4. User Interface (UI) Design:**

- Create an intuitive UI for users to browse and interact with the media streaming platform.
- Implement features for content search, recommendations, and user-generated playlists.

#### **5. Video Player Customization:**

- Customize the video player using HTML, CSS, and JavaScript to provide a branded and user-friendly viewing experience.
- Implement features like adaptive bitrate streaming for optimal playback quality.

#### **6. User Authentication and Authorization:**

- Implement secure user authentication mechanisms to control access to premium content and user-specific features.
- Utilize IBM Cloud Identity and Access Management (IAM) for authentication.

#### **7. Monetization (Optional):**

- Implement monetization strategies such as subscription models, pay-per-view, or ad-based revenue streams using IBM Monetize.

#### **8. Analytics and Insights:**

- Integrate analytics tools to gather user engagement data, including views, watch time, and popular content.
- Utilize IBM Cloud Analytics for in-depth insights into user behavior and content performance.

## **9. Content Recommendation Engine (Optional):**

- Develop a recommendation engine using machine learning algorithms to suggest personalized content to users.

## **10. Social Integration (Optional):**

- Implement social sharing features to allow users to share content on social media platforms.

## **11. Content Moderation (Optional):**

- Integrate content moderation tools to ensure compliance with community guidelines and prevent inappropriate or offensive content from being published.

## **12. Continuous Improvement and Innovation:**

- Stay updated with emerging technologies in media streaming and leverage IBM Cloud services for ongoing enhancements.
- Gather user feedback and conduct usability testing to refine features and optimize the user experience.

# **MODULES DESCRIPTION**

## **1. Content Acquisition and Storage Module:**

- Objective: Acquire and store video content efficiently for the media streaming platform.
- Key Tasks:
  - Gather video content from various sources, including user uploads and licensed content.
  - Store video files in IBM Cloud Object Storage for easy retrieval.

## **2. Content Preprocessing Module:**

- Objective: Prepare video content for optimal streaming quality and compatibility.
- Key Tasks:
  - Ensure video quality meets specified standards, addressing resolution, aspect ratio, and compression.
  - Transcode videos into multiple formats and resolutions for diverse user devices and network conditions.

## **3. Platform Development and Integration Module:**

- Objective: Utilize IBM Cloud Video Streaming for hosting and managing video content.
- Key Tasks:
  - Integrate IBM Cloud Video Streaming APIs for seamless content upload, retrieval, and streaming.

## **4. User Interface (UI) Design Module:**

- Objective: Design an intuitive user interface for seamless interaction with the media streaming platform.
- Key Tasks:
  - Create a user-friendly UI with features for content search, recommendations, and user-generated playlists.

## **5. Video Player Customization Module:**

- Objective: Customize the video player to provide an enhanced viewing experience.
- Key Tasks:
  - Utilize HTML, CSS, and JavaScript to customize the video player for branding and optimal playback.

## **6. User Authentication and Authorization Module:**

- Objective: Implement secure access control mechanisms for user authentication and authorization.
- Key Tasks:
  - Integrate IBM Cloud Identity and Access Management (IAM) for authentication and authorization.

## **7. Monetization Strategies Module (Optional):**

- Objective: Implement revenue-generating models for the media streaming platform.
- Key Tasks:
  - Introduce monetization strategies like subscription models, pay-per-view, or ad-based revenue streams.

## **8. Analytics and Insights Module:**

- Objective: Gather and analyze user engagement data to gain valuable insights.
- Key Tasks:
  - Integrate analytics tools to track views, watch time, and popular content.
  - Leverage IBM Cloud Analytics for comprehensive user behavior analysis.

## **9. Continuous Improvement and Innovation Module:**

- Objective: Stay up-to-date with emerging technologies and continuously enhance the platform's features and performance.
- Key Tasks:
  - Gather user feedback and conduct usability testing to refine features and optimize the user experience.

- Explore and incorporate new technologies and trends in media streaming.

## **ALGORITHM AND TECHNOLOGY USED**

### **1. Content Acquisition and Storage:**

- Technology: IBM Cloud Object Storage
- Description: Utilize IBM Cloud Object Storage for efficient storage and retrieval of video content. This technology allows seamless handling of large video files.

### **2. Content Preprocessing:**

- Technology: HTML, CSS, JavaScript (for frontend processing)
- Description: Use HTML, CSS, and JavaScript to ensure video content meets specified quality standards, addressing resolution, aspect ratio, and compression. This step ensures optimal viewing experience.

### **3. Platform Development and Integration:**

- Technology: IBM Cloud Video Streaming APIs, HTML, CSS, PHP, JS
- Description: Leverage IBM Cloud Video Streaming APIs for hosting and managing video content. Additionally, utilize HTML, CSS, PHP, and JS for seamless integration and user interaction within the platform.

### **4. User Interface (UI) Design:**

- Technology: HTML, CSS, JavaScript
- Description: Create an intuitive user interface using HTML, CSS, and JavaScript. This UI allows users to browse, search, and interact with the media streaming platform effortlessly.

## **5. Video Player Customization:**

- Technology: HTML, CSS, JavaScript
- Description: Customize the video player using HTML, CSS, and JavaScript to provide a branded and user-friendly viewing experience. This customization ensures optimal playback quality.

## **6. User Authentication and Authorization:**

- Technology: HTML, CSS, PHP, JS
- Description: Implement secure user authentication and authorization mechanisms using HTML, CSS, PHP, and JS. This step ensures controlled access to premium content and user-specific features.

## **7. Monetization Strategies (Optional):**

- Technology: HTML, CSS, PHP, JS
- Description: If required, implement revenue-generating models using HTML, CSS, PHP, and JS. This may include subscription models, pay-per-view, or ad-based revenue streams.

## **8. Analytics and Insights:**

- Technology: IBM Cloud Analytics, HTML, CSS, PHP, JS
- Description: Integrate analytics tools, like IBM Cloud Analytics, to gather user engagement data. Use HTML, CSS, PHP, and JS for comprehensive user behavior analysis, tracking views, watch time, and popular content.

## **9. Continuous Improvement and Innovation:**

- Technology: HTML, CSS, PHP, JS
- Description: Stay updated with emerging technologies and trends in media streaming. Continuously enhance the platform's features and

performance using HTML, CSS, PHP, and JS. Gather user feedback and conduct usability testing for refinement.

Utilizing a combination of HTML, CSS, PHP, JS, and IBM Cloud Video Streaming, this project ensures an interactive and user-friendly media streaming platform with features for content customization, monetization (optional), and comprehensive analytics for user engagement insights. Continuous improvement and innovation are emphasized to adapt to evolving trends in media streaming.

## **PROJECT DEVELOPMENT STEPS AND SCREENSHOT**

## Front-End Development

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta http-equiv="X-UA-Compatible" content="IE=edge">
7      <meta name="viewport" content="width=device-width, initial-scale=1.0">
8      <link rel="stylesheet" href="homepage.css">
9      <title>Cloudflix - Watch your uploaded videos anywhere & anytime</title>
10 </head>
11 <body>
12
13     <div class="container">
14         <nav class="navbar">
15             <div class="left">
16                 
17             </div>
18             <div class="right">
19                 <select name="language" class="language">
20                     <option value="English">English</option>
21                     <option value="Hindi">Tamil</option>
22                 </select>
23                 <button><a href="login.html">Login</a></button>
24             </div>
25         </nav>
26
27         <div class="title">
28             <div class="content">
29                 <h1>Upload your videos and watch it anytime</h1>
30                 <h2>Watch anywhere. Cancel anytime.</h2>
31                 <form action="#">
32                     <h3>Ready to watch? Enter your email to create your account.</h3>
33                     <div class="email">
34                         <input type="email" name="email" placeholder="Email address">
35                         <button><a href="signup.html">Get Started </a></button>
36                     </div>
37                 </form>
38             </div>
39         </div>
40     </div>
41 </body>
42 </html>
```



```

1  *
2
3  padding: 0;
4  margin: 0;
5  background-color: #fff;
6
7  *
8  width: 100%;
9  height: 100%;
10 color: #333;
11 background-color: #fff;
12 font-family: sans-serif;
13
14 *
15 container: none;
16 content: none;
17 background: url(images/water.jpg) no-repeat center center/cover;
18 position: absolute;
19 height: 100%;
20 width: 100%;
21 opacity: 0.5;
22 z-index: 1;
23
24 *
25 container: none;
26 height: 100%;
27 width: 100%;
28 position: absolute;
29
30 *
31 container: none;
32 height: 100%;
33 width: 100%;
34 position: absolute;
35
36 *
37 container: none;
38 height: 100%;
39 width: 100%;
40 position: absolute;
41
42 *
43 container: none;
44 height: 100%;
45 width: 100%;
46 position: absolute;
47
48 *
49 container: none;
50 height: 100%;
51 width: 100%;
52 position: absolute;
53
54 *
55 container: none;
56 height: 100%;
57 width: 100%;
58 position: absolute;
59
60 *
61 container: none;
62 height: 100%;
63 width: 100%;
64 position: absolute;
65
66 *
67 container: none;
68 height: 100%;
69 width: 100%;
70 position: absolute;
71
72 *
73 container: none;
74 height: 100%;
75 width: 100%;
76 position: absolute;
77
78 *
79 container: none;
80 height: 100%;
81 width: 100%;
82 position: absolute;
83
84 *
85 container: none;
86 height: 100%;
87 width: 100%;
88 position: absolute;
89
90 *
91 container: none;
92 height: 100%;
93 width: 100%;
94 position: absolute;
95
96 *
97 container: none;
98 height: 100%;
99 width: 100%;
100 position: absolute;
101
102 *
103 container: none;
104 height: 100%;
105 width: 100%;
106 position: absolute;
107
108 *
109 container: none;
110 height: 100%;
111 width: 100%;
112 position: absolute;
113
114 *
115 container: none;
116 height: 100%;
117 width: 100%;
118 position: absolute;
119
120 *
121 container: none;
122 height: 100%;
123 width: 100%;
124 position: absolute;
125
126 *
127 container: none;
128 height: 100%;
129 width: 100%;
130 position: absolute;
131
132 *
133 container: none;
134 height: 100%;
135 width: 100%;
136 position: absolute;
137
138 *
139 container: none;
140 height: 100%;
141 width: 100%;
142 position: absolute;
143
144 *
145 container: none;
146 height: 100%;
147 width: 100%;
148 position: absolute;
149
150 *
151 container: none;
152 height: 100%;
153 width: 100%;
154 position: absolute;
155
156 *
157 container: none;
158 height: 100%;
159 width: 100%;
160 position: absolute;
161
162 *
163 container: none;
164 height: 100%;
165 width: 100%;
166 position: absolute;
167
168 *
169 container: none;
170 height: 100%;
171 width: 100%;
172 position: absolute;
173
174 *
175 container: none;
176 height: 100%;
177 width: 100%;
178 position: absolute;
179
180 *
181 container: none;
182 height: 100%;
183 width: 100%;
184 position: absolute;
185
186 *
187 container: none;
188 height: 100%;
189 width: 100%;
190 position: absolute;
191
192 *
193 container: none;
194 height: 100%;
195 width: 100%;
196 position: absolute;
197
198 *
199 container: none;
200 height: 100%;
201 width: 100%;
202 position: absolute;
203
204 *
205 container: none;
206 height: 100%;
207 width: 100%;
208 position: absolute;
209
210 *
211 container: none;
212 height: 100%;
213 width: 100%;
214 position: absolute;
215
216 *
217 container: none;
218 height: 100%;
219 width: 100%;
220 position: absolute;
221
222 *
223 container: none;
224 height: 100%;
225 width: 100%;
226 position: absolute;
227
228 *
229 container: none;
230 height: 100%;
231 width: 100%;
232 position: absolute;
233
234 *
235 container: none;
236 height: 100%;
237 width: 100%;
238 position: absolute;
239
240 *
241 container: none;
242 height: 100%;
243 width: 100%;
244 position: absolute;
245
246 *
247 container: none;
248 height: 100%;
249 width: 100%;
250 position: absolute;
251
252 *
253 container: none;
254 height: 100%;
255 width: 100%;
256 position: absolute;
257
258 *
259 container: none;
260 height: 100%;
261 width: 100%;
262 position: absolute;
263
264 *
265 container: none;
266 height: 100%;
267 width: 100%;
268 position: absolute;
269
270 *
271 container: none;
272 height: 100%;
273 width: 100%;
274 position: absolute;
275
276 *
277 container: none;
278 height: 100%;
279 width: 100%;
280 position: absolute;
281
282 *
283 container: none;
284 height: 100%;
285 width: 100%;
286 position: absolute;
287
288 *
289 container: none;
290 height: 100%;
291 width: 100%;
292 position: absolute;
293
294 *
295 container: none;
296 height: 100%;
297 width: 100%;
298 position: absolute;
299
300 *
301 container: none;
302 height: 100%;
303 width: 100%;
304 position: absolute;
305
306 *
307 container: none;
308 height: 100%;
309 width: 100%;
310 position: absolute;
311
312 *
313 container: none;
314 height: 100%;
315 width: 100%;
316 position: absolute;
317
318 *
319 container: none;
320 height: 100%;
321 width: 100%;
322 position: absolute;
323
324 *
325 container: none;
326 height: 100%;
327 width: 100%;
328 position: absolute;
329
330 *
331 container: none;
332 height: 100%;
333 width: 100%;
334 position: absolute;
335
336 *
337 container: none;
338 height: 100%;
339 width: 100%;
340 position: absolute;
341
342 *
343 container: none;
344 height: 100%;
345 width: 100%;
346 position: absolute;
347
348 *
349 container: none;
350 height: 100%;
351 width: 100%;
352 position: absolute;
353
354 *
355 container: none;
356 height: 100%;
357 width: 100%;
358 position: absolute;
359
360 *
361 container: none;
362 height: 100%;
363 width: 100%;
364 position: absolute;
365
366 *
367 container: none;
368 height: 100%;
369 width: 100%;
370 position: absolute;
371
372 *
373 container: none;
374 height: 100%;
375 width: 100%;
376 position: absolute;
377
378 *
379 container: none;
380 height: 100%;
381 width: 100%;
382 position: absolute;
383
384 *
385 container: none;
386 height: 100%;
387 width: 100%;
388 position: absolute;
389
390 *
391 container: none;
392 height: 100%;
393 width: 100%;
394 position: absolute;
395
396 *
397 container: none;
398 height: 100%;
399 width: 100%;
400 position: absolute;
401
402 *
403 container: none;
404 height: 100%;
405 width: 100%;
406 position: absolute;
407
408 *
409 container: none;
410 height: 100%;
411 width: 100%;
412 position: absolute;
413
414 *
415 container: none;
416 height: 100%;
417 width: 100%;
418 position: absolute;
419
420 *
421 container: none;
422 height: 100%;
423 width: 100%;
424 position: absolute;
425
426 *
427 container: none;
428 height: 100%;
429 width: 100%;
430 position: absolute;
431
432 *
433 container: none;
434 height: 100%;
435 width: 100%;
436 position: absolute;
437
438 *
439 container: none;
440 height: 100%;
441 width: 100%;
442 position: absolute;
443
444 *
445 container: none;
446 height: 100%;
447 width: 100%;
448 position: absolute;
449
450 *
451 container: none;
452 height: 100%;
453 width: 100%;
454 position: absolute;
455
456 *
457 container: none;
458 height: 100%;
459 width: 100%;
460 position: absolute;
461
462 *
463 container: none;
464 height: 100%;
465 width: 100%;
466 position: absolute;
467
468 *
469 container: none;
470 height: 100%;
471 width: 100%;
472 position: absolute;
473
474 *
475 container: none;
476 height: 100%;
477 width: 100%;
478 position: absolute;
479
480 *
481 container: none;
482 height: 100%;
483 width: 100%;
484 position: absolute;
485
486 *
487 container: none;
488 height: 100%;
489 width: 100%;
490 position: absolute;
491
492 *
493 container: none;
494 height: 100%;
495 width: 100%;
496 position: absolute;
497
498 *
499 container: none;
500 height: 100%;
501 width: 100%;
502 position: absolute;
503
504 *
505 container: none;
506 height: 100%;
507 width: 100%;
508 position: absolute;
509
510 *
511 container: none;
512 height: 100%;
513 width: 100%;
514 position: absolute;
515
516 *
517 container: none;
518 height: 100%;
519 width: 100%;
520 position: absolute;
521
522 *
523 container: none;
524 height: 100%;
525 width: 100%;
526 position: absolute;
527
528 *
529 container: none;
530 height: 100%;
531 width: 100%;
532 position: absolute;
533
534 *
535 container: none;
536 height: 100%;
537 width: 100%;
538 position: absolute;
539
540 *
541 container: none;
542 height: 100%;
543 width: 100%;
544 position: absolute;
545
546 *
547 container: none;
548 height: 100%;
549 width: 100%;
550 position: absolute;
551
552 *
553 container: none;
554 height: 100%;
555 width: 100%;
556 position: absolute;
557
558 *
559 container: none;
560 height: 100%;
561 width: 100%;
562 position: absolute;
563
564 *
565 container: none;
566 height: 100%;
567 width: 100%;
568 position: absolute;
569
570 *
571 container: none;
572 height: 100%;
573 width: 100%;
574 position: absolute;
575
576 *
577 container: none;
578 height: 100%;
579 width: 100%;
580 position: absolute;
581
582 *
583 container: none;
584 height: 100%;
585 width: 100%;
586 position: absolute;
587
588 *
589 container: none;
590 height: 100%;
591 width: 100%;
592 position: absolute;
593
594 *
595 container: none;
596 height: 100%;
597 width: 100%;
598 position: absolute;
599
600 *
601 container: none;
602 height: 100%;
603 width: 100%;
604 position: absolute;
605
606 *
607 container: none;
608 height: 100%;
609 width: 100%;
610 position: absolute;
611
612 *
613 container: none;
614 height: 100%;
615 width: 100%;
616 position: absolute;
617
618 *
619 container: none;
620 height: 100%;
621 width: 100%;
622 position: absolute;
623
624 *
625 container: none;
626 height: 100%;
627 width: 100%;
628 position: absolute;
629
630 *
631 container: none;
632 height: 100%;
633 width: 100%;
634 position: absolute;
635
636 *
637 container: none;
638 height: 100%;
639 width: 100%;
640 position: absolute;
641
642 *
643 container: none;
644 height: 100%;
645 width: 100%;
646 position: absolute;
647
648 *
649 container: none;
650 height: 100%;
651 width: 100%;
652 position: absolute;
653
654 *
655 container: none;
656 height: 100%;
657 width: 100%;
658 position: absolute;
659
660 *
661 container: none;
662 height: 100%;
663 width: 100%;
664 position: absolute;
665
666 *
667 container: none;
668 height: 100%;
669 width: 100%;
670 position: absolute;
671
672 *
673 container: none;
674 height: 100%;
675 width: 100%;
676 position: absolute;
677
678 *
679 container: none;
680 height: 100%;
681 width: 100%;
682 position: absolute;
683
684 *
685 container: none;
686 height: 100%;
687 width: 100%;
688 position: absolute;
689
690 *
691 container: none;
692 height: 100%;
693 width: 100%;
694 position: absolute;
695
696 *
697 container: none;
698 height: 100%;
699 width: 100%;
700 position: absolute;
701
702 *
703 container: none;
704 height: 100%;
705 width: 100%;
706 position: absolute;
707
708 *
709 container: none;
710 height: 100%;
711 width: 100%;
712 position: absolute;
713
714 *
715 container: none;
716 height: 100%;
717 width: 100%;
718 position: absolute;
719
720 *
721 container: none;
722 height: 100%;
723 width: 100%;
724 position: absolute;
725
726 *
727 container: none;
728 height: 100%;
729 width: 100%;
730 position: absolute;
731
732 *
733 container: none;
734 height: 100%;
735 width: 100%;
736 position: absolute;
737
738 *
739 container: none;
740 height: 100%;
741 width: 100%;
742 position: absolute;
743
744 *
745 container: none;
746 height: 100%;
747 width: 100%;
748 position: absolute;
749
750 *
751 container: none;
752 height: 100%;
753 width: 100%;
754 position: absolute;
755
756 *
757 container: none;
758 height: 100%;
759 width: 100%;
760 position: absolute;
761
762 *
763 container: none;
764 height: 100%;
765 width: 100%;
766 position: absolute;
767
768 *
769 container: none;
770 height: 100%;
771 width: 100%;
772 position: absolute;
773
774 *
775 container: none;
776 height: 100%;
777 width: 100%;
778 position: absolute;
779
780 *
781 container: none;
782 height: 100%;
783 width: 100%;
784 position: absolute;
785
786 *
787 container: none;
788 height: 100%;
789 width: 100%;
790 position: absolute;
791
792 *
793 container: none;
794 height: 100%;
795 width: 100%;
796 position: absolute;
797
798 *
799 container: none;
800 height: 100%;
801 width: 100%;
802 position: absolute;
803
804 *
805 container: none;
806 height: 100%;
807 width: 100%;
808 position: absolute;
809
810 *
811 container: none;
812 height: 100%;
813 width: 100%;
814 position: absolute;
815
816 *
817 container: none;
818 height: 100%;
819 width: 100%;
820 position: absolute;
821
822 *
823 container: none;
824 height: 100%;
825 width: 100%;
826 position: absolute;
827
828 *
829 container: none;
830 height: 100%;
831 width: 100%;
832 position: absolute;
833
834 *
835 container: none;
836 height: 100%;
837 width: 100%;
838 position: absolute;
839
840 *
841 container: none;
842 height: 100%;
843 width: 100%;
844 position: absolute;
845
846 *
847 container: none;
848 height: 100%;
849 width: 100%;
850 position: absolute;
851
852 *
853 container: none;
854 height: 100%;
855 width: 100%;
856 position: absolute;
857
858 *
859 container: none;
860 height: 100%;
861 width: 100%;
862 position: absolute;
863
864 *
865 container: none;
866 height: 100%;
867 width: 100%;
868 position: absolute;
869
870 *
871 container: none;
872 height: 100%;
873 width: 100%;
874 position: absolute;
875
876 *
877 container: none;
878 height: 100%;
879 width: 100%;
880 position: absolute;
881
882 *
883 container: none;
884 height: 100%;
885 width: 100%;
886 position: absolute;
887
888 *
889 container: none;
890 height: 100%;
891 width: 100%;
892 position: absolute;
893
894 *
895 container: none;
896 height: 100%;
897 width: 100%;
898 position: absolute;
899
899

```

```
1  <!DOCTYPE html>
2  <!-- Coding By CodingNepal - codingnepalweb.com -->
3  <html lang="en" dir="ltr">
4    <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title> Cloudflif - Registration or Sign Up </title>
8      <link rel="stylesheet" href="signup.css">
9    </head>
10   <body>
11     <div class="wrapper">
12       <h2>Sign Up</h2>
13       <form action="main.html">
14         <div class="input-box">
15           <input type="text" placeholder="Enter your name" required>
16         </div>
17         <div class="input-box">
18           <input type="text" placeholder="Enter your email" required>
19         </div>
20         <div class="input-box">
21           <input type="password" placeholder="Create password" required>
22         </div>
23         <div class="input-box">
24           <input type="password" placeholder="Confirm password" required>
25         </div>
26         <div class="policy">
27           <input type="checkbox">
28           <h3>I accept all terms & condition</h3>
29         </div>
30         <div class="input-box button">
31           <input type="Submit" value="Sign Up">
32         </div>
33         <div class="text">
34           <h3>Already have an account? <a href="login.html">Login now</a></h3>
35         </div>
36       </form>
37     </div>
38   </body>
39 </html>
```



```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Cloudflifx - Login page </title>
7   <link rel="stylesheet" href="login.css">
8   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.3/css/all.min.css"/>
9 </head>
10 <body>
11   <div class="wrapper">
12     <header>Login</header>
13     <form action="">
14       <div class="field email">
15         <div class="input-area">
16           <input type="text" placeholder="Email Address">
17           <i class="icon fas fa-envelope"></i>
18           <i class="error error-icon fas fa-exclamation-circle"></i>
19         </div>
20         <div class="error error-txt">Email can't be blank</div>
21       </div>
22       <div class="field password">
23         <div class="input-area">
24           <input type="password" placeholder="Password">
25           <i class="icon fas fa-lock"></i>
26           <i class="error error-icon fas fa-exclamation-circle"></i>
27         </div>
28         <div class="error error-txt">Password can't be blank</div>
29       </div>
30       <div class="pass-txt"><a href="#">Forgot password?</a></div>
31       <input type="submit" value="Login">
32     </form>
33     <div class="sign-txt"><p>Need an account?</p> <a href="signup.html">Signup now</a></div>
34   </div>
35
36   <script src="login.js"></script>
37
38 </body>
39 </html>

```

```

1 // Login.js
2
3 // Selectors
4 const emailInput = document.querySelector('input[type="text"]');
5 const passwordInput = document.querySelector('input[type="password"]');
6 const loginForm = document.querySelector('form');
7 const forgotLink = document.querySelector('a[href="#"]');
8 const signupLink = document.querySelector('a[href="signup.html"]');
9
10 // Event Listeners
11 loginForm.addEventListener('submit', login);
12 forgotLink.addEventListener('click', forgotPassword);
13 signupLink.addEventListener('click', signup);
14
15 // Login Function
16 function login(e) {
17   e.preventDefault();
18   const email = emailInput.value;
19   const password = passwordInput.value;
20
21   // Validation
22   if (!email || !password) {
23     showError('Email and password are required');
24     return;
25   }
26
27   // Simulating API call
28   setTimeout(() => {
29     // Success
30     if (email === 'admin@gmail.com' & password === '12345678') {
31       // Redirect to dashboard
32       window.location.href = 'dashboard.html';
33     } else {
34       // Invalid credentials
35       showError('Invalid email or password');
36     }
37   }, 1000);
38 }
39
40 // Forgot Password Function
41 function forgotPassword() {
42   // Redirect to forgot password page
43   window.location.href = 'forgot-password.html';
44 }
45
46 // Signup Function
47 function signup() {
48   // Redirect to signup page
49   window.location.href = 'signup.html';
50 }
51
52 // Show Error Function
53 function showError(message) {
54   // Create error message element
55   const errorDiv = document.createElement('div');
56   errorDiv.classList.add('error', 'error-txt');
57   errorDiv.textContent = message;
58
59   // Append to the form
60   loginForm.appendChild(errorDiv);
61
62   // Hide error message after 5 seconds
63   setTimeout(() => {
64     errorDiv.remove();
65   }, 5000);
66 }

```

```

1  const form = document.querySelector("form");
2  eField = form.querySelector(".email");
3  eInput = eField.querySelector("input");
4  pField = form.querySelector(".password");
5  pInput = pField.querySelector("input");
6
7  form.onsubmit = (e) => {
8    e.preventDefault(); //preventing from form submitting
9    //if email and password is blank then add shake class in it else call specified function
10   (eInput.value == "") ? eField.classList.add("shake", "error") : checkEmail();
11   (pInput.value == "") ? pField.classList.add("shake", "error") : checkPass();
12
13   setTimeout(() => { //remove shake class after 500ms
14     eField.classList.remove("shake");
15     pField.classList.remove("shake");
16   }, 500);
17
18   eInput.onkeyup = () => {checkEmail();} //calling checkEmail function on email input keyup
19   pInput.onkeyup = () => {checkPass();} //calling checkPassword function on pass input keyup
20
21   function checkEmail(){ //checkEmail function
22     let pattern = /^[^ ]+@[^ ]+\.[a-z]{2,3}$/; //pattern for validate email
23     if(!eInput.value.match(pattern)){ //if pattern not matched then add error and remove valid class
24       eField.classList.add("error");
25       eField.classList.remove("valid");
26       let errorTxt = eField.querySelector(".error-txt");
27       //if email value is not empty then show please enter valid email else show Email can't be blank
28       (eInput.value != "") ? errorTxt.innerText = "Enter a valid email address" : errorTxt.innerText = "Email can't be blank";
29     }else{ //if pattern matched then remove error and add valid class
30       eField.classList.remove("error");
31       eField.classList.add("valid");
32     }
33   }
34
35   function checkPass(){ //checkPass function
36     if(pInput.value == ""){ //if pass is empty then add error and remove valid class
37       pField.classList.add("error");
38       pField.classList.remove("valid");
39     }else{ //if pass is empty then remove error and add valid class
40       pField.classList.remove("error");
41       pField.classList.add("valid");
42     }
43   }
44
45   //if eField and pField doesn't contains error class that mean user filled details properly
46   if(!eField.classList.contains("error") && !pField.classList.contains("error")){
47     window.location.href = form.getAttribute("action"); //redirecting user to the specified url which is inside action attribute of form tag
48   }
49 }

```

## Back-end Development

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>video upload php and mysql</title>
7   <style>
8     body {
9       display: flex;
10      justify-content: center;
11      align-items: center;
12      flex-direction: column;
13      min-height: 100vh;
14    }
15    input {
16      font-size: 2rem;
17    }
18    a {
19      text-decoration: none;
20      color: #006CFF;
21      font-size: 1.5rem;
22    }
23  </style>
24 </head>
25 <body>
26   <a href="view.php">Videos</a>
27   <?php if (isset($_GET['error'])) { ?>
28     <p><?=$_GET['error']?></p>
29   <?php } ?>
30   <form action="upload.php"
31     method="post"
32     enctype="multipart/form-data">
33
34     <input type="file"
35       name="my_video">
36
37     <input type="submit"
38       name="submit"
39       value="Upload">
40   </form>
41 </body>
42 </html>
```

```
1 <?php
2
3 $sname = "localhost";
4 $uname = "root";
5 $password = "";
6
7 $db_name = "test_db";
8
9 $conn = mysqli_connect($sname, $uname, $password, $db_name);
10
11 if (!$conn) {
12   echo "Connection failed!";
13   exit();
14 }
```

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>View</title>
7   <style>
8     body {
9       display: flex;
10      justify-content: center;
11      align-items: center;
12      flex-wrap: wrap;
13      min-height: 100vh;
14    }
15    video {
16      width: 640px;
17      height: 360px;
18    }
19    a {
20      text-decoration: none;
21      color: #006CFF;
22      font-size: 1.5rem;
23    }
24  </style>
25 </head>
26 <body>
27   <a href="index.php">UPLOAD</a>
28
29   <div class="alb">
30     <?php
31       include "db_conn.php";
32       $sql = "SELECT * FROM videos ORDER BY Id DESC";
33       $res = mysqli_query($conn, $sql);
34
35       if (mysqli_num_rows($res) > 0) {
36         while ($video = mysqli_fetch_assoc($res)) {
37
38           <video src="uploads/<?=$video['video_url']?>"
39             controls>
40
41           </video>
42
43         <?php
44         }
45       }else {
46         echo "<h1>Empty</h1>";
47       }
48     }
49   <?>
50 </div>
51 </body>
52 </html>

```

```

1 <?php
2   $file_name = $_FILES['file']['name'];
3   $tmp_name = $_FILES['file']['tmp_name'];
4   $file_up_name = time() . $file_name;
5   move_uploaded_file($tmp_name, "files/" . $file_up_name); // Specify the path where you want to save the video files
6 >?>

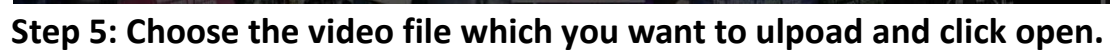
```

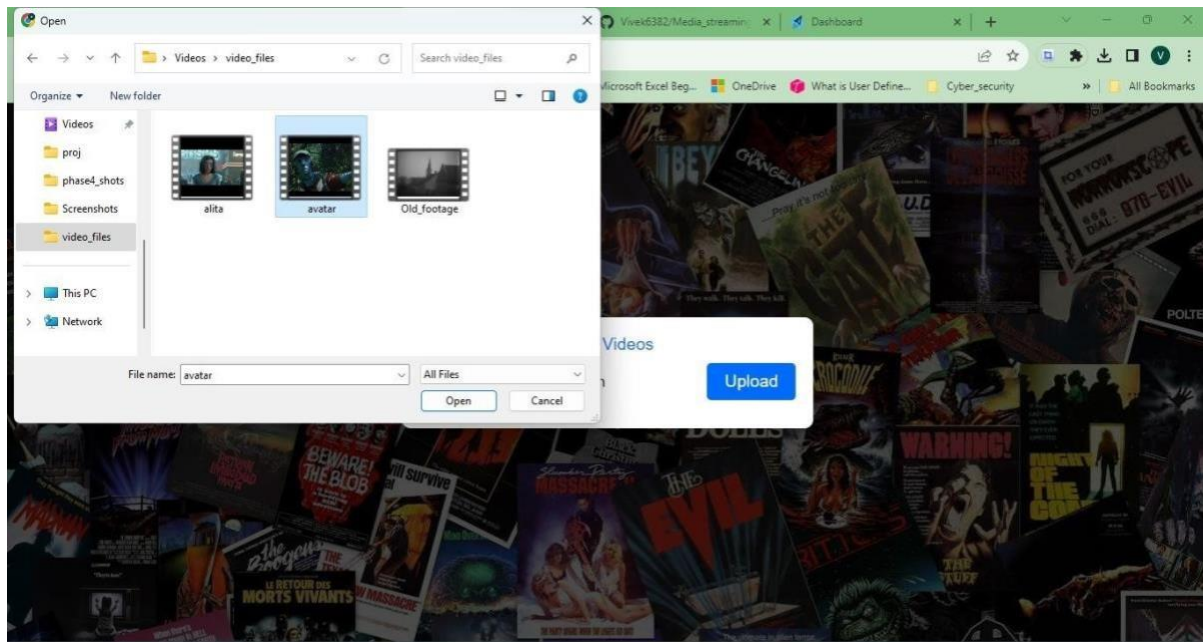
**Step 1: Enter your e-mail id to create your account in cloudflax.**



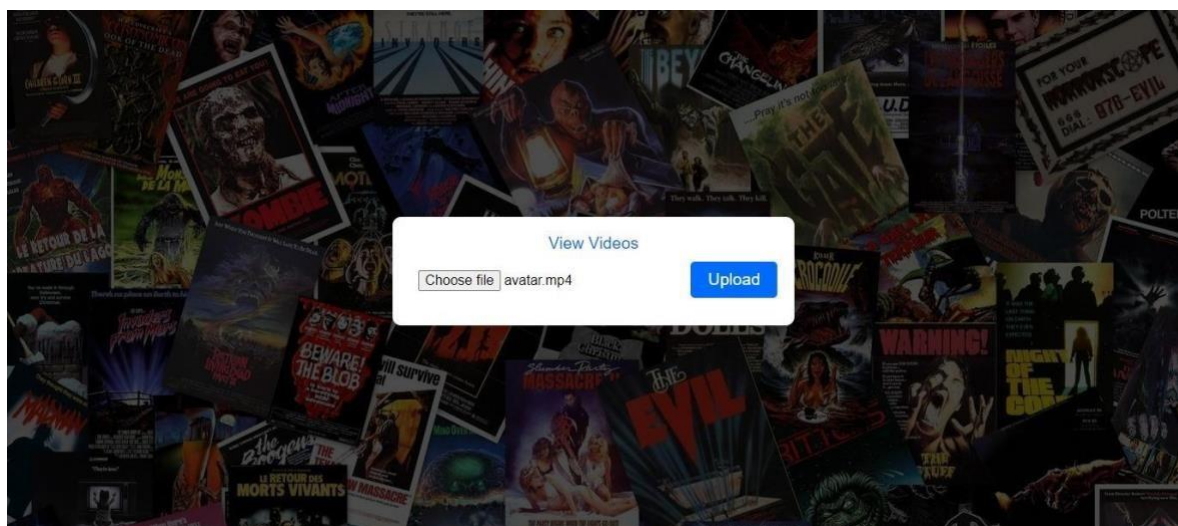




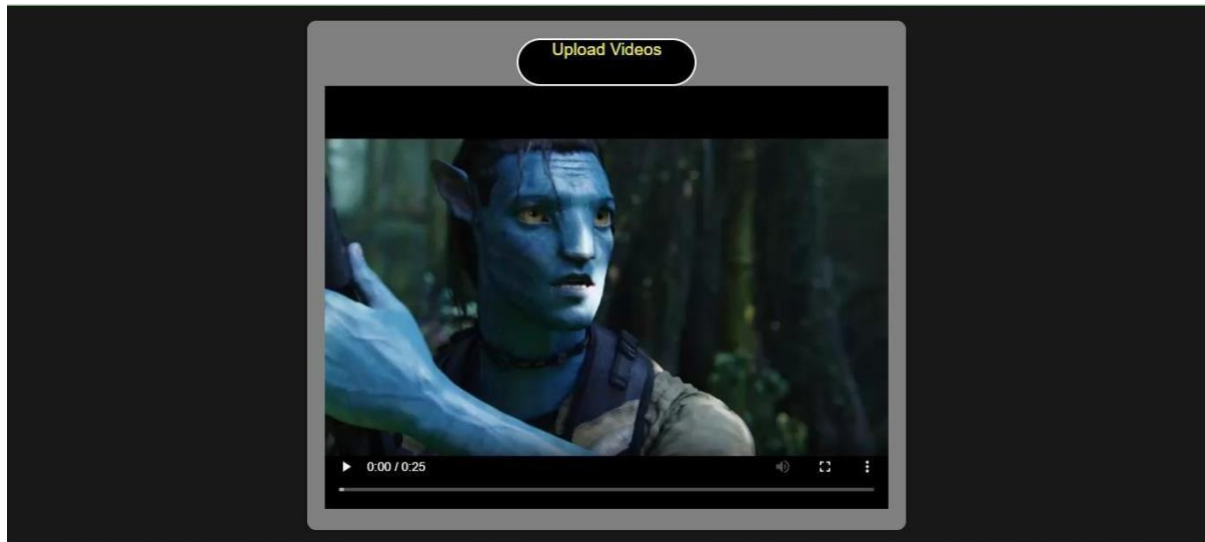




**Step 6: Now the name of the chosen file will be displayed.**

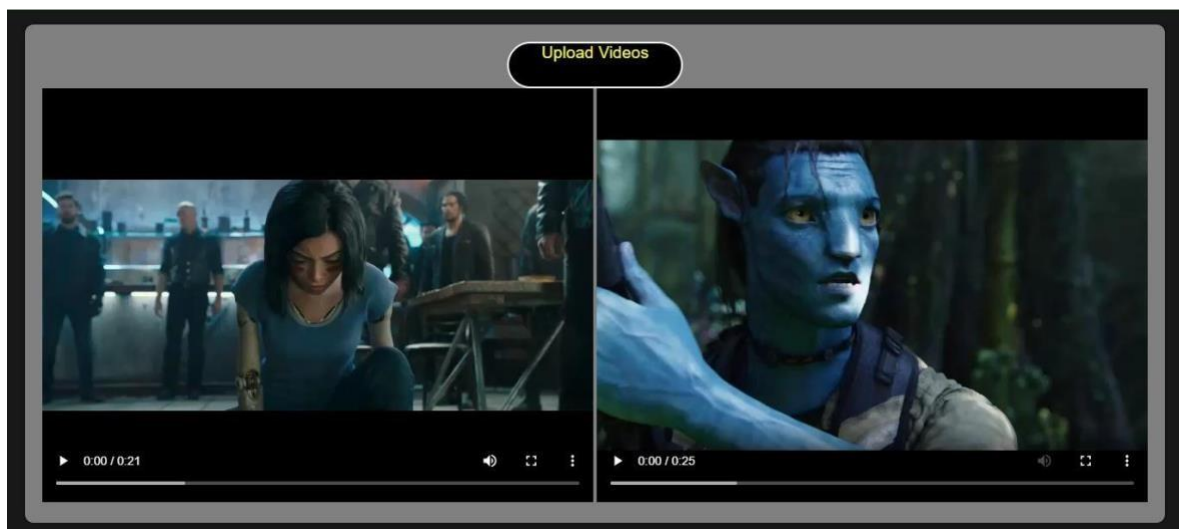
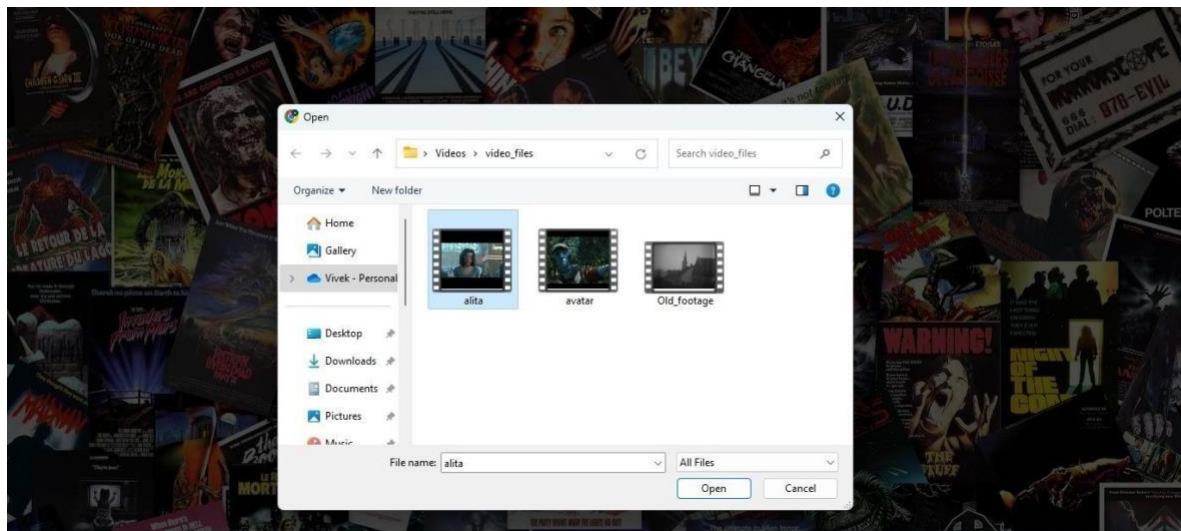


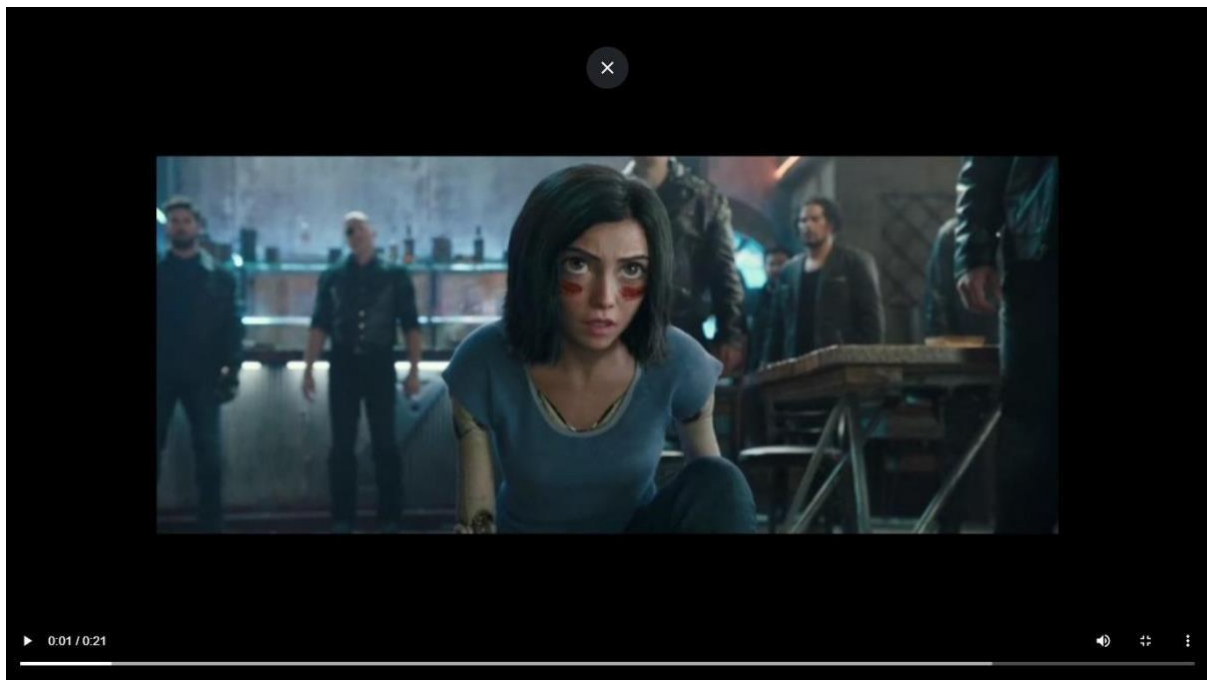
**Step 7: Now click upload to upload the selected video file and now the video will be displayed .**



**Step 8: Likewise you can upload another video and it will be displayed with the previously uploaded video.**







## CONCLUSION

In conclusion, the implementation of cloud media streaming exemplifies the boundless opportunities technology offers in the realm of content delivery and consumption. With the ability to seamlessly stream media from remote servers to various devices, it not only revolutionizes the way we access and enjoy content but also underscores the immense potential of cloud-based solutions. The scalability, convenience, and cost-efficiency of cloud media streaming make it a game-changer for both consumers and providers, shaping the future of entertainment and information dissemination. As we continue to witness rapid advancements in this field, it is clear that cloud media streaming will play a pivotal role in redefining the digital landscape and how we interact with media.