



دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

گزارش پروژه نهایی سیستم‌های چندرسانه‌ای

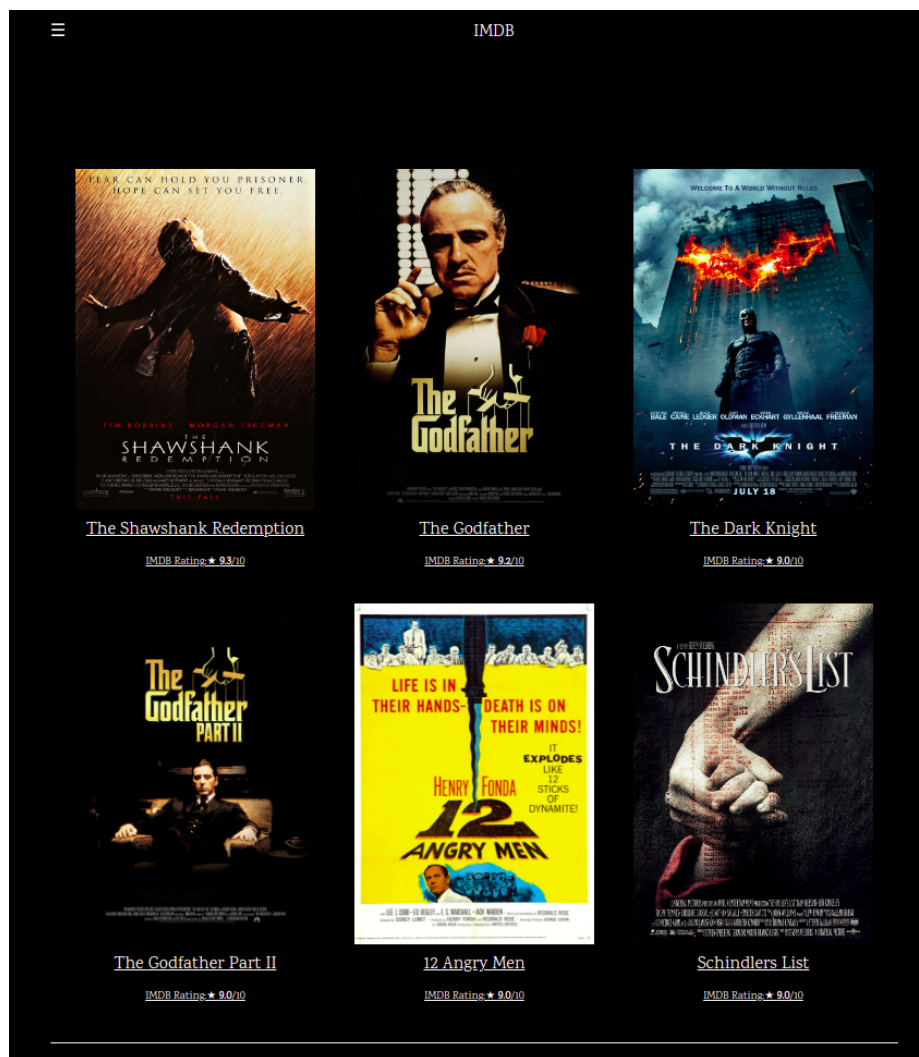
محمدعرفان قاسمی ۹۷۳۱۰۴۸

آریان بوکانی ۹۷۳۱۰۱۲


- ما از داکر برای پیاده‌سازی این پروژه استفاده نکردیم و ویدیو توضیحات پروژه در [این](#) لینک قرار دارد.

- سوالهای ۱ و ۲ به طور مفصل در ویدیو ذکر شده توضیح داده شده‌اند و در گزارش خلاصه از آن آورده شده است. برای بالا آمدن وبسایت از آدرس localhost:5000 استفاده کنید.


نمای کلی وبسایت در صفحه نخست به شکل زیر است.




نمای کلی برای هر فیلم‌ها نیز به شکل زیر می‌باشد.



IMDB






Description

Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.

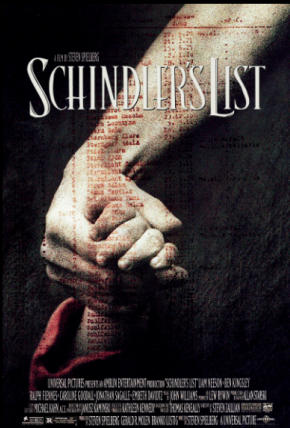
The Shawshank Redemption


Director: Frank Darabont

IMDB Rating: ★ 9.3/10



IMDB





Description

In German-occupied Poland during World War II, industrialist Oskar Schindler gradually becomes concerned for his Jewish workforce after witnessing their persecution by the Nazis.

Schindlers List

Director: Steven Spielberg

IMDB Rating: ★ 9.0/10

سوالات گزارش

۱. برای ایجاد Chunk هایی از ویدیو اصلی تریلر برای پهنای باندهای مختلف از کد dash.py استفاده کردیم.

```
app.py  homepage.html  movie.html  dash.py 1,2 X  schema.sql
dash.py > ...
1 import ffmpeg_streaming
2
3 # Generate DASH files for Movies
4
5 video = ffmpeg_streaming.input('/home/erfan/Projects/DASH/TheGodfather.mp4')
6
7 dash = video.dash(ffmpeg_streaming.Formats.h264())
8 dash.auto_generate_representations()
9 dash.output("/home/erfan/Projects/DASH/TheGodfather/TheGodfather.mpd")
10
```

در این کد همانطور که مشخص است از کتابخانه ffmpeg_streaming استفاده کردیم و بعد از آن با تابع Input فایل mp4 مرتبط با ویدیو مورد نظر را به آن پاس دادیم تا یک آبجکت از ویدیو مورد نظر در متغیر video داشته باشیم. در مرحله‌ی بعد با استفاده از Formats.h264 با تغییر فرمت فیلم به فرمت مطلوب برای ادامه‌ی کار آن را در متغیر dash ذخیره کرده و بعد از آن می‌توان برای کیفیت‌های مختلف مانند 144, 360, 480, ... به صورت دستی چانک‌ها را ایجاد کرد یا مانند کدی که ما استفاده کردیم از روش auto_generate استفاده کنیم. در آخر نیز فایل‌های تولید شده (chunks) و فایل mpd. مربوطه را در مسیر داده شده ذخیره می‌کنیم تا بتوان از آن برای پخش ویدیو در وبسایت ایجاد شده استفاده کرد.

۲.

برای توسعه سایت قسمت Front-End و Back-End یک جا و توسط Flask توسعه داده شده است. در قسمت Front-End از هیچ فریم‌ورک خاصی استفاده نشده است و تنها از کلاس‌های استایل‌دهی w3 استفاده شده است.

در قسمت Back-End از Flask برای توسعه‌ی API ها و همچنین پایگاه داده‌ی sqlite3 برای ذخیره‌ی اطلاعات ویدیوها استفاده شده است. (کدهای فرانت در بخش template قرار دارند که بسته به خروجی API تغییر می‌کنند)

```
1 from flask import Flask, render_template
2 import sqlite3
3
4 app = Flask(__name__)
5
6
7 def get_db_connection():
8     conn = sqlite3.connect('database.db')
9     conn.row_factory = sqlite3.Row
10    return conn
11
12
13 def get_movie(movie_id):
14     conn = get_db_connection()
15     post = conn.execute('SELECT * FROM movies WHERE id = ?',
16                         (movie_id,)).fetchone()
17     conn.close()
18     return post
19
20
21 @app.route('/', methods=['GET'])
22 def return_all_movies():
23     conn = get_db_connection()
24     movies = conn.execute('SELECT * FROM movies').fetchall()
25     conn.close()
26     return render_template('homepage.html', movies=movies)
27
28
29 @app.route('/movies/<movie_id>', methods=['GET'])
30 def return_specific_movies(movie_id):
31     movie = get_movie(movie_id)
32     return render_template('movie.html', movie=movie)
33
34
35 if __name__ == "__main__":
36     app.run(host='0.0.0.0', port=8000)
37
```

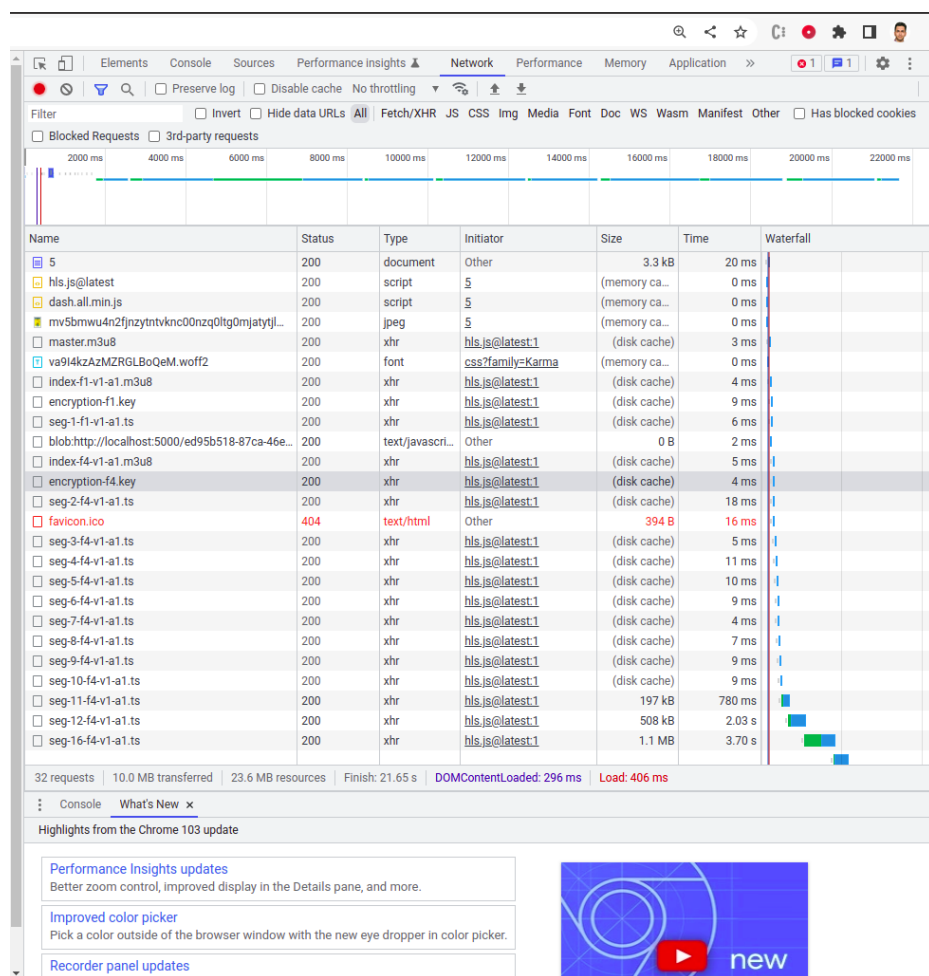
در قسمت Back-End برای ساختن پایگاه داده باید اسکریپت init_db.py را اجرا کرد. همچنین فایل schema.sql ساختار مدل استفاده شده را نشان داده است.

```
1 DROP TABLE IF EXISTS movies;
2
3 CREATE TABLE movies (
4     id INTEGER PRIMARY KEY AUTOINCREMENT,
5     title TEXT NOT NULL,
6     director TEXT NOT NULL,
7     rate TEXT NOT NULL,
8     movie_description TEXT NOT NULL,
9     trailer_url TEXT NOT NULL,
10    image_url TEXT NOT NULL
11 );
```

در بک‌اند همانطور که در ویدیو نیز توضیح داده شد ما از دو اندپوینت "/" و "<movies_id/<movies/" استفاده کردیم و در آن‌ها بعد از ارتباط با دیتابیس و خواندن دیتاهای مورد نیاز از دیتابیس آن‌ها را به صفحات html مرتبط با خود پاس داده و رندر کردیم.

۳.

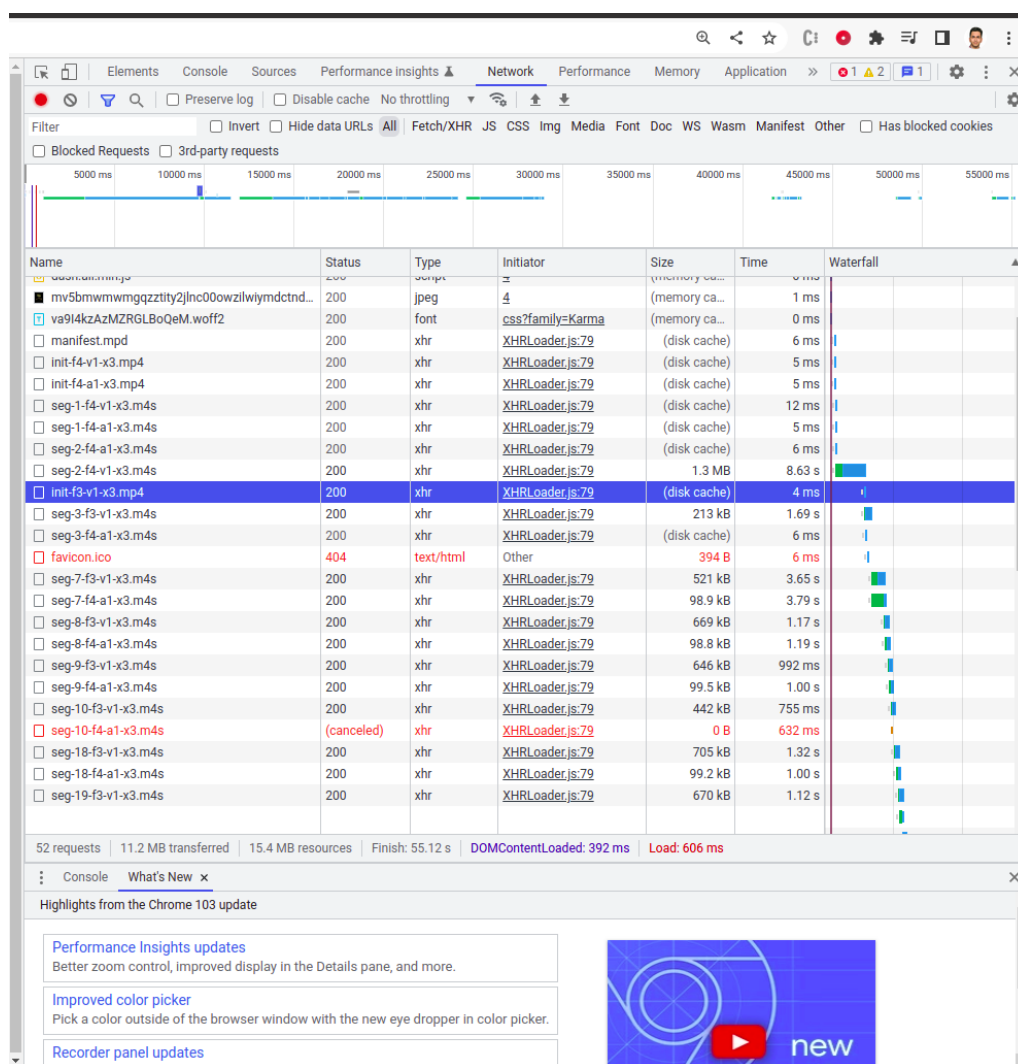
برای مثال از یکی از فیلم‌هایی که تریلر آن با استفاده از HLS پخش می‌شد استفاده کردیم و تب نتورک آن را مشاهده کردیم.



همانطور که مشخص است ابتدا هنگامی که صفحه را لود می‌کنیم فایل master.m3u8 دریافت می‌شود بعد از آن با توجه به پهنای باند کاربر فایلی تحت عنوان index-f1-v1-a1.m3u8 در شروع فرستاده می‌شود و بعد از آن segment های مرتبط با آن کیفیت ارسال می‌شود بعد از مدتی از شروع index-f4-v1-a1.m3u8 ارسال می‌شود که برای پهنای باندهای بیشتر و کیفیت بهتر است و این نشان می‌دهد با توجه پهنای باند اینترنت کاربر فایل های

m3u8 های متناسب با آن به کاربر ارسال می‌شوند و بعد از آن segment ها آن به کاربر برای پخش محتوای ویدیویی ارسال می‌شوند. در صورت تغییر پهنای باند مورد استفاده‌ی کاربر دوباره فایل index-f#-v1-a1.m3u8 برای پهنای باند کمتر برای کاربر ارسال می‌شود تا بتواند به استریم ویدیو ادامه دهد.

برای مثال از یکی از فیلم‌هایی که تریلر آن با استفاده از DASH پخش می‌شد استفاده کردیم و تب نتورک آن را مشاهده کردیم.



در شکل بالا که تب نتورک برای DASH است می‌توانیم ببینیم در ابتدا بعد از اینکه کاربر صفحه را لود کرد فایل manifest.mpd را دریافت می‌کند. بعد از آن با توجه به پهنای باند مورد نظر فایل‌های init_f#_v1_x3.m4s , ini_f#_a1_x3.m4s برای کاربر ارسال می‌شوند و بعد از تعیین کیفیت مورد نظر سگمنت‌های ویدیو که با فرمت m4s مشخص شده‌اند برای نمایش در کلاینت کاربر به آن ارسال می‌شوند. در صورت تغییر پهنای باند کاربر برای استریم ویدیو دوباره فایل‌های init_f#_v1_x3.m4s ,

ini_f#_a1_x3.mp4 برای پهنایباند کمتر به کاربر ارسال شده و سگمنتهای مختص آن به کاربر ارسال

می شوند.