

در این پروژه، یک سامانه وب ساده برای نمایش و بررسی فیلم‌های کوتاه یا تریلرها با استفاده از Flask پیاده‌سازی شده است. صفحه اصلی سامانه شامل لیست کاملی از فیلم‌ها است که کاربران می‌توانند با کلیک بر روی هر فیلم، به صفحه‌ی نمایش اصلی آن منتقل شوند. در این صفحه، کاربران می‌توانند ویدئوی فیلم را پخش کنند. علاوه بر آن صفحه اصلی هر فیلم شامل اطلاعاتی مانند عنوان فیلم، سال تولید، ژانر، مدت زمان، خلاصه فیلم و نوع پخش (Local, DASH, CDN + DASH, CDN + HLS) می‌باشد. ابزارهای استفاده شده در این پروژه شامل ابزار Flask که یک میکروفریم‌ورک پایتون است و برای ساخت برنامه‌های وب کاربرد دارد. در این پروژه از Flask برای مدیریت روتینگ، رندرینگ صفحات HTML و مدیریت فرم‌ها استفاده شده است. از ابزارهای HTML/CSS برای طراحی صفحات وب و استایل‌دهی به عناصر استفاده شده است. ساختار فایل‌های پروژه به صورت زیر است که در ادامه به بحث در خصوص نحوه پیاده‌سازی هریک خواهیم پرداخت:

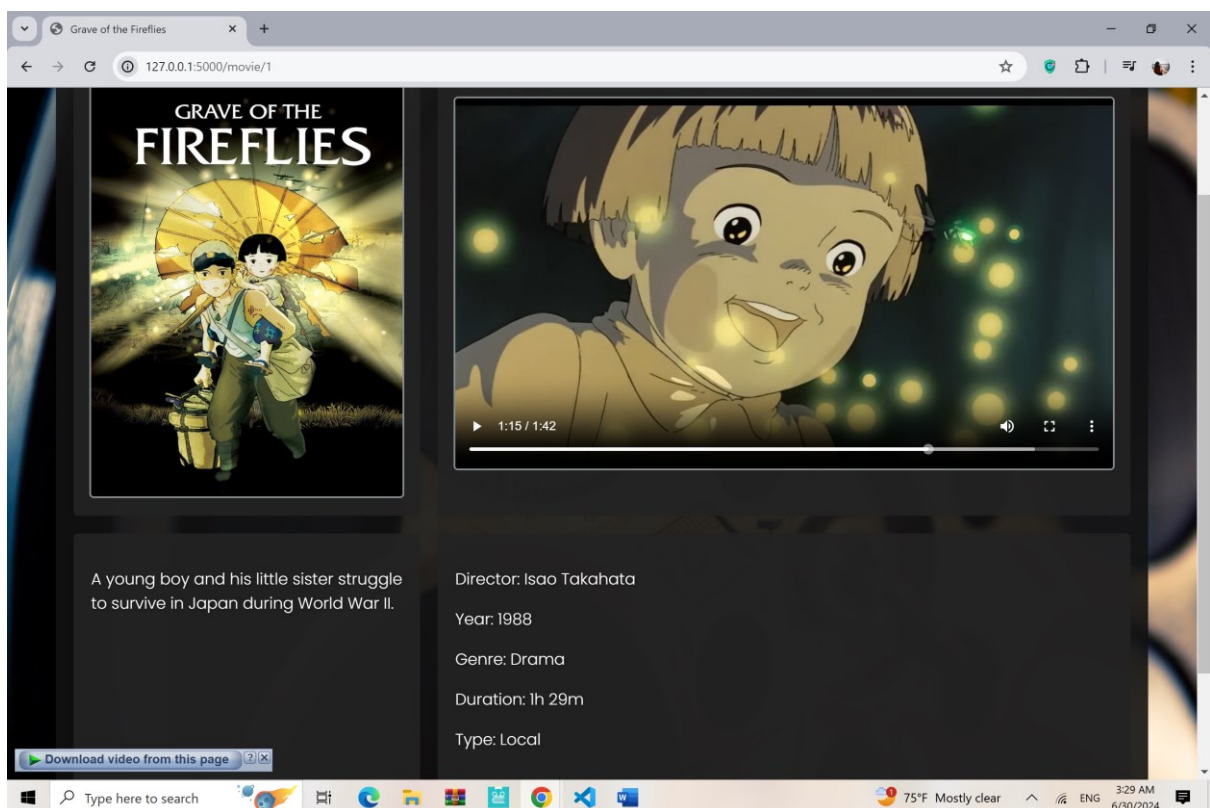
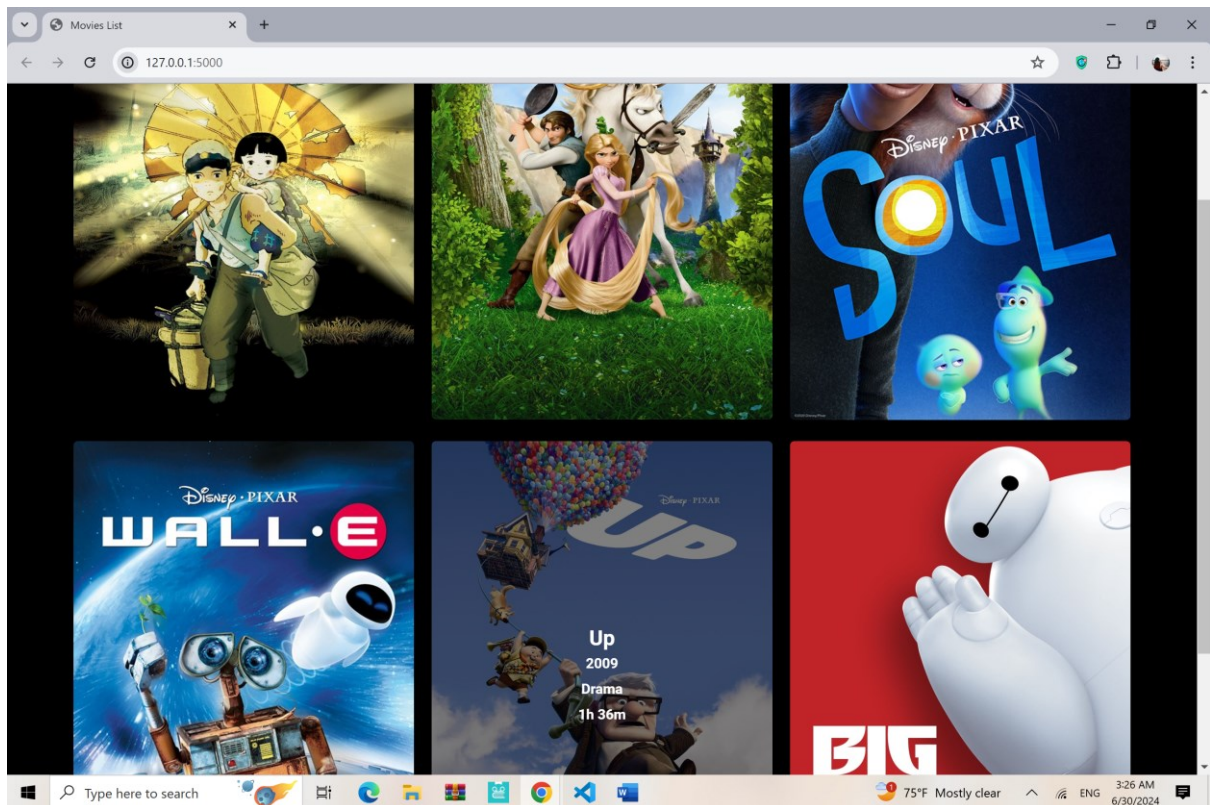
Multimedia Final Project/

```

|
├─ app.py
├─ local_dash.py
├─ requirements.txt
├─ static/
│   └─ css/
│       ├── index.css
│       └─ style.css
│   └─ images/
│       ├── movieback.jpg
│       ├── top1.PNG
│       ├── top2.PNG
│       ├── top3.PNG
│       ├── top4.PNG
│       ├── top5.PNG
│       └─ top6.PNG
├─ media/
│   ├── movie1.mp4
│   ├── movie2.mp4
│   ├── movie3/dash
│   └─ movie4/dash
└─ templates/
    ├── index.html
    ├── movie1.html
    ├── movie2.html
    ├── movie3.html
    ├── movie4.html
    ├── movie5.html
    └─ movie6.html

```

نمایی از پلتفرم پیاده‌سازی شده:



بخش app.py :

این کد یک برنامه وب ساده با استفاده از فریمورک Flask است که لیستی از فیلم‌ها را نمایش می‌دهد و به کاربر اجازه می‌دهد تا جزئیات هر فیلم را ببیند و نظرات خود را درباره هر فیلم ثبت کند.

در ابتدای کد، کتابخانه‌های مورد نیاز وارد شده‌اند که شامل Flask و توابع وابسته (render_template, request, redirect, url_for) برای ایجاد و مدیریت برنامه وب و os و json برای کار با فایل‌ها و داده‌ها.

سپس باید یک نمونه از کلاس Flask ایجاد شود که برنامه وب ما را نمایندگی می‌کند. در ادامه یک لیست از فیلم‌ها با جزئیات مربوط به هر فیلم شامل شناسه، عنوان، سال تولید، ژانر، مدت زمان، نوع فیلم، خلاصه و نظرات تعریف شده است. برای مدیریت نظرات، دو تابع تعریف شده است:

load_comments(movie_id) که نظرات مربوط به یک فیلم خاص را از یک فایل متنی بارگذاری می‌کند. اگر فایل وجود نداشته باشد، یک لیست خالی برمی‌گرداند

save_comments(movie_id, comments) که نظرات را در یک فایل متنی ذخیره می‌کند.

روت‌ها برای مدیریت درخواست‌های ورودی و نمایش صفحات مختلف استفاده می‌شوند:

روت اصلی که این روت قالب index.html را رندر می‌کند و لیست فیلم‌ها را به آن ارسال می‌کند تا در صفحه اصلی نمایش داده شوند.

روت جزئیات فیلم که در این روت، ابتدا فیلم با استفاده از شناسه (id) از لیست فیلم‌ها پیدا می‌شود. اگر فیلم پیدا نشود، خطای ۴۰۴ برگردانده می‌شود..

در نهایت نیز دستور اجرا برنامه را در حالت debug استفاده شده است که برای توسعه و دیباگینگ مفید است.

بخش local-dash.py :

این کد یک فایل ویدیویی را به چندین Representation با وضوح‌ها و بیت‌ریت‌های مختلف تبدیل کرده و سپس یک فایل DASH manifest (MPD) ایجاد می‌کند که برای Adaptive Streaming مورد استفاده قرار می‌گیرد. این کار با استفاده از کتابخانه ffmpeg انجام می‌شود.

در ابتدا کتابخانه‌های مورد نیاز وارد می‌شوند مانند: ffmpeg برای پردازش و تبدیل فایل‌های ویدیویی و OS برای مدیریت فایل‌ها و دایرکتوری‌ها.

سپس مسیر فایل‌های ورودی و خروجی تعیین شده است. در ادامه Representation های مختلف ایجاد شده است که شامل عرض، ارتفاع، بیت‌ریت ویدیویی و بیت‌ریت صوتی است.

برای ایجاد استریم‌های DASH این بخش از کد برای هر Representation یک فایل ویدیویی با وضوح و بیت‌ریت مشخص ایجاد می‌کند و آن را در دایرکتوری خروجی ذخیره می‌کند.

در بخش انتهایی کد، تمام فایل‌های ویدیویی ایجاد شده به عنوان ورودی گرفته شده و یک فایل DASH manifest با فرمت MPD ایجاد می‌شود.

روش به کار گرفته شده در این کد تبدیل ویدیو به چندین Representation با وضوح‌ها و بیت‌ریت‌های مختلف است تا یک فایل DASH manifest ایجاد کند. این فایل برای استریمینگ تطبیقی مورد استفاده قرار می‌گیرد، به این معنا که ویدیو به صورت پویا و با توجه به پهنای باند و کیفیت اتصال کاربر تنظیم می‌شود. کارآمدی نسبت به روش Local تنها این است که به کاربر اجازه می‌دهد تا بر اساس پهنای باند و کیفیت اتصال خود، به صورت پویا کیفیت ویدیو را تنظیم کند، که منجر به تجربه بهتری می‌شود و کاربران با پهنای باند بالا می‌توانند ویدیو را با کیفیت بالاتر مشاهده کنند. همچنین کاربران با پهنای باند کم می‌توانند ویدیو را با کیفیت پایین‌تر و در نتیجه مصرف پهنای باند کمتر مشاهده کنند.

اما لازم به ذکر است تبدیل یک ویدیو به چندین Representation با وضوح‌ها و بیت‌ریت‌های مختلف فرآیندی زمان‌بر است، به خصوص اگر ویدیو اصلی طولانی باشد یا تعداد نمایه‌ها زیاد باشد. ایجاد فایل manifest نیز به زمان نسبتاً زیادی نیاز دارد، اگرچه این مرحله نسبت به تبدیل ویدیوها سریع‌تر است.

بخش CSS :

بخش index.css

از فونت Roboto استفاده شده است که از Google Fonts وارد شده است. برای پس‌زمینه صفحه سیاه رنگ انتخاب شده است و فونت متن‌ها Roboto تنظیم شده است. رنگ متن عنوان‌ها سفید است و در مرکز صفحه قرار دارند. از یک شبکه برای نمایش آیت‌ها استفاده شده که به صورت خودکار ستون‌هایی با حداقل عرض ۳۰۰ پیکسل ایجاد می‌کند و شبکه در وسط صفحه قرار دارد و حاشیه‌ها به صورت خودکار تنظیم شده‌اند تا در مرکز قرار بگیرد. هر آیت موقعیت نسبی دارد و هنگام حرکت ماوس روی آن، تصویر به آرامی بزرگ‌تر شده و شفافیت آن کاهش می‌یابد که دارای یک overlay است که با قرار گرفتن ماوس روی آیت، ظاهر می‌شود و دارای پس‌زمینه نیمه شفاف سیاه است و متن آن سفید رنگ است و شامل عنوان و توضیحات می‌باشد.

بخش style.css

از فونت Poppins استفاده شده است و پس‌زمینه صفحه یک تصویر ثابت از نوار فیلم است که تمام صفحه را پوشش می‌دهد و رنگ متن‌ها سفید و پس‌زمینه کلی صفحه سیاه تنظیم شده است. رنگ متن عنوان‌ها سفید است و دارای سایه‌ای برای برجسته‌تر شدن متن هستند. تصاویر به طور کامل متناسب با فضای موجود تنظیم می‌شوند و دارای حاشیه‌های گرد و سایه‌ای ظریف هستند. ویدئوها نیز در داخل فضایشان حداکثر عرض و ارتفاع خود را حفظ می‌کنند و دارای حاشیه‌های گرد و سایه‌ای هستند.

بخش HTML:

در ساختار HTML اولیه از نوع سند `<!DOCTYPE html>` استفاده شده که نشان‌دهنده نوع سند HTML5 است و عنصر `<html lang="en">` برای تعیین زبان پیش‌فرض سند به کار رفته است. در بخش `<head>` متا تگ‌ها برای تعیین تنظیمات مربوط به رمزگذاری کاراکترها (`<meta charset="UTF-8">`)، حالت سازگاری (`<meta http-equiv="X-UA-Compatible" content="IE=edge">`) و تنظیمات نمایشگر موبایل (`<meta name="viewport" content="width=device-width, initial-scale=1.0">`) با استفاده از تگ `<link>` و تابع `url_for` به سند اضافه شده است که مسیر فایل استایل را از فولدر استاتیک دریافت می‌کند. در بخش `<body>` یک `div` با کلاس `grid-container` برای نگهداری آیتم‌های فیلم‌ها استفاده شده است. یک حلقه `for` از Jinja2 برای تکرار بر روی لیست فیلم‌ها و ایجاد عناصر HTML برای هر فیلم استفاده شده است. برای هر فیلم، یک `div` با کلاس `grid-item` ایجاد شده که با کلیک بر روی آن به صفحه جزئیات فیلم منتقل می‌شود (`onclick="location.href='{url_for('movie', id=movie.id)}'"`). یک `div` با کلاس `poster` قرار دارد که شامل تصویر فیلم و یک `overlay` با اطلاعات فیلم (عنوان، سال، ژانر، مدت زمان) است.

صفحه جزئیات فیلم اول: "Grave of the Fireflies"

از نوع سند `<!DOCTYPE html>` و عنصر `<html lang="en">` مانند صفحه `index` استفاده شده است. در بخش `<head>` متا تگ‌ها و تنظیمات مربوط به سازگاری هستند. عنوان صفحه به نام "Grave of the Fireflies" تنظیم شده است. در بخش `<body>` عنوان فیلم در یک تگ `<h1>` در بالای صفحه قرار گرفته است. یک `div` با کلاس `grid-container` شامل ۵ آیتم مختلف برای نمایش اطلاعات مختلف فیلم ایجاد شده است شامل تصویر فیلم، ویدیویی از فیلم با قابلیت کنترل پخش، خلاصه داستان فیلم، اطلاعات تکمیلی مانند کارگردان، سال تولید، ژانر، مدت زمان و نوع نمایش فیلم و امتیاز.

صفحه جزئیات فیلم دوم: "Tangled"

پیاده سازی مشابه فیلم اول است صرفاً اطلاعات تفاوت پیدا کرده‌اند.

صفحه جزئیات فیلم سوم: "Soul"

ساختار HTML اولیه و بخش `<head>` مشابه قبل است اما در بخش `<body>` ویدیوی فیلم با استفاده از پخش DASH پیاده شده‌است که به کمک کتابخانه Dash.js اجرا می‌شود.

صفحه جزئیات فیلم چهارم: "WALL·E"

پیاده سازی مشابه فیلم سوم است صرفاً اطلاعات تفاوت پیدا کرده‌اند.

صفحه جزئیات فیلم پنجم: "Up"

ساختار HTML اولیه و بخش <head> مشابه قبل است اما در بخش <body> ویدئوی فیلم با استفاده از پخش DASH با استفاده از کتابخانه Dash.js، که منبع آن از یک آدرس اینترنتی CDN است و از طریق بارگذاری فایل در سرور ابرآوان ایجاد شده است.

صفحه جزئیات فیلم ششم: "Big Hero 6"

ساختار HTML اولیه و بخش <head> مشابه قبل است اما در بخش <body> ویدئوی فیلم با استفاده از پخش HLS با استفاده از کتابخانه Hls.js، که منبع آن از یک آدرس اینترنتی CDN است و از طریق بارگذاری فایل در سرور ابرآوان ایجاد شده است.

به طور کلی می‌توان گفت این ساختار HTML به نحوی طراحی شده است تا اطلاعات هر فیلم را به صورت مجزا و سازمان‌یافته نمایش دهد. استفاده از قالب‌های Jinja2 برای تولید محتوای پویا و نمایش اطلاعات فیلم‌ها به صورت دینامیک بسیار کارآمد است. همچنین استفاده از فایل‌های CSS مجزا برای استایل‌دهی باعث می‌شود که کد HTML تمیز و خوانا باقی بماند. از نظر کارایی، استفاده از شبکه‌های CSS و تگ‌های HTML5 باعث می‌شود که صفحه‌ها به خوبی در دستگاه‌های مختلف نمایش داده شوند و تجربه کاربری بهتری فراهم شود. با این حال، باید دقت شود که تصاویر و ویدئوها بهینه‌سازی گردند تا زمان بارگذاری صفحات کاهش یابد و عملکرد کلی سایت بهتر شود. در فیلم سوم و چهارم استفاده از پخش DASH برای ویدئوها با استفاده از کتابخانه Dash.js بهبود کیفیت و سرعت بارگذاری ویدئوها را فراهم می‌آورد تا یک تجربه‌ی کاربری دینامیک و جذاب ارائه دهند و همچنین بهینه‌سازی برای سرعت بارگذاری و بهره‌وری سرور را فراهم کنند. همچنین در نمایش ویدئو چهارم و پنجم استفاده از CDN برای ذخیره و پخش فایل‌های ویدئویی منجر به بهبود سرعت بارگذاری و کیفیت ویدئو می‌شود و استفاده از کتابخانه‌های Dash.js و Hls.js جهت پشتیبانی از فرمت‌های پخش ویدئو است که اجازه می‌دهد تا ویدئوها با بهترین کیفیت ممکن و با استفاده از پروتکل‌های مناسب پخش شوند. این ساختارها از طریق استفاده از تکنولوژی‌های مدرن، بهینه‌سازی برای سرعت بارگذاری ویدئو و تجربه‌ی کاربری بهتر را فراهم می‌کنند.

توضیحات تکمیلی برای جمع‌بندی :

در کدهای آورده شده، از چهار روش مختلف برای استریم ویدئو استفاده شده است. هر یک از این روش‌ها دارای مزایا و معایب خاص خود هستند که در ادامه به توضیح و مقایسه آن‌ها پرداخته می‌شود:

روش Local : ویدئوها به صورت مستقیم از سرور محلی بارگذاری و پخش می‌شوند

مزایا: پیاده‌سازی ساده و سریع، عدم نیاز به تنظیمات و زیرساخت پیچیده، مناسب برای استفاده‌های کوچک و محلی.

معایب: محدودیت در پهنای باند و ظرفیت سرور محلی، عدم پشتیبانی از استریم تطبیقی (Adaptive Streaming)، کاهش کیفیت تجربه کاربری در صورت افزایش تعداد کاربران.

روش Local + DASH : ویدئوها به صورت تطبیقی از سرور محلی پخش می‌شوند و فایل‌های ویدئویی به قطعات کوچک‌تر تقسیم شده و به صورت تطبیقی با توجه به پهنای باند برای کاربر پخش می‌شوند.

مزایا: پشتیبانی از استریم تطبیقی، بهبود کیفیت تجربه کاربری، کاهش بار سرور با تقسیم ویدئو به قطعات کوچک‌تر، تناسب بیشتر برای شبکه‌ها با پهنای باند متغیر

معایب: نیاز به تنظیمات و زیرساخت پیچیده‌تر نسبت به روش Local، محدودیت در پهنای باند و ظرفیت سرور محلی.

روش CDN + DASH : ویدئوها از طریق شبکه توزیع محتوا (CDN) و به صورت تطبیقی پخش می‌شوند و فایل‌های ویدئویی در سرورهای مختلف در نقاط جغرافیایی مختلف توزیع می‌گردند.

مزایا: پشتیبانی از استریم تطبیقی، بهبود کیفیت تجربه کاربری، کاهش بار سرور محلی با استفاده از سرورهای توزیع شده، افزایش سرعت بارگذاری و پخش ویدئو با استفاده از سرورهای نزدیک به کاربر، مقیاس‌پذیری بالا و مناسب برای تعداد کاربران زیاد.

معایب: هزینه‌های بالا برای استفاده از CDN، نیاز به تنظیمات پیچیده‌تر نسبت به روش‌های محلی در صورت نیاز به پیاده‌سازی شبکه از صفر و عدم استفاده از شبکه‌های موجود.

روش CDN + HLS : ویدئوها از طریق شبکه توزیع محتوا (CDN) و با استفاده از پروتکل HLS (HTTP Live Streaming) پخش می‌شوند. HLS توسط اپل توسعه یافته و بسیار محبوب است و فایل‌های ویدئویی به قطعات کوچک‌تر تقسیم شده و به صورت تطبیقی با توجه به پهنای باند کاربر پخش می‌شوند.

مزایا: پشتیبانی از استریم تطبیقی، بهبود کیفیت تجربه کاربری، کاهش بار سرور محلی با استفاده از سرورهای توزیع شده، افزایش سرعت بارگذاری و پخش ویدئو با استفاده از سرورهای نزدیک به کاربر، مقیاس‌پذیری بالا و مناسب برای تعداد کاربران زیاد، پشتیبانی گسترده در دستگاه‌ها و مرورگرها.

معایب: هزینه‌های بالا برای استفاده از CDN و نیاز به تنظیمات پیچیده‌تر نسبت به روش‌های محلی.

مقایسه و برتری‌ها

روش Local برای پروژه‌های کوچک و محلی با تعداد کاربران کم مناسب است و برتری آن ساده بودن پیاده‌سازی و کم بودن هزینه است.

روش Local + DASH برای پروژه‌های متوسط که نیاز به استریم تطبیقی دارند اما بودجه و زیرساخت استفاده از CDN را ندارند، مناسب است و برتری آن بهبود کیفیت تجربه کاربری با استریم تطبیقی است.

روش CDN + DASH برای پروژه‌های بزرگ با تعداد کاربران زیاد که نیاز به کیفیت بالا و سرعت بارگذاری سریع دارند مناسب است و برتری آن در مقیاس‌پذیری بالا و بهبود تجربه کاربری است.

روش CDN + HLS برای پروژه‌های بزرگ با تعداد کاربران زیاد و تنوع دستگاه‌ها و مرورگرها مناسب است و برتری آن در پشتیبانی گسترده در دستگاه‌ها و مرورگرها، مقیاس‌پذیری بالا و بهبود تجربه کاربری است.

انتخاب روش مناسب برای استریم ویدئو بستگی به نیازها و محدودیت‌های پروژه دارد. برای پروژه‌های کوچک، روش Local ساده و کم‌هزینه است. اما برای پروژه‌های بزرگتر با نیاز به کیفیت و مقیاس‌پذیری، استفاده از CDN همراه با DASH یا HLS برتری دارد. در نهایت، انتخاب بین DASH و HLS بستگی به پشتیبانی دستگاه‌ها و نیازهای خاص پروژه دارد.

تفاوت DASH و HLS پیاده‌سازی و کاربرد

HLS (HTTP Live Streaming) و DASH (Dynamic Adaptive Streaming over HTTP) دو پروتکل محبوب برای استریم ویدئو هستند که با هدف ارائه تجربه کاربری بهتر از طریق پخش تطبیقی ویدئو طراحی شده‌اند. هر یک از این پروتکل‌ها ویژگی‌ها، مزایا و معایب خاص خود را دارند. در ادامه به بررسی دقیق تفاوت‌های این دو پروتکل از نظر پیاده‌سازی و کاربرد می‌پردازیم.

تفاوت‌های پیاده‌سازی:

الف) فرمت فایل و لیست پخش

DASH

- فایل DASH از فرمت‌های استاندارد MP4 استفاده می‌کند.
- لیست پخش فایل مانیفست (MPD - Media Presentation Description) است که شامل اطلاعات قطعات ویدئویی و نرخ بیت‌های مختلف می‌باشد.
- برای پیاده‌سازی DASH نیاز به ایجاد یک فایل MPD و تقسیم ویدئو به قطعات مختلف با نرخ بیت‌های متفاوت است که این فرآیند معمولاً با استفاده از ابزارهای خاص یا نرم‌افزارهای کدگذاری انجام می‌شود.

HLS

- فایل HLS از فرمت‌های TS (Transport Stream) استفاده می‌کند، هرچند که از HLS نسخه ۷ به بعد پشتیبانی از فرمت‌های فایلی دیگر مانند MP4 نیز افزوده شده است.
- لیست پخش فایل مانیفست (M3U8) است که شامل اطلاعات قطعات ویدئویی و نرخ بیت‌های مختلف است.
- برای پیاده‌سازی HLS نیاز به ایجاد یک فایل M3U8 و تقسیم ویدئو به قطعات مختلف با نرخ بیت‌های متفاوت است. این فرآیند نیز با استفاده از ابزارهای خاص یا نرم‌افزارهای کدگذاری انجام می‌شود.

ب) تطبیق‌پذیری و سازگاری

DASH

- بیشتر توسط مرورگرهای مدرن و در پلتفرم‌های مختلف پشتیبانی می‌شود، اما نیاز به کتابخانه‌های اضافی مانند dash.js برای پخش در مرورگرها دارد و به صورت بومی توسط دستگاه‌های اندرویدی و برخی تلویزیون‌های هوشمند پشتیبانی می‌شود.

HLS

- به صورت بومی توسط مرورگر سافاری و iOS پشتیبانی می‌شود و در مرورگرهای دیگر نیاز به کتابخانه‌های اضافی مانند hls.js دارد و توسط دستگاه‌های iOS (iPhone)، iPad و اپل تی‌وی پشتیبانی می‌شود.

(ج) امنیت و رمزگذاری

DASH

- از رمزگذاری‌های مختلف مانند AES-128 و DRM (Digital Rights Management) پشتیبانی می‌کند که DRM به خوبی با سیستم‌های DRM مانند Widevine و PlayReady سازگار است.

HLS

- از رمزگذاری AES-128 و همچنین DRM پشتیبانی می‌کند که با FairPlay (محصول اپل) سازگار است.

تفاوت‌های کاربرد:

الف) کیفیت و پایداری استریم

DASH

- به دلیل استفاده از فرمت MP4 می‌تواند کیفیت ویدئوی بسیار بالایی ارائه دهد و برای شرایط با پهنای باند متغیر مناسب است و با تغییر نرخ بیت تطبیقی کیفیت را تنظیم می‌کند.

HLS

- به دلیل استفاده از فرمت TS کیفیت ویدئوی خوبی ارائه می‌دهد، اما ممکن است کمی پایین‌تر از DASH باشد و برای شرایط با پهنای باند متغیر مناسب است و با تغییر نرخ بیت تطبیقی کیفیت را تنظیم می‌کند.

ب) کاربردها و موارد استفاده

DASH

- به دلیل پشتیبانی گسترده‌تر در مرورگرها و پلتفرم‌های مختلف برای پروژه‌هایی که نیاز به سازگاری گسترده دارند مناسب است و برای پروژه‌های دستگاه‌های اندروید و تلویزیون‌های هوشمند کاربردی است.

HLS

- به دلیل پشتیبانی بومی در دستگاه‌های iOS و اپل تی‌وی برای پروژه‌هایی که تمرکز اصلی آن‌ها روی کاربران دستگاه‌های اپل است مناسب است. علاوه بر آن به دلیل طراحی اولیه برای پخش زنده نیز بسیار مناسب است.

استفاده از CDN برای پخش ویدئو

CDN (Content Delivery Network) شبکه‌ای از سرورها است که در مکان‌های جغرافیایی مختلف توزیع شده‌اند. هدف اصلی CDN ارائه محتوای اینترنتی به کاربران با بالاترین سرعت و کمترین تاخیر ممکن است. با استفاده از سرورهای نزدیک به موقعیت جغرافیایی کاربر، زمان تاخیر کاهش یافته و سرعت بارگذاری و پخش ویدئو افزایش می‌یابد. همچنین با توزیع بار روی سرورهای مختلف، فشار را از روی سرور اصلی کاهش می‌دهد و باعث بهبود عملکرد کلی سیستم می‌شود. علاوه بر آن می‌تواند به طور همزمان تعداد زیادی از کاربران را پشتیبانی کند بدون اینکه کیفیت پخش کاهش یابد و می‌تواند ترافیک را به صورت هوشمند مدیریت کند و از ازدحام شبکه جلوگیری نماید. CDN می‌تواند به عنوان یک لایه حفاظتی عمل کند و حملات DDoS را مدیریت کرده و از سرور اصلی محافظت نماید. اما در کنار مزایای ذکر شده معایبی نیز دارد که در این خصوص می‌توان به موارد روبه‌رو اشاره نمود: استفاده از CDN معمولاً هزینه‌بر است و ممکن است برای پروژه‌های کوچک مقرون به صرفه نباشد. پیاده‌سازی و مدیریت CDN نیاز به دانش فنی دارد و ممکن است پیچیده باشد و برای بهره‌برداری بهینه از CDN نیاز به پیکربندی مناسب و مدیریت مداوم وجود دارد.