



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده مهندسی برق و کامپیوتر



توسعه سیستم تحت وب جهت نمایش ویدیوهای YouTube به همراه زیرنویس فارسی با
استفاده از مترجم فرازین

پایان نامه برای دریافت درجه کارشناسی
در رشته مهندسی کامپیوتر گرایش سیستم‌های نرم افزاری

نام
علی مهرانی

شماره دانشجویی
810198542

استاد راهنما:
دکتر هشام فیلی

بهمن ماه 1402

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

تعهدنامه اصالت اثر

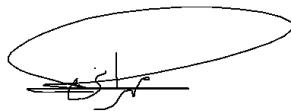
باسمه تعالی

اینجانب علی مهرانی تأیید می‌کنم که مطالب مندرج در این پایان‌نامه حاصل تلاش اینجانب است و به دستاوردهای پژوهشی دیگران که در این نوشته از آنها استفاده شده است مطابق مقررات ارجاع گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم سطح یا بالاتر ارائه نشده است. کلیه حقوق مادی و معنوی این اثر متعلق به دانشکده فنی دانشگاه تهران می‌باشد.

نام و نام خانوادگی دانشجو :

علی مهرانی

امضای دانشجو :



تشکر و قدردانی

با سپاس فراوان از استاد ارجمند جناب آقای دکتر فیلی، استاد راهنمای این پژوهش که از راهنمایی‌های مفید ایشان بهره‌مند گردیدم.

با سپاس فراوان از استاد ارجمند جناب آقای دکتر اسدپور، که زحمات نقد و داوری این پژوهش را با نهایت لطف پذیرفتند.

چکیده¹

YouTube به عنوان بزرگترین سیستم اشتراک ویدیو از مهم‌ترین پلتفرم‌های سرگرمی می‌باشد. کاربران در سراسر دنیا با اهداف مختلف در این سامانه اقدام به تماشای ویدیوهای در این سامانه مختلف می‌کنند. YouTube و پلتفرم‌های Streaming مانند آن نیز از طرفی برای ایجاد تجربه‌ای لذت‌بخش برای کاربران‌شان از روش‌های متفاوتی استفاده می‌کنند. از جمله این کارها می‌توان به مواردی مانند محلی‌سازی محتوا² اشاره کرد.

محلی‌سازی محتوا همواره مورد توجه بوده و برای رسیدن به آن نیز از روش‌های متفاوتی استفاده می‌شود که از جمله آن‌ها در مورد نمایش ویدیوها در YouTube می‌توان به ایجاد زیرنویس‌های مختلف برای زبان‌های جامعه هدف اشاره کرد. با وجود پیشرفت‌های صورت گرفته اما در کشور ما محدودیت‌های اعمال شده و همچنین فقدان محتوای محلی‌سازی شده به زبان فارسی برای بسیاری از ویدیوها در YouTube، وجود سیستمی که بتواند این کار را انجام دهد می‌تواند بسیار مفید واقع شود.

برای رسیدن به این هدف، محلی‌سازی محتوا از طریق ترجمه زیرنویس‌ها صورت خواهد گرفت. برای پیاده‌سازی این قابلیت ابتدا پس از آشنایی با نحوه کار کردن YouTube، روش‌های دریافت اطلاعات از آن را بررسی و در ادامه ساختار برنامه خود را مشخص می‌کنیم، سپس بر پایه مدل طراحی شده، برنامه تحت وب خود را با استفاده از فریم‌ورک‌های Django و React توسعه خواهیم داد. همچنین برای دریافت محتوای زیرنویس و ویدیو از YouTube نیز از کتابخانه‌های موجود در زبان Python استفاده خواهیم کرد.

کلمات کلیدی: زیرنویس، Web Application، YouTube، محلی‌سازی

¹ Abstract

² Content Localization

فهرست مطالب

فصل 1: مقدمه و بیان مساله	1
1-1- مقدمه	2
1-2- تاریخچه‌ای از موضوع تحقیق	2
1-3- شرح مسئله تحقیق	2
1-4- تعریف موضوع تحقیق	3
1-5- اهداف و آرمان‌های کلی تحقیق	3
1-6- روش انجام تحقیق	4
1-7- ساختار پایان‌نامه	4
فصل 2: مفاهیم اولیه و پیش زمینه پروژه	5
2-1- مقدمه	6
2-2- YouTube و Video Streaming	6
2-3- زیرنویس‌های ترجمه‌شده	7
2-4- مترجم فرازین	8
2-5- اطلاعات و پیش‌نیازهای پیاده‌سازی پروژه	8
2-5-1- دریافت اطلاعات از YouTube	8
2-5-1-2- Web Scraping	9
2-5-1-3- YouTube Data API	10
2-5-1-4- سیستم Invidious	11
2-5-2- کتابخانه Pytube	11
2-5-3- کتابخانه Youtube_transcript_api	11

12	2-6- خلاصه و جمع بندی
13	فصل 3: طراحی سیستم نمایش ویدیو و ایجاد زیرنویس فارسی
14	3-1- مقدمه
14	3-2- روش پیشنهادی برای طراحی رابط کاربری سیستم
16	3-3- ابزارهای مورد نیاز برای طراحی و توسعه سیستم
16	3-3-1- کتابخانه React
19	3-3-2- فریم‌ورک Bootstrap
22	3-3-3- فریم‌ورک Django
24	3-3-4- استفاده از YouTube Data API
25	3-3-5- استفاده از Invidious API
27	3-4- معیار ارزیابی
29	3-5- نتایج بدست آمده از طراحی رابط کاربری و نمایش اولیه ویدیو
33	3-6- تحلیل نتایج
34	3-7- خلاصه و جمع بندی
35	فصل 4: پیاده سازی سیستم دریافت اطلاعات و نمایش ویدیو و ایجاد زیرنویس فارسی
36	4-1- مقدمه
36	4-2- نحوه پیاده سازی
36	4-3- قسمت Client یا Front-end
41	4-4- قسمت Server یا Back-end
42	4-4-1- بخش Stream در سرور
44	4-4-2- بخش Captions در سرور

45بخش Invidious در سرور
45خلاصه و جمع‌بندی
46فصل 5: جمع‌بندی، نتیجه‌گیری و پیشنهادها
475-1- جمع‌بندی
475-2- نتیجه‌گیری
475-2-1- نوآوری / دستاوردها
475-2-2- محدودیتها
485-2-3- پیشنهادها
49فصل 6: مراجع
51پیوست‌ها

فهرست شکل‌ها

- شکل (1-2) نمونه از لینک اصلی یک ویدیو YouTube 7
- شکل (2-2) فرآیند کلی استخراج اطلاعات ویدیو از YouTube 9
- شکل (1-3) ساختار یک نمونه Component در React 18
- شکل (2-3) استفاده از State Value در React 19
- شکل (3-3) صفحه اصلی طراحی شده برنامه 21
- شکل (4-3) نمونه‌ای از خروجی YouTube Data API برای ویدیو با شناسه Oe421EPjeBE 24
- شکل (5-3) نمونه‌ای از خروجی Invidious API برای ویدیو با شناسه PtQiiknWUCI 26
- شکل (6-3) پیش‌نمایش اطلاعات ویدیو با شناسه MNeX4EGtR5Y 29
- شکل (7-3) صفحه مخصوص نمایش ویدیو با شناسه MNeX4EGtR5Y 30
- شکل (8-3) لیست فرمت‌های دانلود ویدیو با شناسه MNeX4EGtR5Y 30
- شکل (9-3) قابلیت دانلود زیرنویس ویدیو 31
- شکل (10-3) نمایش در حالت آزمایشی ویدیو با شناسه MNeX4EGtR5Y قسمت اول 31
- شکل (11-3) نمایش در حالت آزمایشی ویدیو با شناسه MNeX4EGtR5Y قسمت دوم 32
- شکل (12-3) اشتراک‌گذاری Embed ویدیو 32
- شکل (13-3) نتایج جستجو ویدیو YouTube در برنامه 33
- شکل (1-4) ساختار برنامه React ایجادشده در تابع App 37
- شکل (2-4) ساختار Component با نام Layout 38
- شکل (3-4) دریافت اطلاعات ویدیو از Invidious با استفاده از Axios 39
- شکل (4-4) ساختار VideoPlayer برای نمایش ویدیو در حالت عادی (نیازمند به VPN) 40
- شکل (5-4) ساختار VideoPlayer برای نمایش ویدیو در حالت آزمایشی (بدون نیاز به VPN) 41
- شکل (6-4) دریافت اطلاعات Stream ویدیو مورد نظر در YouTube 42
- شکل (7-4) دریافت و انتقال اطلاعات Thumbnail ویدیو 43
- شکل (8-4) دریافت اطلاعات Stream ویدیو مورد نظر در YouTube 44
- شکل (9-4) فایل urls.py در برنامه Stream در سرور 44
- شکل (10-4) دریافت اطلاعات از Invidious در سمت سرور 45

فهرست جدول‌ها

فهرست علائم اختصاری

DOM	Document Object Model
API	Application Programming Interface
SPA	Single-page Application
MPA	Multi-page Application
NPM	Node Package Manager
HTTP	Hypertext Transfer Protocol

فصل 1:

مقدمه و بیان مساله

در این فصل نخست به بیان توضیحات ابتدایی درباره video streaming و محلی سازی media در بستر اینترنت می پردازیم. سپس درباره مساله خود و نحوه ایجاد سیستم مورد نظر اشارات مختصری میکنیم. در نهایت درباره ساختار این پایان نامه نیز توضیحاتی داده خواهد شد.

1-1- مقدمه

امروزه YouTube به عنوان محبوب‌ترین سرویس اشتراک ویدیو در دنیا معرفی می‌شود و محبوبیت آن نیز روز به روز در حال افزایش است. تنوع محتوا در این سامانه نیز در کنار محبوبیت آن همواره رو به افزایش بوده و کاربران تنها با یک جستجوی ساده می‌توانند هزاران ویدیو مرتبط با علاقه‌شان تماشا کنند. محتوای YouTube نیز دارای تنوع زبانی بسیاری می‌باشد و این سامانه برای استفاده مفید کاربران، امکان قرارداد و نمایش زیرنویس‌های متعدد را نیز فراهم می‌کند.

یکی از علاقه‌مندی‌های کاربران این سامانه، مشاهده ویدیوهای مورد علاقه‌شان به زیرنویس زبان محلی می‌باشد که در خصوص زیرنویس با زبان فارسی، خلا این خدمات به شدت محسوس می‌باشد. همچنین بدلیل سیاست‌های کنترلی، کلیه فایل‌های ویدیویی این سایت به صورت غیر هوشمند، دچار محدودیت و غیر قابل دسترسی گردیده است. از این رو ایجاد سیستمی که بتواند محدودیت‌های ذکر شده در این سامانه مشهور را برای کاربران فارسی زبان برطرف کند، می‌تواند بسیار مفید واقع گردد.

1-2- تاریخچه‌ای از موضوع تحقیق

سیستم‌های اشتراک ویدیو یا Media/Video Streaming همواره از محبوبیت خاصی برخوردار بوده‌اند، از مشهورترین این سامانه‌ها می‌توان به YouTube و یا Netflix اشاره کرد. در میان این سیستم‌ها YouTube امروزه محبوب‌ترین گزینه کاربران می‌باشد و بسیاری از این سیستم‌ها از جمله YouTube به دنبال راه‌هایی برای محلی‌سازی محتوای‌شان بوده‌اند.

1-3- شرح مسئله تحقیق

استفاده از سیستمی مانند YouTube می‌تواند برای کاربران محدودیت‌هایی ایجاد کند که از جمله آن‌ها نیاز کاربران به مشاهده ویدیوها در زبان محلی‌شان می‌باشد، همچنین امکان دانلود ویدیو از جمله دیگر

ویژگی‌هایی است که در این سامانه امکان انجام آن بدون پرداخت هزینه و یا استفاده از نرم‌افزارهای خارجی یا THIRD PARTY وجود ندارد.

امروزه امکان مشاهده محتوا در بستر اینترنت برای کاربران در سراسر دنیا به زبان محلی، مورد توجه است و ایجاد این امکان همواره چالش مهمی برای شرکت‌های بزرگ بوده است.

4-1- تعریف موضوع تحقیق

برای افزایش کارایی پلتفرم‌های Streaming و به جهت کمک به افرادی که امکان مشاهده محتوا در زبان اصلی آن را ندارند تا به حال تلاش‌هایی صورت گرفته است. از جمله این تلاش‌ها میتوان به ویژگی‌های زیر در این برنامه‌ها اشاره کرد:

- 1- ایجاد امکان قراردادن زیرنویس به زبان‌های متنوع برای محتوا
 - 2- ارائه محتوا به صورت دوبله شده (که البته برای همه ویدیوها ممکن نیست)
 - 3- ترجمه اتوماتیک که با استفاده از مدل‌های یادگیری ماشین ایجاد می‌شود
- البته هرکدام از روش‌های بالا دارای محدودیت‌هایی هستند اما یکی از روش‌های ساده‌ای که میتوان برای حل این مشکل به کار برد، ترجمه مستقیم یک زیرنویس از زبان اصلی به زبان مورد نظر می‌باشد. که موضوع تحقیق ما نیز بررسی و اجرای این امکان می‌باشد.

4-1- اهداف و آرمان‌های کلی تحقیق

با توجه به محدودیت‌های YouTube که در قسمت‌های قبل ذکر شدند، هدف ما ایجاد سامانه‌ای است که بتواند این محدودیت‌ها را برطرف کند و محیط مناسبی برای کاربران فارسی زبان ایجاد کند. اهداف این سامانه به شرح زیر می‌باشد:

- 1- ارائه مدلی برای نمایش ویدیوهای YouTube به همراه زیرنویس فارسی که در مدل ارائه شده، محدودیت‌های اعمالی دولت‌ها، کاهش هزینه کاربر نهایی، اقتصادی بودن مدل و قابل اطمینان بودن آن در نظر گرفته خواهد شد.
- 2- توسعه یک Web Application بر مبنای مدل ارائه شده.

6-1- روش انجام تحقیق

با توجه به علاقه‌مندی‌های کاربران و اهداف پروژه، اقدام به طراحی مدل مناسب برای آن خواهیم کرد و سیستم مورد نظر را بر پایه مدل ارائه شده، توسعه خواهیم داد. برای دریافت اطلاعات مورد نظر از YouTube کتابخانه‌های مختلفی در زبان‌های برنامه‌نویسی مختلف وجود دارد که مناسب‌ترین آن‌ها کتابخانه‌های Pytube و Youtube_transcript_api در زبان Python می‌باشد که به ترتیب برای دریافت اطلاعات نمایش ویدیو و دریافت اطلاعات زیرنویس ویدیوها از آن‌ها استفاده خواهیم کرد.

همچنین با توجه به زبان برنامه‌نویسی انتخاب شده، از فریم‌ورک Django برای ایجاد یک Server استفاده خواهیم کرد و برای ایجاد رابط کاربری سیستم نیز از کتابخانه React استفاده خواهد شد.

7-1- ساختار پایان‌نامه

در فصل دوم، به مفاهیم و پیش‌نیازهای اولیه پروژه از جمله تعاریف مربوط به Video Streaming و نحوه دریافت اطلاعات از YouTube و کتابخانه‌های مورد استفاده خواهیم پرداخت.

فصل سوم در برگیرنده‌ی توضیح مربوط به طراحی ساختار و معماری پیشنهادی برنامه و نحوه پیاده‌سازی سیستم می‌باشد.

در فصل چهارم در مورد ابزارهای استفاده شده برای پیاده‌سازی و جزئیات مربوط به آن و همچنین معیار ارزیابی نتایج بدست آمده صحبت خواهیم کرد. همچنین این فصل ارائه دهنده‌ی نتایج اجرای سیستم خواهد بود. در این فصل علاوه بر ارائه و تحلیل نتایج، در مورد ویژگی‌های سیستم ایجادشده صحبت خواهیم کرد.

در نهایت، در فصل پنجم، نتیجه‌گیری‌های کلی حاصل شده در این تحقیق، پیاده‌سازی‌ها و محدودیت‌ها مورد بحث قرار می‌گیرد و پیشنهادهایی برای ادامه‌ی مسیر به علاقمندان این حوزه‌ی ارائه خواهد شد.

فصل 2: مفاهیم اولیه و پیش زمینه پروژه

در فصل پیش رو مقدمات، مفاهیم اولیه و پیش زمینه‌هایی را که جهت درک هر چه بهتر موضوع-های مطرح شده در این پایان‌نامه مورد نیاز است، از مفاهیم مربوط به Video Streaming تا روش‌های استخراج اطلاعات از YouTube ارائه خواهد شد.

1-2- مقدمه

در این فصل به طور خلاصه، ابتدا به مفاهیم و پیش‌نیازهایی که برای درک و اجرای این پروژه الزامی هستند اشاره می‌کنیم و سپس با توجه به ماهیت مسئله خود، به روش‌های استخراج اطلاعات از YouTube به جهت پیاده‌سازی پروژه می‌پردازیم و به تعدادی از کتابخانه‌های مناسب برای این کار در زبان Python اشاره می‌کنیم. همچنین درباره مترجم فرازین که مترجم مورد استفاده ما برای ترجمه زیرنویس ویدیو می‌باشد، توضیحاتی ارائه می‌دهیم.

2-2- Video Streaming و YouTube

امروزه YouTube محبوب‌ترین سیستم اشتراک ویدیو دنیاست و در این حوزه یکی از مهم‌ترین چالش‌ها نمایش ویدیو و محتوا در بستر اینترنت با بهترین کیفیت ممکن و از طرفی به حداقل رساندن اختلالات نمایش ویدیو بدلیل بروز مشکلات در شبکه می‌باشد تا بتواند تجربه‌ای لذت بخش و راحتی برای کاربر هنگام مشاهده محتوا در اینترنت می‌باشد. برای رسیدن به این هدف YouTube از تکنیکی با نام Adaptive Biterate Streaming استفاده می‌کند که به کمک آن کیفیت ویدیو در لحظه بسته به وضعیت اینترنت کاربر تغییر می‌کند و نرخ انتقال اطلاعات به مقدار مناسبی تنظیم می‌شود تا حتی‌الامکان باعث توقف نمایش نشود.

برای رسیدن به این شرایط، YouTube ویدیوهای خود را در تمام کیفیت‌هایی که وجود دارند به بخش‌های مختلف تقسیم می‌کند و برای هرکدام از آن‌ها نیز یک URL تولید می‌کند که این URL در دامنه googlevideo.com می‌باشد و بسیار پیچیده و طولانی‌ست. سپس همزمان با نمایش ویدیو برای کاربر تشخیص می‌دهد که آیا نمایش ویدیو برای کاربر به صورت روان و Smooth در حال صورت گرفتن است یا خیر. در صورتی که کاربر در حال تجربه قطع و وصلی ویدیو بود، قسمت بعدی ویدیو از URL متناظر با کیفیت پایین‌تر دریافت می‌شود تا نمایش روان‌تر و بهتری را برای کاربر رقم زند و حداقل Interruption را داشته باشد. [9][10]

شکل (2-1) نمونه‌ای از لینک اصلی ویدیو در دامنه googlevideo را نشان می‌دهد که ترکیبی از اطلاعات ویدیو و فرمت و کیفیت و موارد دیگری می‌باشد که توسط الگوریتم پیچیده‌ای تولید می‌شود.

```
https://rr4---sn-4g5e6nsr.googlevideo.com/videoplayback?
expire=1706797771&ei=a1a7ZefoIou0sfIPovq7wAg&ip=181.215.182.60&id=o-
AFEmf4C10NwZKZsQ1Msvo8jkS-MoOy-
5afrnz4PuYRPz&itag=18&source=youtube&requiressl=yes&xpc=EgVo2aDSNQ%3D%
3D&vprv=1&svpuc=1&mime=video%2Fmp4&cnr=14&ratebypass=yes&dur=24641.480
&lt=1701251567064642&fexp=24007246&c=ANDROID&txp=5438434&sparams=expir
e%2Ce%2Cip%2Cid%2Citag%2Csource%2Crequiresl%2Cxpc%2Cvprv%2Csvpuc%2C
mime%2Ccnr%2Cratebypass%2Cdur%2Clt&sig=AJfQdSswRgIhAOr0-
20gtxeT_T46SLG6tRKfghrKhxGD-KsremwpAj8VAiEaiHcxFxs-NF816-
9FN7bykRe53y27CAer1T_2jqaIwdY%3D&title=Node.js%20Full%20Course%20for%2
0Beginners%20%7C%20Complete%20All-in-
One%20Tutorial%20%7C%207%20Hours&redirect_counter=1&cm2rm=sn-
q4fez17l&req_id=ff67d953245da3ee&cms_redirect=yes&cmsv=e&mh=0j&mip=95.
179.223.161&mm=34&mn=sn-
4g5e6nsr&ms=ltu&mt=1706777270&mv=u&mvi=4&pl=23&lsparams=mh,mip,mm,mn,m
s,mv,mvi,pl&lsig=AA05W4owRAIgMWButvlee5ksuYJDurcFxa-
BI0eVx42w0i3HTEhQYMgCIHhbWr1J9eQ3sIgZqt1BQ5JpotCWsJia4sGFwkwMkzN
```

شکل (2-1) نمونه از لینک اصلی یک ویدیو YouTube.

2-3- زیرنویس‌های ترجمه‌شده

برای محلی‌سازی محتوا با ایجاد زیرنویس با زبان‌های مختلف، روش‌های متفاوتی وجود دارد. سیستم YouTube که مورد تحقیق ماست از تکنولوژی بازشناسی گفتار¹ و الگوریتم‌های یادگیری ماشین برای ایجاد اتوماتیک زیرنویس به زبان‌های مختلف برای ویدیوها استفاده می‌کند. [1][2]

در این پروژه، هدف ما نیز ایجاد زیرنویس با زبان فارسی برای ویدیوها می‌باشد که روش انجام آن نیز، ترجمه زیرنویس زبان انگلیسی ویدیو به فارسی با حفظ ساختار فایل زیرنویس می‌باشد.

برای فرآیند ترجمه نیز از سامانه مترجم فرازین استفاده می‌کنیم.

¹ Speech Recognition

4-2- مترجم فرازین

مترجم فرازین یک سیستم ترجمه هوشمند است که پروژه آن از سال 1389 در آزمایشگاه پردازش متن و زبان‌های طبیعی دانشگاه تهران آغاز شد. این مترجم از یک موتور ترجمه نرونی استفاده می‌کند و خدمات متنوعی از جمله ترجمه آنلاین کتاب، سایت، مقاله و ترجمه فایل‌های متنوعی از جمله فایل‌های Word، PDF، Powerpoint و Text ارائه می‌دهد. همچنین یکی دیگر از خدمات اصلی آن که در این پروژه مورد توجه ما است، قابلیت ترجمه فایل‌های زیرنویس از انگلیسی به فارسی و بالعکس می‌باشد. [3]

این ویژگی‌ها، فرازین را به انتخابی مناسب جهت اجرای عملیات ترجمه زیرنویس تبدیل می‌کند.

5-2- اطلاعات و پیش‌نیازهای پیاده‌سازی پروژه

در ادامه این بخش به پیش‌نیازهای پیاده‌سازی پروژه می‌پردازیم. در ابتدا به دریافت و استخراج اطلاعات از YouTube و روش‌های انجام آن می‌پردازیم و سپس به کتابخانه‌های کاربردی در زبان Python برای انجام این پروژه اشاره می‌کنیم.

1-5-2- دریافت اطلاعات از YouTube

برای پیاده‌سازی پروژه خود باید اطلاعات یک ویدیو YouTube را دریافت کنیم که دریافت این اطلاعات با استفاده از کد یا ID آن ویدیو صورت می‌گیرد. در یک دیدگاه کلی باید بتوانیم با دادن ID یک ویدیو اطلاعات آن نظیر نام آن، تاریخ آپلود، زمان و موارد مشابه را دریافت کنیم.

شکل (2-2) یک تصویر کلی از این فرآیند را به ما نشان می‌دهد.

با توجه به این شکل، در ادامه روش‌هایی که می‌توان با کمک آن‌ها چنین سیستمی را پیاده‌سازی کرد، بررسی می‌کنیم.



شکل (2-2) فرآیند کلی استخراج اطلاعات ویدیو از YouTube

Web Scraping -2-5-1-2

Web Scraping فرآیند استخراج اطلاعات از صفحات وب می باشد که این اطلاعات استخراج شده می تواند در قالب هایی مانند متن یا تصویر باشد. نرم افزارهای Web Scraper برای استخراج اتوماتیک اطلاعات از صفحات وب ایجاد می شوند و در موارد متعددی کاربرد دارند که چند نمونه از آن ها به شرح زیر می باشند:

1. **مقایسه اطلاعات:** scraper ها میتوانند اطلاعاتی مانند قیمت کالاها را از انواع مختلفی از وبسایت ها استخراج کنند و آن ها را با هم مقایسه کنند.
2. **پایش رسانه ها و خبرگزاری ها:** از Web Scraper ها می توان برای استخراج و جمع آوری اطلاعات از وبسایت های خبری استفاده کرد.
3. **پیش بینی اطلاعات:** از این نرم افزارها می توان برای استخراج اطلاعات در گذر زمان استفاده کرد و با تحلیل این اطلاعات، پیش بینی های مناسب انجام داد.

فرآیند Scraping به طور کلی ابتدا با فرستادن یک درخواست به آدرس مورد نظر و دریافت¹ محتوا یا اطلاعات صفحه آغاز می شود و سپس با استفاده ابزارهای مناسب اطلاعات کلیدی از آن استخراج می شود که این استخراج معمولاً از روش های متفاوتی صورت می گیرد که به دو نمونه معروف آن ها اشاره می کنیم:

¹ Fetch

1. **HTML Parsing**: با دادن درخواست و دریافت اطلاعات یک صفحه، محتوای render شده قالب HTML آن صفحه در دسترس ما قرار می‌گیرد، بدیهی‌ست که بسیاری از اطلاعات کلیدی در قالب HTML قابل مشاهده هستند و با استفاده از تکنیک HTML Parsing می‌توان این اطلاعات را از صفحه استخراج و ذخیره کرد. این تکنیک به ما قابلیت دسترسی به قسمت مورد نظر خود در قالب HTML را می‌دهد.

2. **DOM Parsing**: با استفاده از Web Scraping علاوه بر قالب HTML می‌توان به مدل شی‌گرا سند¹ یک صفحه نیز دسترسی داشت، این مدل که تحت عنوان DOM شناخته می‌شود، ساختار یک Document یا همان صفحه در وب را در قالب یک ساختار درخت از اشیا (Objects) نشان می‌دهد که هر Object متناظر با یک قسمت از Document می‌باشد مانند یک Text یا Element در صفحه. با تکنیک‌های مختلف تحت عنوان DOM Parsing می‌توان اطلاعات مورد نیاز را از آن استخراج کرد.

برای این که بتوانیم از یک ویدیو YouTube اطلاعات مورد نظر را استخراج کنیم، یک راه می‌تواند استفاده از تکنیک‌های Web Scraping و دریافت اطلاعات صفحه متناظر آن ویدیو در YouTube باشد.

YouTube Data API -2-5-1-3

استخراج اطلاعات ویدیو از YouTube با استفاده از Web Scraping می‌تواند پیچیده و زمان‌بر باشد اما خود YouTube روشی را برای دریافت اطلاعات از آن تحت عنوان YouTube Data API در اختیار عموم قرار داده است.

با استفاده از واسط برنامه‌نویسی کاربردی² YouTube می‌توانیم اطلاعات Entity های متنوعی از YouTube مانند ویدیوها، کانال‌ها، playlistها را دریافت کنیم. این API ها در قالب یک سیستم Restful می‌باشند و برای استفاده از آنها کافی‌ست یک Key یا کلید API دریافت کنیم و با فراخوانی نسخه شماره 3 API مورد نظر، اطلاعات لازمه را دریافت کنیم.

محتوای Documentation این API برای عموم قابل مشاهده می‌باشد. [4]

¹ Document Object Model

² Application Programming Interface

2-5-1-4 Invidious سیستم

سیستم Invidious که خود را به عنوان یک Front-end جایگزین برای YouTube معرفی میکند، نرم‌افزاری است که در آن می‌توان ویدیوهای YouTube را نمایش کرد و از بسیاری از قابلیت‌های YouTube در آن استفاده کرد. این سیستم با استفاده از تکنیک‌های Web Scraping اطلاعات موجود در YouTube را دریافت کرده و نمایش می‌دهد. [6][7] این سیستم از سوی YouTube نیز بدلیل نقض سیاست‌های آن اخطار دریافت کرده است. [8]

نرم‌افزار Invidious بر روی سرورهای مختلفی که از آن‌ها تحت عنوان “Instance” نام برده میشود، قرار داده شده‌اند و در صورت بروز مشکل در یک سرور می‌توان از Instance دیگری استفاده کرد. [6][7]

این سیستم همچنین API‌هایی برای استفاده از آن در اختیار قرار داده که می‌توان با استفاده از این API‌ها اطلاعات لازمه را دریافت کرد. [6]

2-5-2-2 کتابخانه Pytube

کتابخانه Pytube یک کتابخانه متن باز¹ در زبان Python می‌باشد که برای دانلود کردن ویدیوهای YouTube ساخته شده است. با استفاده از این کتابخانه می‌توان اطلاعات فرمت‌های مختلف یک ویدیو و URL آن‌ها در سرور اصلی googlevideo پیدا کرد. با استفاده از قابلیت‌های این کتابخانه امکان دانلود و نمایش ویدیو برای ما محیا می‌شود. لازم به ذکر است که بدلیل محدودیت‌های اعمال شده به YouTube، استفاده از این کتابخانه تنها در صورت داشتن VPN امکان‌پذیر می‌باشد.

2-5-3-2 کتابخانه Youtube_transcript_api

کتابخانه Youtube_transcript_api یک کتابخانه متن باز دیگر در زبان Python است که با کمک آن می‌توان زیرنویس‌های یک ویدیو در YouTube را دریافت کرد و همچنین امکان دریافت زیرنویس‌های ترجمه اتوماتیک YouTube نیز با این کتابخانه وجود دارد. از دیگر قابلیت‌های آن می‌توان به امکان

¹ Open Source

تبدیل کردن محتوای زیرنویس به فرمت‌های مختلف آن از جمله SRT و VTT اشاره کرد. با کمک این کتابخانه می‌توانیم اطلاعات زیرنویس ویدیو مورد نظر خود را دریافت و به فرمت لازم تبدیل کنیم. لازم به ذکر است که بدلیل محدودیت‌های اعمال شده به YouTube، استفاده از این کتابخانه تنها در صورت داشتن VPN امکان‌پذیر می‌باشد.

6-2- خلاصه و جمع بندی

در این فصل با مفاهیم اولیه و پیش‌زمینه‌هایی جهت درک هرچه بهتر ساختار YouTube و نحوه استخراج اطلاعات از آن آشنا شدیم. در ادامه از این مفاهیم جهت طراحی و پیاده‌سازی سیستم مدنظر استفاده خواهیم کرد.

فصل 3: طراحی سیستم نمایش ویدیو و ایجاد

زیرنویس فارسی

فصل سوم در برگرفته‌ی توضیح مربوط به ساختار و طراحی سیستم نمایش ویدیو و ایجاد زیرنویس فارسی و پیاده‌سازی آن می‌باشد.

1-3- مقدمه

در این فصل نخست به معرفی ساختار کلی سیستم و طراحی آن می‌پردازیم و در ادامه درباره بخش‌های مختلف آن و تکنولوژی‌های استفاده‌شده و جزئیات پیاده‌سازی آن توضیحاتی ارائه می‌دهیم. در هر بخش تصاویری از نتایج پیاده‌سازی صورت‌گرفته در آن بخش ارائه داده می‌شود.

2-3- روش پیشنهادی برای طراحی رابط کاربری سیستم

پس از بررسی‌های صورت‌گرفته و آشنایی با پیش‌نیازهای لازم اقدام به طراحی ساختار کلی سیستم و رابط کاربری آن می‌نماییم

در بخش اول طراحی ابتدا می‌خواهیم سیستمی را پیاده‌سازی کنیم که در صفحه اصلی آن مکانی برای قرار دادن لینک ویدئو موجود می‌باشد و کاربر می‌تواند لینک ویدئو مورد نظر خود را در آنجا وارد کند. سیستم با دریافت لینک یک ویدئو در یوتیوب اطلاعات کلی آن را به کاربر نمایش می‌دهد از جمله این اطلاعات می‌توان به مواردی مانند نام ویدئو، مدت زمان آن، تاریخ آپلود آن، تعداد View‌های آن، توضیحات آن و همچنین زبان‌هایی که برای آنها زیرنویس موجود است، اشاره کرد.

پس از دریافت این اطلاعات توسط سیستم آنها در قالب یک Dialog Box به کاربر نمایش داده می‌شوند و از کاربر پرسیده می‌شود که آیا این اطلاعات منطبق با ویدئو مد نظر اوست یا خیر.

در ادامه در صورت تایید کاربر، او به صفحه‌ای هدایت خواهد شد که مخصوص نمایش ویدئو مد نظر اوست در این صفحه ویدئو برای کاربر نمایش داده می‌شود و علاوه بر آن تمامی اطلاعات ویدئو که در قبل ذکر شد نیز قابل مشاهده است همچنین در این صفحه امکاناتی نیز برای کاربر فراهم می‌باشد. این امکانات شامل قابلیت دانلود ویدئو با کیفیت و فرمت مد نظر و همچنین قابلیت دانلود زیرنویس در زبان‌هایی که برای آنها زیرنویس ایجاد شده و یا ترجمه فارسی فرازین، می‌باشد.

در صورتی که کاربر درخواست دانلود ویدئو را دهد، لیستی از فرمت‌های ویدئو جهت انتخاب او برای دانلود نمایش داده می‌شود و در صورتی که درخواست دانلود زیرنویس فرستاده شود، لیست تمامی زیرنویس‌های موجود به او نمایش داده خواهد شد.

همچنین می‌خواهیم قابلیت نمایش ویدئو به صورت embed در صفحات دیگر در بستر اینترنت نیز فراهم باشد که برای اضافه کردن این قابلیت، صفحه‌ای تحت عنوان نمایش ویدئو به صورت embed طراحی و اضافه خواهیم کرد.

در ادامه می‌خواهیم قابلیت جستجو در YouTube در برنامه ما برای کاربر فراهم باشد، بسیار بهتر است که کاربر بتواند به جای انتخاب و کپی کردن لینک یک ویدئو از YouTube و قرار دادن آن در برنامه ما، بتواند مستقیماً در برنامه ما جستجو انجام دهد و نتایج متناظر آن جستجو در YouTube، به او نمایش داده شود. در برنامه ما این قابلیت باید برای کاربر فراهم باشد.

برای نمایش ویدئو می‌خواهیم لینک اصلی ویدئو YouTube در دامنه googlevideo را پیدا کنیم و پس از یافتنش، آن را در یک video player نمایش دهیم.

با توجه به موارد بالا الگو پیاده‌سازی سیستم مشخص می‌شود اما درباره موضوع نمایش ویدئو باید به این مورد توجه کنیم که دسترسی به دامنه googlevideo.com مانند YouTube دچار محدودیت گردیده است و به این دلیل با وجود فرستادن لینک ویدئو به قسمت client برنامه، امکان مشاهده ویدئو برای کاربر در صورت نداشتن VPN فراهم نمی‌باشد. برای حل این مشکل، علاوه بر این نوع نمایش ویدئو، می‌خواهیم نوع دیگری از نمایش را داشته باشیم که در آن به جای اینکه آدرس ویدئو برای کاربر یعنی بخش client فرستاده شود، سرور اطلاعات ویدئو را بخش به بخش دریافت کند و محتوای binary این اطلاعات را برای بخش client بفرستد تا در این صورت، تنها با فعال‌سازی یک vpn در سمت سرور، امکان مشاهده ویدئو برای کاربر حتی در صورت عدم داشتن vpn نیز وجود داشته باشد و سرور بتواند از کتابخانه‌های ذکر شده در قسمت قبل نیز استفاده کند.

برای پیاده‌سازی این نوع سیستم نمایش، باید لینک نمایش ویدئو در سرور به لینک اصلی ویدئو YouTube در googlevideo به گونه‌ای “map” شده باشد تا سرور به عنوان واسطه‌ای بین قسمت client و دامنه googlevideo قرار بگیرد و مسئول دریافت و انتقال اطلاعات باشد.

همچنین لینک Thumbnail ویدئوهای YouTube که در دامنه yting.com می‌باشد نیز دچار محدودیت دسترسی قرار گرفته است که برای حل این مشکل نیز می‌توانیم از سرور برای دریافت و ارسال اطلاعات تصویر مانند مورد قبلی استفاده کنیم اما در هر دو حالت باید توجه کنیم که معمولاً دریافت محتوای رسانه‌ای مانند ویدئو و تصویر از سمت Client صورت می‌گیرد و چون بدلیل برخی محدودیت‌ها می‌خواهیم این فرآیند در سمت سرور صورت گیرد، باید حتماً ارسال اطلاعات به صورت بخش بخش در قسمت‌های کوچک باشد چون در غیر اینصورت، با افزایش تعداد کاربران و درخواست‌ها به سرور، امکان بروز Bottleneck در سرور وجود خواهد داشت.

در ادامه درباره ابزارهای/تکنولوژی‌های مورد نیاز برای طراحی و توسعه این سیستم توضیحاتی ارائه خواهیم داد.

3-3- ابزارهای مورد نیاز برای طراحی و توسعه سیستم

در این بخش با ابزارهای مورد نیاز که از آن‌ها برای طراحی و توسعه سیستم استفاده خواهیم کرد، آشنا خواهیم شد.

1-3-3- کتابخانه React

React یک کتابخانه Open-source در زبان JavaScript برای توسعه رابط‌های کاربری^۱ تعاملی می باشد که توسط شرکت Facebook در سال ۲۰۱۳ ایجاد شده است. این کتابخانه از معماری برنامه تک صفحه‌ای^۲ پشتیبانی می کند که به معنای این است که برنامه توسعه داده شده با آن، در یک صفحه Web در مرورگر اجرا می شود و و تعامل آن با Server برای تبادل داده به وسیله Javascript و بدون نیاز به Reload شدن صفحه صورت می گیرد که باعث می شود کاربر بسیار تجربه راحت تری در کارکردن با یک سایت داشته باشد.

در معماری SPA برخلاف معماری برنامه‌های چند صفحه‌ای^۳، عوض شدن url باعث نمی شود که یک صفحه جدید دوباره از ابتدا اقدام به Load شدن در مرورگر کند، بلکه در این معماری تنها یک صفحه وجود دارد و تغییر url مانند یک "Trigger" عمل می کند که باعث می شود صفحه‌ای که نمایش داده شده بود، به گونه‌ای "پاک" شود و محتوای متناظر جدید در آن نمایش داده شود، که این فرایند توسط هسته اصلی برنامه SPA که در اینجا React می باشد صورت می گیرد.

React در هنگام توسعه به عنوان یک برنامه/سرور جدا که می تواند با سرورهای دیگر ارتباط برقرار کند، می باشد و از این رو برای اجرا شدن به یک محیط اجرایی^۴ سمت سرور برای زبان Javascript نیاز دارد که این محیط به Node JS معروف می باشد. بنابراین برای استفاده از React ابتدا لازم است که Node JS را نصب کرده باشیم.

¹ User Interface

² Single Page Application

³ Multi Page Application

⁴ Runtime Environment

پس از نصب کردن Node با استفاده از دستور زیر یک برنامه جدید در React ایجاد می‌کنیم.

```
npx create-react-app my-react-app
```

سپس به Directory پروژه می‌رویم

```
cd my-react-app
```

در نهایت با دستور زیر برنامه خود در React را اجرا می‌کنیم

```
npm start
```

Node JS نیز مانند عمده محیط‌های توسعه سرور، یک سیستم مدیریت پکیج دارد که این سیستم NPM نام دارد و از آن برای نصب کردن Package های متفاوت در محیط Node استفاده می‌شود.

پس از وارد کردن دستور بالا، برنامه React به عنوان یک سرور جدا به صورت پیش فرض در پورت شماره 3000 ماشین خود یا همان میزبان محلی¹، شروع به کار می‌کند.

در React، Component ها اساسی ترین بخش از ساختار توسعه می‌باشند. طراحی React بر مبنای ساختار مبتنی بر Component (Component-Based Architecture) است که به توسعه دهندگان این امکان را می‌دهد که واحدهای کوچکتری از کد را باز استفاده کرده و مدیریت کنند. در واقع می‌توانیم بگوییم که در React ساختار صفحات Render شده از بخش‌هایی تحت عنوان Component تشکیل شده‌اند.

یک Component این قابلیت را دارد که یک قالب html را نمایش یا Render کند

شکل شماره ساختار یک Component در React را نمایش می‌دهد که در بخش Return آن، قالب html ای که قرار است Render شود، مشخص می‌شود.

¹ Localhost

```

component.jsx > ...
1  export default function SampleReactComponent() {
2      return (
3          <div>
4              <p>Sample React Component</p>
5          </div>
6      );
7  }

```

شکل (3-1) ساختار یک نمونه Component در React.

Component ها در React میتوانند شامل متغیرهایی تحت عنوان State نیز باشند.

State یک مفهوم مهم در React است که به Component ها این امکان را می‌دهد که وضعیت داخلی خود را مدیریت کنند و با تغییر آن، Render مجدد صفحه انجام شود بدون اینکه نیاز باشد کل صفحه مجدداً از نو Reload شود. در React بسیار بهتر است که طراحی Component ها به گونه‌ای باشد که از آنها بتوان در قسمت‌ها و صفحات مختلف سایت استفاده کرد.

کدهای React در قالب syntax ای تحت عنوان 'JSX' نوشته می‌شوند. JSX یک extension برای syntax زبان Javascript می‌باشد که امکان ایجاد درخت‌های DOM در کد Javascript را به ما می‌دهد این Syntax توسط Facebook ارائه شده و شامل ویژگی‌هایی مانند امکان داشتن نمایش شرطی^۲ با کمک blockهای شرطی و همچنین امکان استفاده از expression ها داخل element های html می‌باشد.

با استفاده از ساختار JSX و همچنین مفهوم State در React می‌توانیم نمایش صفحات خود را بر پایه منطق مورد نظر خود کنترل کنیم که این کار مشابه استفاده و تغییر مقادیر DOM در صفحات وب می‌باشد. شکل (3-2) یک نمونه Counter پیاده‌سازی شده در React با مقدار اولیه 0 را نشان می‌دهد که مقدار آن با هر بار فشردن دکمه، افزایش می‌یابد. در نهایت نیز، کد JSX به معادل خود در Javascript تبدیل خواهد شد.

¹ JavaScript XML

² Conditional Rendering

```

1  import React, { useState } from "react";
2
3  const Counter = () => {
4    const [count, setCount] = useState(0);
5
6    const handleIncrement = () => {
7      setCount(count + 1);
8    };
9
10   return (
11     <div>
12       <p>Count: {count}</p>
13       <button onClick={handleIncrement}>Increment</button>
14     </div>
15   );
16 };
17
18 export default Counter;

```

شکل (2-3) استفاده از State Value در React.

React برای مدیریت State از مفهومی به نام DOM مجازی یا Virtual DOM استفاده می‌کند.

Virtual DOM یک مفهوم مهم در React است که برای بهبود عملکرد و بهینه‌سازی Rendering در برنامه‌های React استفاده می‌شود. زمانی که تغییری در وضعیت یک Component رخ می‌دهد، React ابتدا یک نسخه مجازی از DOM تحت عنوان Virtual DOM ایجاد می‌کند و سپس با مقایسه آن با نسخه قبلی Virtual DOM، تغییرات را پیدا کرده و مجموعه‌ای از عملیات (مثل افزودن یا حذف عناصر) را برای به‌روزرسانی DOM واقعی یا Real DOM تولید می‌کند.

3-3-2- فریم‌ورک Bootstrap

برای طراحی رابط کاربری برنامه خود و طراحی Style‌های آن از فریم‌ورک Bootstrap استفاده خواهیم کرد. Bootstrap یک کتابخانه open-source و یک فریم‌ورک CSS برای طراحی وب است که توسط توسعه‌دهندگان Twitter ایجاد شده است. این فریم‌ورک ابزارها و قالب‌های HTML، CSS و JavaScript را برای طراحی و توسعه وبسایت‌ها و برنامه‌های وب به صورت آماده ارائه می‌دهد. امروزه Bootstrap یکی از

محبوب‌ترین و پراستفاده‌ترین فریم‌ورک‌های طراحی UI می‌باشد که ویژگی‌های زیر از جمله دلایل محبوبیت آن می‌باشد:

1. **طراحی Responsive:** Bootstrap از طراحی Responsive پشتیبانی می‌کند که به وبسایت یا برنامه‌ی شما این امکان را می‌دهد که در اندازه‌های مختلف صفحات به صورت مناسب نمایش داده شود، برخلاف طراحی adaptive که در آن باید برای هر دسته اندازه صفحه، باید یک طراحی جداگانه صورت بگیرد.

2. **سیستم Grid:** Bootstrap ابزارها و کلاس‌های آماده برای مدیریت سیستم گرید (Grid System) را فراهم می‌کند که به طراحی و چینش مناسب در صفحات کمک می‌کند و از عوامل بسیار موثر در طراحی Responsive می‌باشد.

3. **Component های آماده:** Bootstrap دارای اجزای طراحی شده مانند Navbar، فرم‌های ورودی، دکمه‌ها، جداول، و غیره است که با استفاده از کلاس‌های CSS آماده قابل استفاده‌اند و باعث صرفه‌جویی بسیار در زمان می‌شوند و دیگر لازم نیست که تمام اجزای صفحه از اول ایجاد و Style شوند.

4. **افزونه‌های¹ Javascript:** Bootstrap افزونه‌های Javascript آماده‌ای دارد که امکانات اضافی را برای وبسایت شما اضافه می‌کنند، مانند Modal box، Slideshow، تقویم‌ها و موارد مشابه.

5. **Customizable بودن:** با وجود امکانات زیاد، Bootstrap به شما اجازه می‌دهد که قسمت‌ها و Style‌های مورد نظر خود را تغییر دهید و بتوانید Component مورد نظر خود را Customize کنید.

برای استفاده از Bootstrap در React می‌توانیم با استفاده از NPM آن را در محیط پروژه خود نصب کنیم:

```
npm install bootstrap
```

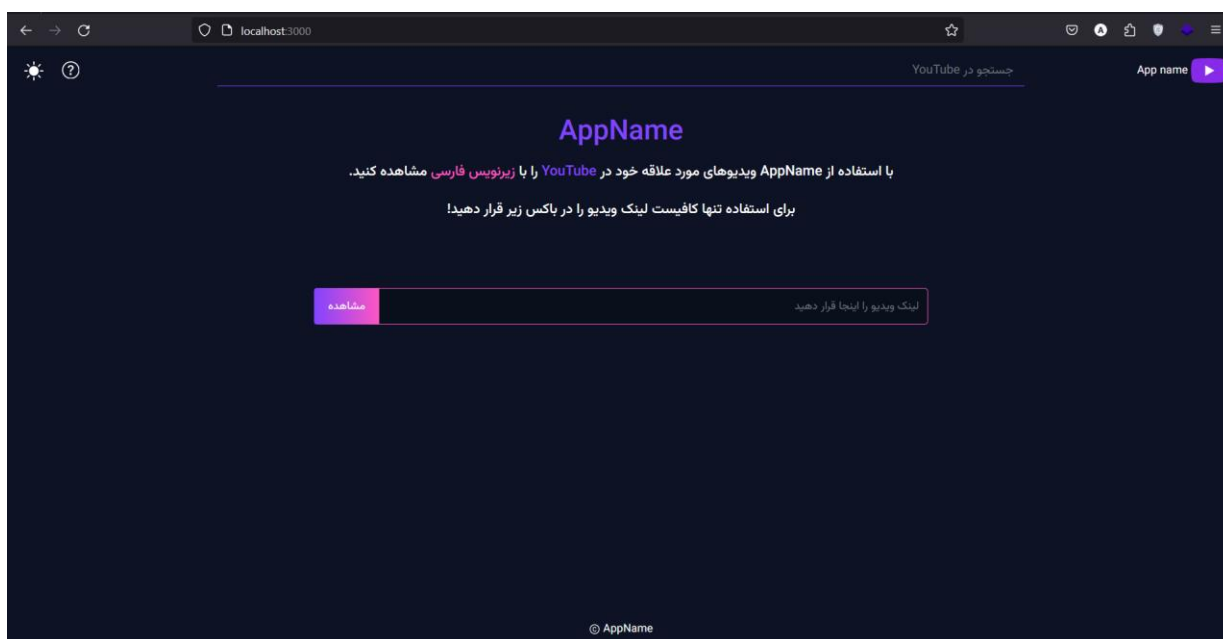
¹ Plugin

همچنین Package ای تحت عنوان React-Bootstrap نیز وجود دارد که بسیاری از Component های آن را در قالب Component های React در اختیار ما قرار می دهد و با استفاده از دستور زیر آن را نصب می کنیم:

```
npm install react-bootstrap
```

پس از آشنایی و استفاده از React و Bootstrap اقدام به ساختن رابط کاربری برنامه خود می کنیم.

شکل شماره (3-3) صفحه اصلی برنامه ما در React را نشان می دهند. همانگونه که از تصویر نیز مشخص است در صفحه اصلی فرد لینک ویدئو مورد نظر خود را می تواند قرار می دهد. برای بهبود تجربه کاربری نیز مواردی از جمله امکان انتخاب بین تم تیره و روشن پیاده سازی شده است.



شکل (3-3) صفحه اصلی طراحی شده برنامه.

در ادامه درباره نحوه پردازش لینک و دریافت اطلاعات توضیحاتی ارائه خواهد شد.

3-3-3 فریم‌ورک Django

با توجه به این که کتابخانه‌های مورد استفاده در این پروژه در زبان Python می‌باشند، لازم است بتوانیم یک سرور برای زبان Python داشته باشیم تا بتوانیم از امکانات این کتابخانه با فرستادن درخواست های HTTP استفاده کنیم.

برای انجام این کار از فریم‌ورک Django که یک فریم‌ورک برای ایجاد برنامه‌های تحت Web می‌باشد که در سال 2005 توسعه داده شده است و امروزه توسط بسیاری از شرکت‌ها و برنامه‌ها استفاده می‌شود.

Django بر اساس معماری Model-View-Template (MVT)، ساخته شده و ابزارها و امکانات بسیاری را برای توسعه وبسایت‌های پویا و مقیاس‌پذیر فراهم می‌کند. چند مورد از ویژگی‌ها و مزایای Django به شرح زیر می‌باشند:

1. **وجود ORM (Object-Relational Mapping):** Django از ORM استفاده می‌کند که به توسعه‌دهندگان این امکان را می‌دهد تا با پایگاه داده‌ها به صورت شی‌گرا برخورد کنند و نیازی به نوشتن کوئری‌های SQL نداشته باشند.

2. **داشتن بخش Admin:** Django از یک رابط مدیریت به صورت پیش‌فرض به نام Django Admin برخوردار است که به توسعه‌دهندگان این امکان را می‌دهد تا بدون نوشتن کد، پنل مدیریت اولیه‌ای داشته باشند.

3. **ابزارهای Authentication و Authorization:** Django ابزارهای مختلف حریم خصوصی و احراز هویت کاربران را ارائه می‌دهد که می‌تواند به راحتی در پروژه‌های مختلف استفاده شود.

4. **داشتن ویژگی‌های امنیتی:** Django از ویژگی‌های امنیتی مانند پیشگیری از حملات CSRF، SQL Injection و حفاظت از نحوه ذخیره و دریافت پسورد استفاده می‌کند.

5. **سیستم URL Routing:** سیستم Routing در Django این امکان را می‌دهد که URL‌ها به توابع مشخص در کد مرتبط شوند.

6. داشتن **Template Engine**: Django از یک موتور قالب ساده و قوی برای تولید و ارائه HTML پویا استفاده می‌کند. البته با توجه به این‌که برنامه client ما از React استفاده خواهد کرد، از امکانات این بخش استفاده نخواهیم کرد.

7. **معماری MVT**: Django بر اساس معماری (Model-View-Template) MVT که تغییری اندک نسبت به معماری MVC دارد، ساخته شده است. در این معماری، مدل‌ها (Model) برای پایگاه داده و اطلاعات مدل‌سازی، ویوها (View) برای پردازش اطلاعات و تولید محتوا (معادل controller در MVC) برای نمایش به کاربر و Template برای نمایش داده‌ها می‌باشد.

در محیط Django سیستم مدیریت package ها بر عهده pipenv می‌باشد که با کمک آن می‌توان در محیط توسعه Django، Package های لازمه را دریافت و نصب کرد. با توجه به ماهیت پروژه ما و الگو پیاده‌سازی آن، package های زیر را در ابتدا در محیط توسعه، نصب خواهیم کرد.

1. **Pytube**: کتابخانه‌ای که در بخش‌های قبل به آن اشاره شد و برای دانلود ویدیوهای YouTube از آن استفاده می‌شود.

2. **Youtube_transcript_api**: کتابخانه‌ای که در بخش‌های قبل به آن اشاره شد و برای دریافت اطلاعات زیرنویس ویدیوهای YouTube از آن استفاده می‌شود.

3. **Django_rest_framework**: package ای که از آن برای ساختن API ها در Django در قالب REST-API استفاده می‌شود.

4. **Pillow**: کتابخانه‌ای که از آن برای پردازش تصویر استفاده می‌شود. در این پروژه کاربرد اصلی آن دریافت اطلاعات Thumbnail یک ویدیو و انتقال آن می‌باشد.

معمولاً توصیه می‌شود هنگام استفاده از Django برنامه به کوچکترین قسمت‌های ممکن تقسیم شود و هرکدام از این قسمت‌ها به عنوان یک “app” در Django ایجاد شود به طور مثال در یک فروشگاه آنلاین، مواردی مانند “Users” یا “Products” می‌توانند به عنوان یک app ایجاد شوند. هر app شامل بخش‌های مربوط به Model و View و Template می‌باشد.

در بخش‌های بعدی در رابطه با جزئیات ساختار برنامه خود در Django توضیحاتی ارائه خواهیم داد.

3-3-4- استفاده از YouTube Data API

با توجه به توضیحات داده شده در قسمت های قبل درباره استفاده از YouTube Data API برای دریافت اطلاعات از YouTube امکان استفاده از آن در برنامه خود و ویژگی های آن را بررسی خواهیم کرد.

شکل (3-4) یک نمونه خروجی این API برای دریافت اطلاعات یک ویدئو می باشد.

```
{kind: 'youtube#videoListResponse', etag: 'P5JDja4K5q9mXqbf_ck5_QH3JQI', items: Array(1), pageInfo: {...}}
  etag: "P5JDja4K5q9mXqbf_ck5_QH3JQI"
  items: Array(1)
    0:
      etag: "fMIQYnDvzeczRZLuJAWhFJddeUeo"
      id: "Oe421EPjeBE"
      kind: "youtube#video"
      snippet:
        categoryId: "27"
        channelId: "UC8butISFWt-W17EV0hUK0BQ"
        channelTitle: "freeCodeCamp.org"
        defaultAudioLanguage: "en"
        defaultLanguage: "en"
        description: "Learn how to use Node and Express in this comprehensive course. First, you will learn the fundam
        liveBroadcastContent: "none"
      localized:
        description: "Learn how to use Node and Express in this comprehensive course. First, you will learn the fund.
        title: "Node.js and Express.js - Full Course"
        [[Prototype]]: Object
        publishedAt: "2021-04-01T12:37:58Z"
      thumbnails:
        default: {url: 'https://i.ytimg.com/vi/Oe421EPjeBE/default.jpg', width: 120, height: 90}
        high: {url: 'https://i.ytimg.com/vi/Oe421EPjeBE/hqdefault.jpg', width: 480, height: 360}
        maxres: {url: 'https://i.ytimg.com/vi/Oe421EPjeBE/maxresdefault.jpg', width: 1280, height: 720}
        medium: {url: 'https://i.ytimg.com/vi/Oe421EPjeBE/mqdefault.jpg', width: 320, height: 180}
        standard: {url: 'https://i.ytimg.com/vi/Oe421EPjeBE/sddefault.jpg', width: 640, height: 480}
        [[Prototype]]: Object
        title: "Node.js and Express.js - Full Course"
        [[Prototype]]: Object
        [[Prototype]]: Object
      length: 1
      [[Prototype]]: Array(0)
      kind: "youtube#videoListResponse"
      pageInfo:
        resultsPerPage: 1
        totalResults: 1
        [[Prototype]]: Object
        [[Prototype]]: Object
```

شکل (3-4) نمونه ای از خروجی YouTube Data API برای ویدئو با شناسه Oe421EPjeBE

مطابق شکل بالا با استفاده از این API اطلاعات ویدئو از جمله نام، توضیحات، مدت، تاریخ آپلود و موارد مشابه را می توان مشاهده کرد، همچنین می توان با تنظیم مناسب پارامترهای API مطابق با Documentation ارائه شده، لیست زبان هایی که در آن ها برای ویدئو زیرنویس موجود است را دریافت کرد.

برخلاف YouTube که دسترسی به آن دچار محدودیت شده، استفاده از API های YouTube محدودیت دسترسی ندارد، همچنین همواره پاسخگو و قابل اعتماد است برخلاف کتابخانه هایی که بر پایه Web Scraping

کار می‌کنند (ایجاد تغییر در قالب ساختار HTML یا DOM در سایت YouTube ممکن است باعث بروز خطا در فرآیند دریافت اطلاعات توسط Web Scraper ها شود).

با وجود این امکانات اما در استفاده از این سیستم، محدودیت‌هایی نیز موجود است که به شرح زیر آمده است:

1. **میزان درخواست محدود:** مطابق با توضیحات ارائه شده در Documentation این API ها، میزان درخواستی که در طول 24 ساعت می‌توان با یک API Key فرستاد، محدود است و در صورت افزایش این مقدار از سطح مشخص شده امکان استفاده از آن تا زمان شارژ دوباره آن وجود ندارد. این محدودیت تحت عنوان “Quota” مطرح می‌شود. [5]

2. **عدم دریافت محتوای زیرنویس‌ها:** مطابق با توضیحات ارائه شده در Documentation این API ها، امکان دریافت محتوای زیرنویس ویدئوها تنها در دو حالت ممکن می‌باشد. برای دریافت محتوای زیرنویس یک ویدئو یا باید آپلودکننده آن این اجازه را در اختیار YouTube قرار داده باشد و اینکه یا کاربر باید با استفاده از API احراز هویت انجام دهد و پس از آن تنها محتوای زیرنویس ویدئوهای آپلودشده توسط خودش را ببیند که با وجود این دو استثنا نیز امکان دریافت زیرنویس بسیاری از ویدئوها وجود ندارد.

3. **عدم دریافت لینک ویدئو:** مطابق با سیاست‌های YouTube امکان دریافت لینک اصلی ویدئو (در دامنه googlevideo) وجود ندارد و امکان دانلود نیز از این رو در این حالت محیا نیست.

با توجه به محدودیت‌های ذکرشده، استفاده از API های YouTube بخصوص درباره محدودیت Quota، در صورت استفاده از این سرویس، ممکن است با محدودیت‌هایی هنگام کار کردن با برنامه مواجه شویم.

در ادامه امکان استفاده از سیستم Invidious را بررسی می‌کنیم و سرویس مورد نظر خود را جهت استفاده مشخص می‌کنیم

3-3-5- استفاده از Invidious API

در بخش قبل به سیستم Invidious و قابلیت‌های آن اشاره کردیم. در ادامه امکان استفاده از api های آن و ویژگی‌های آن را بررسی خواهیم کرد.

API های Invidious نیز مانند API های YouTube قابل دسترس هستند و دچار محدودیت دسترسی

نیستند، برخلاف YouTube Data API اما API های این سیستم محدودیت استفاده ندارند و می توان همواره از آن ها استفاده کرد، البته با توجه به سیاست های YouTube ممکن است دسترسی به برخی Instance های Invidious توسط Google، Block شود که در این حالت نیز می توان با تغییر Instance ای که از API آن استفاده می کنیم، بتوانیم دوباره به اطلاعات دسترسی پیدا کنیم. این سرویس همچنین اطلاعات کامل زیرنویس ها برای دریافت آن ها را نیز در اختیار ما قرار می دهد. شکل شماره (3-5) نمونه ای از خروجی این API می باشد:

```

1 {
2   "type": "video",
3   "title": "Python Django 7 Hour Course",
4   "videoId": "PtQiiknWUcI",
5   "videoThumbnails": [ ...
60 ],
61 "storyboards": [ ...
95 ],
96 "description": "Build a discord-like application with Python Django. V
97 "descriptionHtml": "Build a discord-like application with Python Django
98 "published": 1632873600,
99 "publishedText": "2 years ago",
100 "keywords": ["django", "python", "python django", "django tutorial"],
101 "viewCount": 1487533,
102 "likeCount": 0,
103 "dislikeCount": 0,
104 "paid": false,
105 "premium": false,
106 "isFamilyFriendly": true,
107 "allowedRegions": [ ...
357 ],
358 "genre": "Science & Technology",
359 "genreUrl": "/channel/",
360 "author": "Traversy Media",
361 "authorId": "UC29ju8bIPH5as80GnQzwJyA",
362 "authorUrl": "/channel/UC29ju8bIPH5as80GnQzwJyA",
363 "authorVerified": true,
364 "authorThumbnails": [ ...
395 ],
396 "subCountText": "2.19M",
397 "lengthSeconds": 25805,
398 "allowRatings": true,
399 "rating": 0,
400 "isListed": true,
401 "liveNow": false,
402 "isUpcoming": false,
403 "dashUrl": "https://iv.ggtyler.dev/api/manifest/dash/id/PtQiiknWUcI",
404 "adaptiveFormats": [ ...
883 ],
884 "formatStreams": [ ...
909 ],
910 "captions": [ ...
916 ],
917 "recommendedVideos": [ ...
2238 ]
2239 }
2240

```

شکل (3-5) نمونه ای از خروجی Invidious API برای ویدیو با شناسه PtQiiknWUcI

با توجه به ویژگی‌های هر دو سرویس YouTube Data API و Invidious API و با توجه به مزایا، معایب و محدودیت‌های هر کدام، تصمیم به استفاده از سیستم Invidious API به جهت دریافت اطلاعات از YouTube گرفتیم.

در ادامه درباره نتیجه پیاده‌سازی پروژه و معیار ارزیابی آن توضیحاتی ارائه خواهیم داد.

3-4- معیار ارزیابی^۱

1. اولین معیار ارزیابی ما این است که کاربر باید بتواند ویدئو مد نظر خود را پس از وارد کردن لینک آن مشاهده کند تمامی اطلاعات ویدئو نیز باید برای کاربر قابل مشاهده باشد لازم به ذکر است که در این قسمت فرض می‌شود کاربر در حال استفاده از VPN می‌باشد البته استفاده از VPN تنها باید برای ایجاد امکان مشاهده ویدئو باشد و بقیه اطلاعات ویدئو در صورت عدم فعال بودن VPN کاربر نیز باید قابل مشاهده باشند بدیهی است اطلاعات ویدئو نمایش داده شده باید متناظر با لینک وارد شده باشد.

2. دومین معیار ارزیابی ما این است که قبل از آماده شدن ویدئو برای مشاهده توسط کاربر، زیرنویس فارسی ترجمه فرازین باید آماده شده باشد و در هنگام پخش ویدئو باید مشخص باشد. بدیهی است هماهنگی میان زیرنویس و ویدئو و کیفیت مناسب آن از ملزومات این بخش می‌باشد.

3. سومین معیار ارزیابی ما امکان دانلود کردن ویدئو به فرمت مورد نظر می‌باشد. هنگامی که کاربر درخواست دانلود ویدئو می‌دهد باید تمامی فرمت‌های موجود ویدئو جهت دانلود به او نمایش داده شود و در صورت انتخاب یکی از این فرمت‌ها، فایل مورد نظر باید آماده دانلود شود. بدیهی است مدیریت خطاهای احتمالی مانند مشکلات برقراری ارتباط با سرور در حین انجام این عملیات، از ملزومات این بخش می‌باشد.

4. چهارمین معیار ارزیابی ما امکان دانلود کردن زیرنویس می‌باشد هنگامی که کاربر درخواست دانلود زیرنویس را می‌دهد باید تمامی زبان‌هایی که زیرنویس برای آن ارائه شده است به همراه نسخه ترجمه شده فارسی فرازین آماده دانلود باشد. بدیهی است مدیریت خطاهای احتمالی مانند مشکلات برقراری ارتباط با سرور در حین انجام این عملیات، از ملزومات این بخش می‌باشد.

¹ Evaluation metric

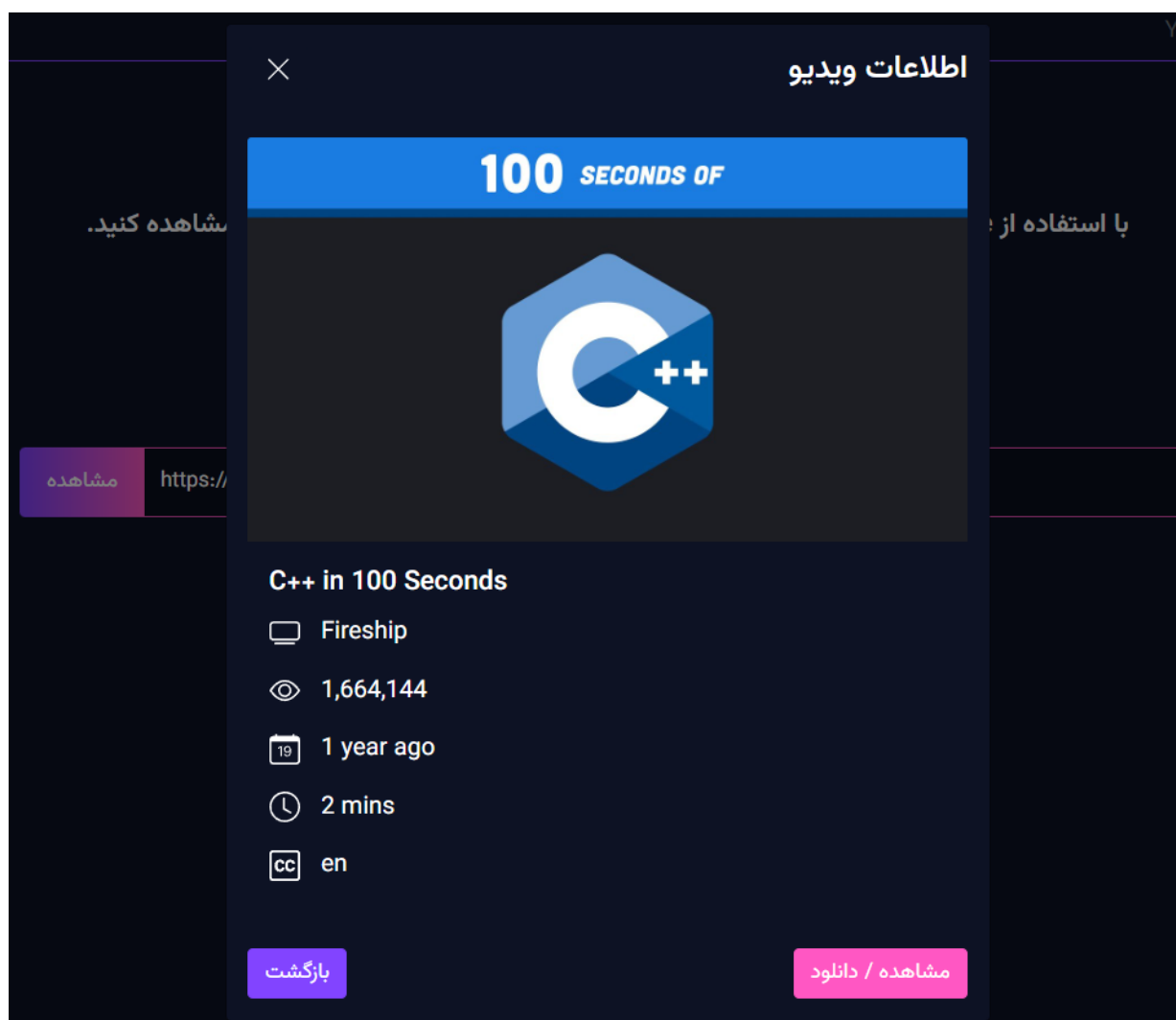
5. پنجمین معیار ارزیابی ما امکان مشاهده ویدئو در حالت وجود VPN در سمت کاربر می‌باشد. کاربر در صفحه مربوط به نمایش ویدئو گزینه مربوطه به نمایش در حالتی که نیاز به داشتن VPN ندارد را انتخاب کرده و باید بتواند ویدئو مورد نظر را تماشا کند.

6. ششمین معیار ارزیابی ما امکان نمایش ویدئو در حالت امبت د صفحات خارج از سایت می‌باشد کاربر باید بتواند قالب HTML لازم جهت ایجاد امکان نمایش امبت را مشاهده و از آن استفاده کند.

7. هفتمین معیار ارزیابی ما قابلیت جستجو YouTube است، کاربر باید بتواند در قسمت مربوطه جستجو انجام دهد و نتایج متناظر جستجو در YouTube را مشاهده کند تا بتواند به راحتی ویدئو مورد نظر خود را جهت مشاهده انتخاب کند.

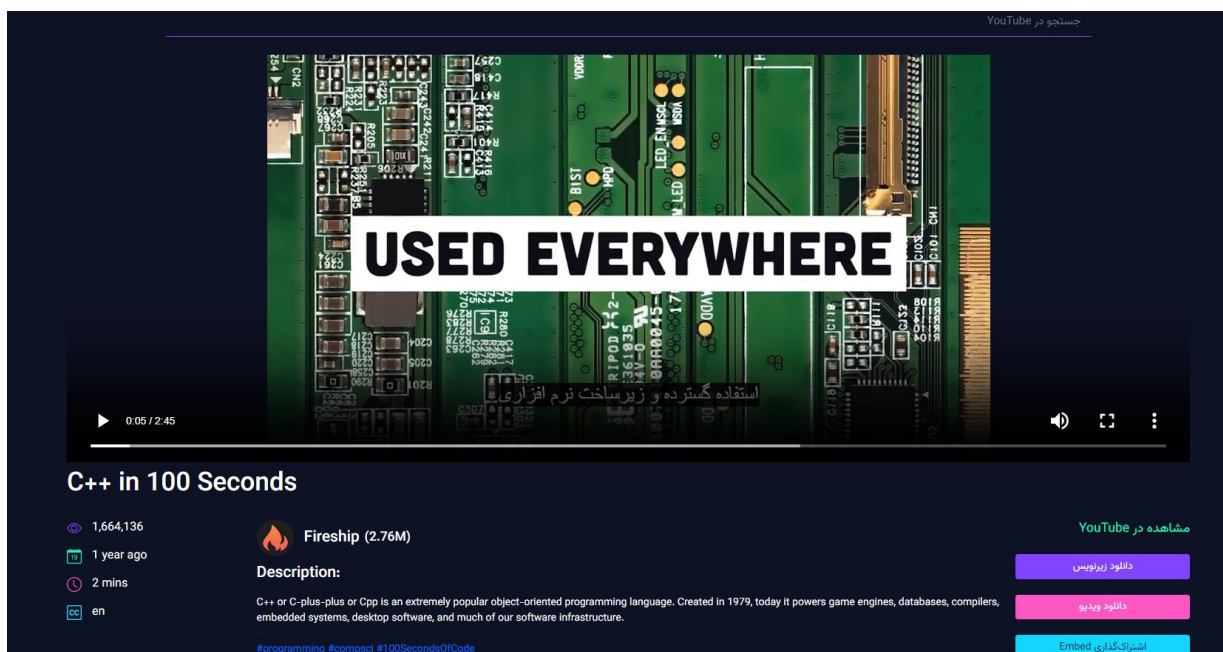
5-3- نتایج بدست آمده از طراحی رابط کاربری و نمایش اولیه ویدئو

در ابتدا کاربر وارد سایت می‌شود و در صفحه اصلی مکانی برای قرار دادن لینک ویدئو مد نظر خود مشاهده می‌کند شکل شماره (3-3) صفحه اصلی برنامه ما را نشان می‌دهد. پس از وارد کردن لینک ویدئو ابتدا صحت لینک بررسی می‌شود و در صورت درستی آن، اطلاعات آن با استفاده از API های Invidious دریافت و در یک Dialog Box نمایش داده می‌شود شکل شماره (3-6) این فرآیند را نمایش می‌دهد.



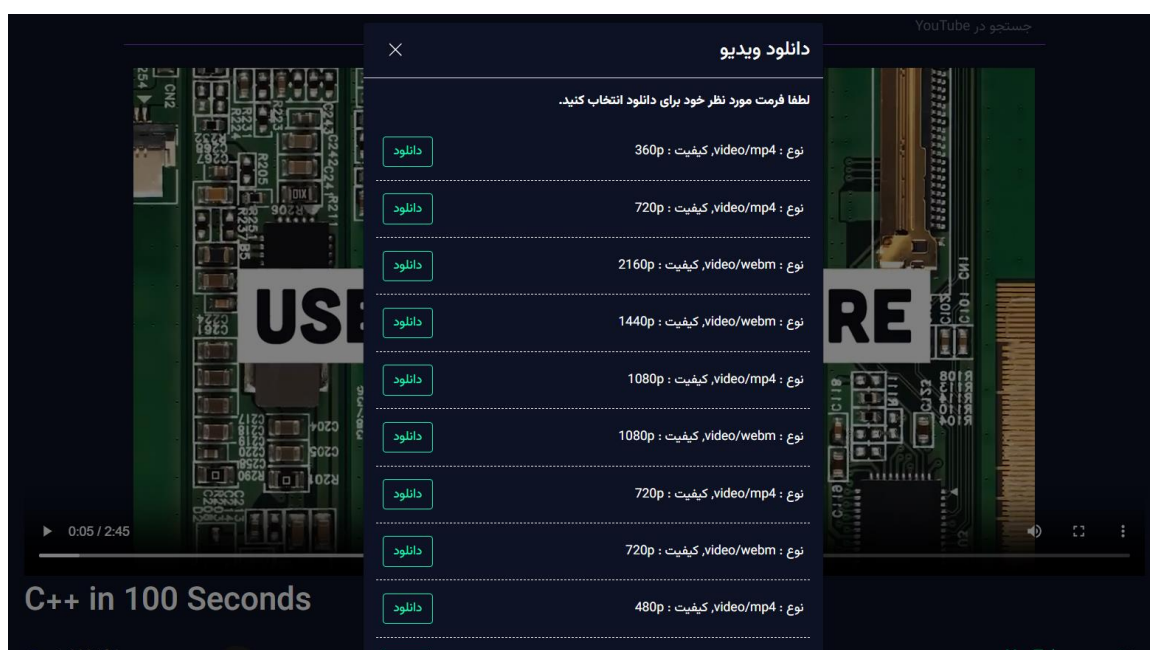
شکل (3-6) پیش‌نمایش اطلاعات ویدئو با شناسه MNeX4EGtR5Y

در ادامه کاربر به صفحه ویدئو مورد نظر خود می‌رود و آن را مشاهده می‌کند شکل شماره (3-7) صفحه مربوط به نمایش ویدئو به همراه زیرنویس فارسی آن را نشان می‌دهد.



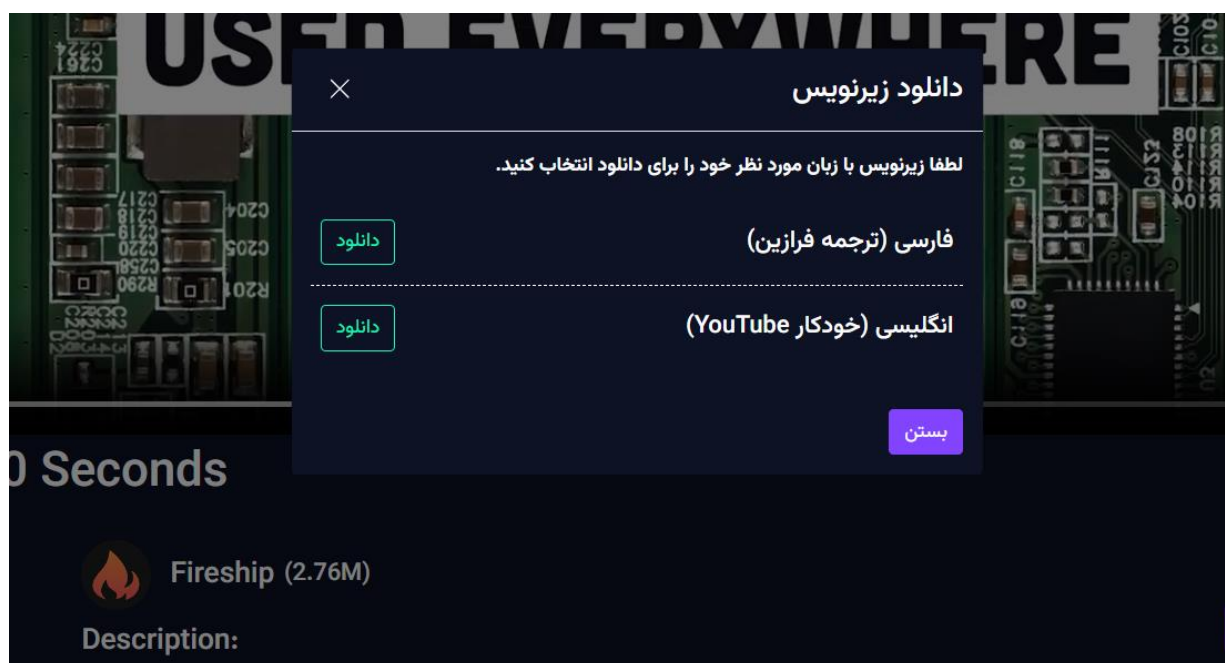
شکل (7-3) صفحه مخصوص نمایش ویدئو با شناسه MNeX4EGtR5Y

در ادامه کاربر بر روی گزینه "دانلود ویدئو" کلیک می‌کند و فرمت مورد نظر خود برای دانلود را انتخاب می‌کند. شکل شماره (3-8) لیست فرمت‌های مختلف ویدئو جهت دانلود کاربر را نمایش می‌دهد. لازم به ذکر است که فرمت‌های Audio نیز در انتهای این لیست قرار دارند.



شکل (8-3) لیست فرمت‌های دانلود ویدئو با شناسه MNeX4EGtR5Y

مشابه این فرآیند برای دانلود زیرنویس نیز اتفاق می افتد و کاربر می تواند زیرنویس هایی که برای ویدیو وجود دارند را دانلود کند. شکل (3-9) این قسمت از برنامه را نشان می دهد. همچنین این برنامه تفاوت میان زیرنویس انگلیسی عادی و زیرنویس انگلیسی ترجمه خودکار YouTube را تشخیص می دهد.

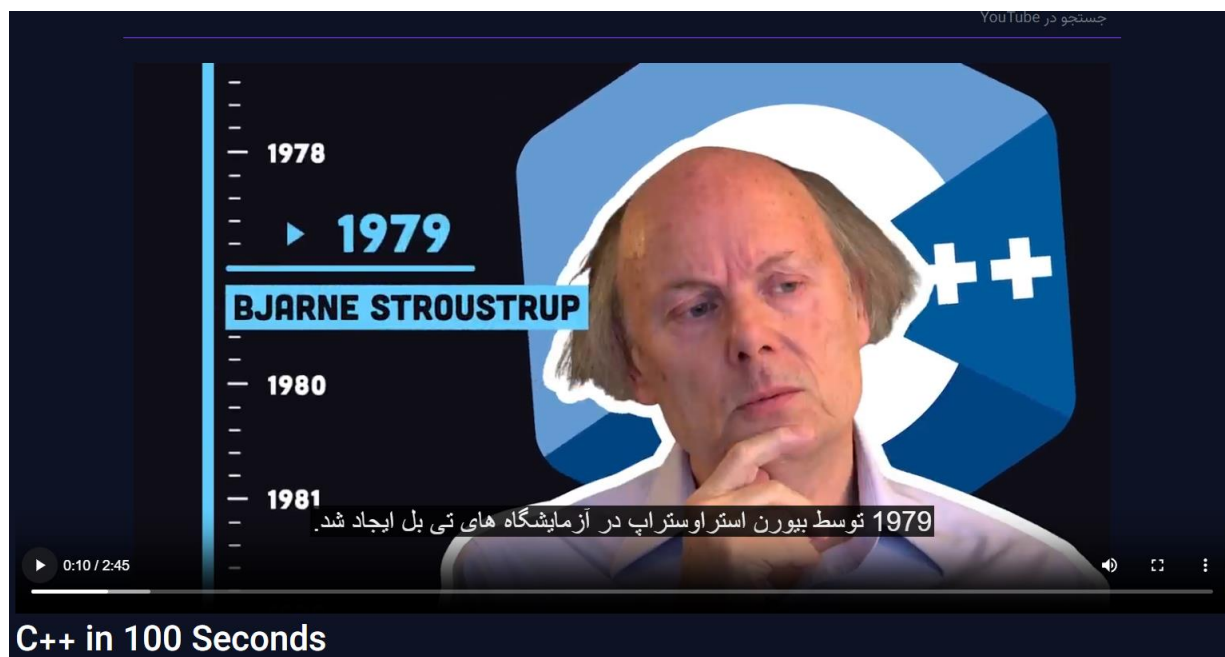


شکل (3-9) قابلیت دانلود زیرنویس ویدیو

در بخش بعدی کاربر حالت نمایش ویدیو بدون نیاز به VPN را انتخاب می کند، برنامه به او درباره این حالت اطلاع رسانی می کند و سپس ویدیو را به او نمایش می دهد، در این نوع از نمایش نیاز به داشتن VPN فعال نمی باشد. شکل های شماره (3-10) و (3-11) این فرآیند را نشان می دهند.



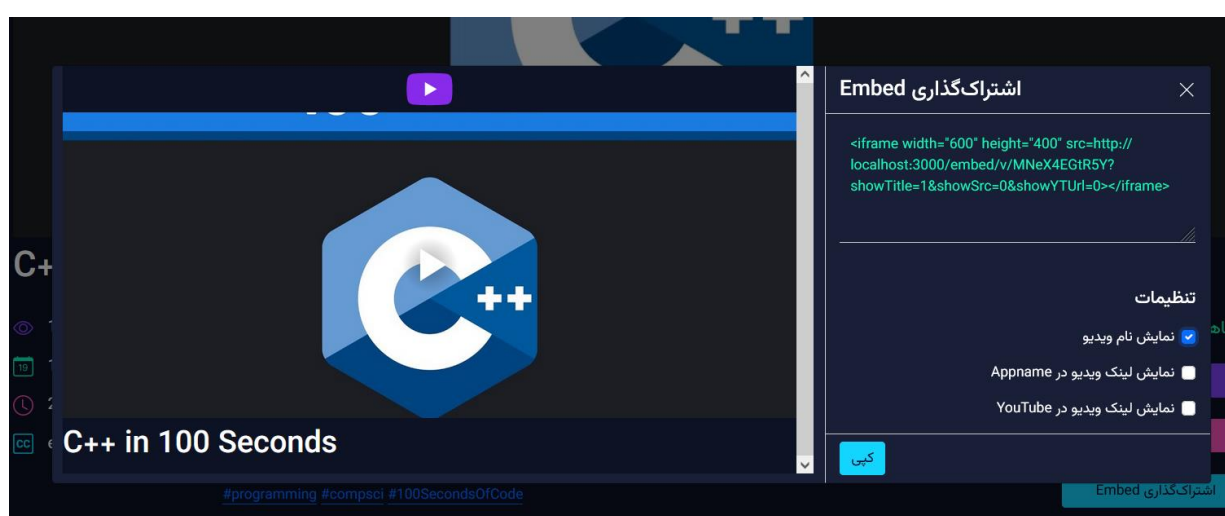
شکل (3-10) نمایش در حالت آزمایشی ویدیو با شناسه MNeX4EGtR5Y قسمت اول



شکل (3-11) نمایش در حالت آزمایشی ویدئو با شناسه MNeX4EGtR5Y قسمت دوم

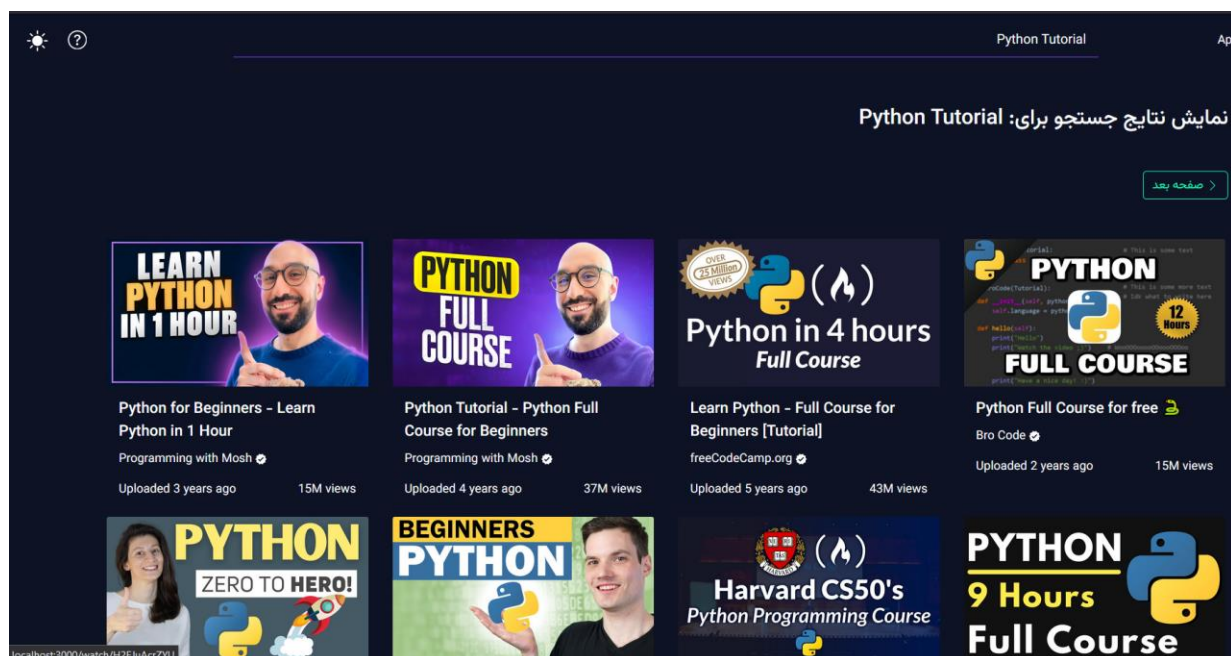
کاربر می‌خواهد ویدئو را به‌صورت Embed در صفحه دیگری قرار دهد که خارج از سایت ما نیز قابل مشاهده باشد و برای این کار گزینه اشتراک‌گذاری Embed را انتخاب می‌کند و می‌تواند قالب HTML مورد نظر خود را جهت مشاهده در خارج از سایت، ایجاد کند. شکل شماره (3-12) این قابلیت را نمایش می‌دهد.

در هنگام ایجاد این قالب می‌توان Customization هایی نیز انجام داد.



شکل (3-12) اشتراک‌گذاری Embed ویدئو

در نهایت نیز کاربر اقدام به جستجو ویدئوهای YouTube از طریق برنامه ما می‌کند و ویدئوهای مربوط به نتایج جستجو او در YouTube به او نمایش داده می‌شود. شکل شماره (3-13) نتایج جستجو YouTube کاربر با عبارت “Python Tutorial” در برنامه ما را نشان می‌دهد.



شکل (3-13) نتایج جستجو ویدئو YouTube در برنامه

تصاویر بالا مجموعه‌ای از قابلیت‌های سیستم توسعه‌داده شده را نشان داد و در ادامه درباره جزئیات پیاده‌سازی این سیستم توضیحاتی ارائه خواهیم داد.

3-6- تحلیل نتایج

در این فصل از ابزارهایی که می‌توانستند به ما کمک کنند جهت پیاده‌سازی پروژه استفاده کردیم و فرآیند نمایش ویدئو به همراه زیرنویس فارسی مطابق خواسته ما صورت می‌گیرد.

7-3- خلاصه و جمع‌بندی

فصل سوم به طور عمده در برگیرنده‌ی طراحی سیستم نمایش ویدئو و ایجاد زیرنویس فارسی و جزئیاتی درباره ابزارهای مورد استفاده ما برای ایجاد این سیستم بود. در فصل بعد به جزئیات پیاده‌سازی این سیستم می‌پردازیم.

فصل 4: پیاده سازی سیستم دریافت اطلاعات و نمایش ویدیو و ایجاد زیرنویس فارسی

پس از طراحی‌های انجام شده در فصل قبل، در این فصل به جزئیات پیاده‌سازی برنامه خود می‌پردازیم و همچنین مروری بر قسمت‌های مختلف برنامه خود همراه با نمایش کدها خواهیم داشت.

4-1- مقدمه

در این فصل به جزئیات پیاده سازی این سیستم و معماری بخش های Client و Server آن می پردازیم و جزئیات نحوه استفاده از کتابخانه ها و سرویس هایی که در بخش های قبل بیان کردیم را ارائه می دهیم.

4-2- نحوه پیاده سازی

به طور کلی ساختار برنامه ما به دو بخش Front-end و Back-end تقسیم می شود که بخش Front-end با استفاده از React و بخش Back-end با استفاده از Django توسعه داده شده است. در ادامه درباره جزئیات پیاده سازی هر کدام از آن ها توضیحات لازمه را ارائه می دهیم.

4-3- قسمت Client یا Front-end

در این بخش به جزئیات پیاده سازی قسمت Client برنامه خود می پردازیم. برای پیاده سازی این قسمت همانطور که در بخش های قبل نیز ذکر شده است، از React استفاده کرده ایم.

ساختار کلی برنامه React و نحوه مدیریت URL ها در آن را می توانیم با بررسی فایل App.js در پروژه React خود، متوجه شویم. شکل شماره (4-1) تابع App در فایل App.js جهت مدیریت مسیرها و Map کردن Component ها به آدرس URL را نمایش می دهد.

```
function App() {
  return (
    <div className="App">
      <Router>
        <Routes>
          <Route path="/" element={<Layout />} />
          <Route index element={<Home />} />
          <Route path="watch/:id" element={<VideoPlayer exp={false} />} />
          <Route path="exp/:id" element={<VideoPlayer exp={true} />} />
          <Route path="faq" element={<FAQ />} />
          <Route path="search" element={<SearchResults />} />
          <Route path="*" element={<Error code={404} />} />
        </Routes>
        <Route path="embed/" element={<EmbedLayout />} />
        <Route path="v/:id" element={<EmbedPlayer exp={false} />} />
      </Routes>
    </Router>
  </div>
);
}
```

شکل (1-4) ساختار برنامه React ایجادشده در تابع App

با توجه به شکل بالا Component های ایجاد شده را به ترتیب توضیح می‌دهیم.

هر Component با نام Route را در واقع می‌توان به نوعی "Map" کردن یک Component به یک آدرس URL در برنامه ما در نظر گرفت که با وارد کردن آن URL آن Component Render می‌شود. آدرس مورد نظر و Component متناظر با آن برای Render شدن به عنوان Attribute های path و element به Route داده می‌شوند.

مشاهده می‌شود که دو Component با نام Route داریم که هر کدام شامل چند Route دیگر زیرمجموعه آن می‌باشد. این دو Component، Layout و EmbedLayout نام دارند.

همانگونه که از اسم آنها نیز مشخص است، این دو Component برای ایجاد قالب کلی صفحات ایجاد شده‌اند. Layout بیانگر قالب تمامی صفحات سایت به جز صفحه مخصوص به نمایش Embed می‌باشد. قالب صفحه Embed نیز به صورت جداگانه ایجاد شده که EmbedLayout نام دارد. حال به بررسی جزئیات Layout برای درک نحوه کارکردن آن می‌پردازیم.

شکل شماره (4-2) ساختار فایل Layout را نمایش می‌دهد.


```
import { Outlet } from "react-router-dom";
import Navbar from "../shared/navbar/Navbar";
import Footer from "../shared/footer/Footer";

export default function Layout() {
  return (
    <div className="rtl bg-1 text-1">
      <div className="clear"></div>
      <Navbar />
      <div className="clear"></div>
      <Outlet />
      <div className="clear"></div>
      <Footer />
    </div>
  );
}
```

شکل (2-4) ساختار Component با نام Layout

همانگونه که از شکل بالا مشخص است ساختار فایل Layout که قالب نمایش صفحات است، شامل Component های Navbar و Footer می باشد و Component دیگری در میان آنها با نام Outlet وجود دارد که این Component توسط خود React ساخته شده و وظیفه آن Render کردن فرزند Component اصلی که در آن از Outlet استفاده شده است، می باشد. در اینجا این Component اصلی همان Layout است و فرزندان آن Component های Home و VideoPlayer و ... می باشند که طبق این ساختار، بسته به URL وارد شده، قالب صفحات ایجاد می شود که مثلاً برای صفحه اصلی یا Index ساختار یا قالب صفحه ابتدا شامل بخش های Navbar و سپس Home و سپس Footer می باشد و برای بقیه صفحات نیز به این صورت Rendering انجام می شود. مشابه این فرآیند برای EmbedLayout نیز صورت می گیرد.

در ادامه به بخش های دیگر برنامه React خود می پردازیم. همانگونه که در بخش قبل اشاره کردیم، برای دریافت اطلاعات از YouTube تصمیم به استفاده از API های Invidious گرفتیم. ارتباط ما با این سرویس در بخش Client و در برنامه React ما صورت می گیرد. برای ارسال درخواست های HTTP به منظور ارتباط با سرویس های خارجی از یک HTTP Client با نام Axios استفاده می کنیم. ساختار Axios به صورت Promise-based می باشد به این معنا که فرآیند دریافت اطلاعات توسط آن به صورت Asynchronous صورت می گیرد و این بدین معنی است که همزمان با دریافت اطلاعات بخش های دیگر برنامه نیز اجرا می شوند و منتظر پایان آن نمی مانند تا بعد از آن

شروع به اجرا شدن کنند، برخلاف عملیات‌های Synchronous که در آن تمامی عملیات‌ها پشت سر هم و به صورت Sequential اجرا می‌شوند.

با استفاده از Axios با API های Invidious ارتباط برقرار می‌کنیم و از آنها اطلاعات لازمه را دریافت می‌کنیم. شکل شماره (3-4) نحوه استفاده از Axios برای ارتباط با Invidious را نشان می‌دهد.

```
export async function getVideoData(videoId) {
  try {
    const response = await axios.get(
      `_${getInvidiousCurrentInstance()}/api/v1/videos/${videoId}`
    );
    return response.data;
  } catch (error) {
    throw new Error(error);
  }
}
```

شکل (3-4) دریافت اطلاعات ویدیو از Invidious با استفاده از Axios

همانگونه که در شکل بالا مشخص است، از یک کلمه کلیدی¹ تحت عنوان await استفاده کردیم. با استفاده از await می‌توانیم در یک عملیات asynchronous، اجرای دستور بعد از یک فرآیند async را وابسته به پایان آن دستور کنیم. در واقع در کد شکل (3-4) ابتدا مقدار متغیر response با خروجی عملیات axios مقداردهی می‌شود و سپس به عنوان خروجی بازگردانده می‌شود. در غیراینصورت خروجی response.data ممکن است undefined باشد.

لازم به ذکر است که استفاده از await فقط در block ها (توابعی) که با کلمه کلیدی async مشخص شده باشند، امکان‌پذیر است.

در کد شکل (3-4) از تابعی به نام getInvidiousCurrentInstance نیز استفاده شده که این تابع درواقع Instance یا همان دامنه سروری که می‌خواهیم از آن استفاده کنیم را برمی‌گرداند و در صورت بروز مشکلی در یک Instance می‌توانیم Instance خود را تعویض کنیم تا بتوان دوباره از اطلاعات Invidious استفاده کرد.

در هنگامی که کاربر لینک ویدیو خود را وارد می‌کند، پس از بررسی صحت آن، ID ویدیو را که دارای 11 کاراکتر می‌باشد، از آن استخراج کرده و تابع شکل (3-4) و توابع مشابه آن را صدا می‌زنیم.

¹ Keyword

برای نمایش ویدیو از package ای با نام React-player استفاده کردیم که با استفاده از دستور زیر قابل نصب می باشد:

```
npm i react-player
```

برای استفاده از این video player ابتدا یک component با نام VideoPlayer ایجاد می کنیم و از آن در این Component استفاده می کنیم.

شکل شماره (4-4) استفاده از ReactPlayer برای نمایش ویدیو در حالت عادی (نیازمند به vpn) را نشان می دهد.

```
return (
  <>
    {videoUrl === undefined && <VideoLoading />}
    {videoUrl !== undefined && (
      <ReactPlayer
        url={url}
        controls
        width="100%"
        height="100%"
        light={videoThumbnail}
        config={{
          file: {
            tracks: [
              {
                kind: "subtitles",
                src: `/api/v1/captions/${videoId}?translate=fa`,
                srcLang: "fa",
                default: true,
              },
              {
                kind: "subtitles",
                src: `/api/v1/captions/${videoId}?lang=en`,
                srcLang: "en",
                default: false,
              },
            ],
          },
        }}
      />
    )}
  </>
);
```

شکل (4-4) ساختار VideoPlayer برای نمایش ویدیو در حالت عادی (نیازمند به VPN)

مطابق شکل نشان داده شده، این component برای نمایش ویدیو به همراه زیرنویس، لینک ویدیو و زیرنویس را دریافت می کند که در این حالت، لینک ویدیو در دامنه googlevideo و لینک زیرنویس نیز، API متناظر در سرور برای دریافت آن است که در بخش بعد درباره دریافت آن در سرور توضیحاتی ارائه می دهیم.

لازم به ذکر است که لینک Thumbnail ویدیو نیز در دامنه Instance های Invidious است و بدون نیاز به VPN نیز قابل دسترسی است.

در قسمت نمایش بدون نیاز به VPN نیز که به عنوان "حالت آزمایشی" معرفی شده، نیز نمایش ویدیو به همین صورت بوده با این تفاوت که لینک نمایش ویدیو، لینک سرور ماست که به گونه ای یک رابط بین Client و سرور Googlevideo می باشد. شکل شماره (4) ساختار VideoPlayer در این بخش را نمایش می دهد.

```
<VideoPlayer
  videoUrl={` /api/v1/tunnel/blob?url=${encodeURIComponent(
    | videoStreamData.url
  )}&id=${id}`}
  thumbnail={videoData.videoThumbnails[0].url}
  videoId={id}
  onError={handleVideoPlayingError}
/>
```

شکل (4-5) ساختار VideoPlayer برای نمایش ویدیو در حالت آزمایشی (بدون نیاز به VPN)

درباره جزئیات این قسمت سرور در بخش های بعدی توضیحاتی ارائه خواهیم داد.
در هر دو حالت ذکر شده، نمایش ویدیو به صورت پیش فرض با بهترین کیفیت آن صورت می گیرد.

4-4- قسمت Server یا Back-end

در این قسمت در رابطه با پیاده سازی سرور خود در محیط Django توضیح می دهیم.

سرور ما در Django به سه app تقسیم شده است، Stream، Captions و Invidious که به ترتیب هر کدام را توضیح می دهیم.

1-4-4- بخش Stream در سرور

از این بخش سرور برای دریافت اطلاعات و محتوای ویدیو و تصاویر در YouTube استفاده می‌کنیم. این برنامه به طور کلی دارای سه تابع در بخش View و سه API متناظر با آن‌ها می‌باشد. موارد زیر وظایف این View ها در برنامه Stream سرور ما می‌باشند:

دریافت اطلاعات لینک و فرمت ویدیو: شکل شماره (4-6) تابعی با نام `get_video_stream_format_data` را به ما نمایش می‌دهد. وظیفه این تابع با توجه به پارامتر با نام `"type"` مشخص می‌شود، در صورتی که مقدار این `query parameter` برابر با `"all"` باشد، این تابع تمامی فرمت‌ها و کیفیت‌هایی که ویدیو در آن‌ها نیز وجود دارد مانند `mp4/720p` یا `audio/webm` به همراه آدرس(لینک) آن‌ها را برمی‌گرداند، در صورتی که مقدار `type` برابر `"default"` باشد، یا اینکه `query parameter` ذکر شده در API نیامده باشد، تابع اطلاعات ذکر شده برای فرمت و لینک ویدیو را، تنها برای بالاترین کیفیت آن بدست آورده و آن را برمی‌گرداند.

```
@api_view(['GET'])
def get_video_stream_format_data(request, video_id):
    format_type = request.GET.get('type', 'default')

    if not youtube_video_exists(f"https://youtu.be/{video_id}"):
        return HttpResponse("Video does not exist!", status=404)

    try:
        if format_type == 'all':
            stream_formats = get_video_stream_formats(video_id)
            return Response(stream_formats)
        elif format_type == 'default':
            stream_data = get_video_default_stream_data(video_id)
            return Response(stream_data)
        else:
            return HttpResponse(status=400)

    except Exception as e:
        return HttpResponse('Connection Failed!', status=500)
```

شکل (4-6) دریافت اطلاعات Stream ویدیو مورد نظر در YouTube

این تابع برای انجام عملیات ذکر شده از تابع Pytube که در بخش‌های قبل به آن اشاره کردیم، استفاده می‌کند.

دریافت Thumbnail ویدیو: همانگونه که در بخش‌های قبل اشاره کردیم، لینک‌های Thumbnail ویدیوهای YouTube ممکن است حتی در صورت استفاده از API های Invidious قابل دسترسی نباشند. جهت حل این مشکل تابعی را ایجاد می‌کنیم که با دادن درخواست به لینک یک تصویر، اطلاعات آن را دریافت کرده و بصورت binary آن را انتقال دهد. این اطلاعات انتقال داده شده می‌توانند در سمت Client به فرم base64 دریافت پردازش شوند. برای انجام این فرآیند از کتابخانه Pillow استفاده می‌کنیم. شکل شماره (4-7) تابع ذکر شده را نمایش می‌دهد.

```
@api_view(['GET'])
def get_video_thumbnail(request, video_id):
    if not youtube_video_exists(f"youtu.be/{video_id}"):
        return HttpResponse(status=404)
    try:
        url = f'https://www.youtube.com/watch?v={video_id}'
        yt = YouTube(url)
        thumbnail_url = yt.thumbnail_url
        response = BytesIO()
        Image.open(requests.get(thumbnail_url, stream=True).raw).save(response, format='PNG')
        image_data = response.getvalue()
        return HttpResponse(image_data, content_type='image/png')
    except Exception as e:
        return HttpResponse('Connection Failed!', status=500)
```

شکل (4-7) دریافت و انتقال اطلاعات Thumbnail ویدیو

انتقال محتوای ویدیو YouTube: همانگونه که در بخش‌های قبل ذکر شد، برای ایجاد امکان مشاهده ویدیو بدون نیاز به داشتن VPN، باید سرور ما به گونه‌ای یک رابط بین Client و دامنه Googlevideo شود و به عنوان یک "تونل" برای انتقال اطلاعات عمل کند. برای پیاده‌سازی این ویژگی از نوعی از پاسخ HTTP در Django تحت عنوان StreaminHttpResponse استفاده می‌کنیم. این سیستم به ما قابلیت ارسال اطلاعات در بسته‌ها یا chunk های کوچک را می‌دهد. از این class معمولاً هنگامی استفاده می‌شود که می‌خواهیم داده بزرگی را ارسال کنیم اما نمی‌خواهیم تمام اطلاعات آن را کاملاً دریافت و در حافظه load کنیم بلکه بخواهیم آن را قسمت

به قسمت ارسال کنیم. شکل شماره (4-8) تابع tunnel_video_blob_content را نمایش می دهد که وظیفه آن دریافت و ارسال محتوای Video مورد نظر با دریافت لینک آن می باشد.

```
@api_view(['GET'])
def tunnel_video_blob_content(request):
    url = request.GET.get('url', '')
    if url == '':
        return HttpResponse('Error: Specify video URL', status=400)

    video_response = requests.get(url, stream=True)
    content_type = video_response.headers.get('Content-Type', 'video/mp4')
    response = StreamingHttpResponse(
        video_response.iter_content(chunk_size=1024 * 1024),
        content_type=content_type,
    )
    return response
```

شکل (4-8) دریافت اطلاعات Stream ویدیو مورد نظر در YouTube

شکل شماره (4-9) نیز ساختار API ها در برنامه Stream را نشان می دهد.

```
urlpatterns = [
    path('stream/<str:video_id>/', views.get_video_stream_format_data, name='get_video_stream_format_data'),
    path('thumbnail/<str:video_id>/', views.get_video_thumbnail, name='get_video_thumbnail'),
    path('tunnel/blob/', views.tunnel_video_blob_content, name='tunnel_video_blob_content'),
]
```

شکل (4-9) فایل urls.py در برنامه Stream در سرور

4-4-2- بخش Captions در سرور

وظیفه این برنامه در سرور ما دریافت، ارسال و ترجمه اطلاعات زیرنویس ویدیوها می باشد. در این بخش از کتابخانه Youtube_transcript_api که در بخش های قبل به آن اشاره شد، استفاده می کنیم. این کتابخانه امکان دریافت محتوای زیرنویس ویدیوها و همچنین تبدیل آن به فرمت مورد نظرم و همچنین دریافت زیرنویس ترجمه اتوماتیک YouTube برای ویدیو در تمامی زبان ها (در صورت فعال بودن این ویژگی برای ویدیو) را ارائه می دهد. ارتباط با مترجم فرازین جهت

استفاده از سرویس های ترجمه آن نیز در این بخش صورت می گیرد. لازم به ذکر است که جهت نمایش و انتقال اطلاعات زیرنویس، از فرمت VTT استفاده می کنیم.

3-4-4- بخش Invidious در سرور

وظیفه این برنامه در سرور ما دریافت اطلاعات از Invidious است. در بخش های قبل ذکر شده که ارتباط با Invidious از طریق برنامه Client صورت می گیرد که این کار باعث کاهش load بر روی سرور نیز می شود. اما در حالتی که ارتباط با API های Invidious نیز مانند سایت YouTube بدون داشتن VPN امکان پذیر نبود، می توان از API های ایجاد شده در این بخش از برنامه استفاده کرد تا باعث از دست رفتن کارایی برنامه نشود.

شکل شماره (4-10) نمونه ای از دریافت اطلاعات از Invidious در این بخش از برنامه که برای دریافت اطلاعات یک ویدیو می باشد را نمایش می دهد.

```
def get_video_data(video_id):
    try:
        response = requests.get(
            f"{get_invidious_current_instance()}api/v1/videos/{video_id}"
        )
        return response.json()
    except requests.exceptions.RequestException as e:
        raise Exception(e)
```

شکل (4-10) دریافت اطلاعات از Invidious در سمت سرور

3-4-5- خلاصه و جمع بندی

در این فصل درباره جزئیات پیاده سازی پروژه خود و بخش های مختلف آن توضیحاتی ارائه دادیم و به کاربرد برنامه هایی مانند React و Django در پروژه خود اشاره کردیم. همچنین تصاویری از کدهای نوشته شده برای بخش های مختلف پروژه نمایش دادیم.

فصل 5

فصل 5: جمع‌بندی، نتیجه‌گیری و پیشنهادها

1-5- جمع‌بندی

در این تحقیق در گام نخست ما با مفهوم Video Streaming و نحوه کارکردن YouTube آشنا شدیم، سپس به معرفی روش‌های مختلف دریافت اطلاعات از آن پرداختیم و درباره کتابخانه‌ها و Framework هایی که میتوانند در پیاده‌سازی این پروژه به ما کمک کنند، اطلاعاتی ارائه دادیم. در نهایت با استفاده از موارد ذکر شده، پروژه خود را پیاده‌سازی کردیم و مطابق با معیارهای ارزیابی خود، صحت کارایی آن را بررسی کردیم. نتیجه نهایی بدست آمده، مورد رضایت بود.

2-5- نتیجه‌گیری

در این پروژه هدف نهایی ما این بود که افراد بتوانند ویدیوهای مورد نظر خود را با زیرنویس فارسی مشاهده کنند و پس از اجرای موارد گفته شده در بخش‌های قبل به هدف خود رسیدیم.

1-2-5- نوآوری / دستاوردها

در این پروژه علاوه بر هدف اصلی خود، که نمایش ویدیو همراه با زیرنویس فارسی بود، قابلیت‌های جدیدی نیز به سیستم خود اضافه کردیم که از جمله آن‌ها می‌توان به امکان نمایش ویدیو بدون نیاز به VPN و همچنین قابلیت جستجو مستقیم YouTube در برنامه خود اشاره کرد.

2-2-5- محدودیتها

با وجود رسیدن به دستاوردهای مورد نظر و ایجاد قابلیت‌های جدید جهت بهبود تجربه کاربر، در اجرای این برنامه محدودیتی نیز وجود دارد. در حالتی که انتقال اطلاعات فایل‌های ویدیو/تصویر از طریق ارسال درخواست در سرور و انتقال محتوا Binary توسط آن صورت می‌گیرد، در صورت افزایش میزان درخواست و load بر روی سرور، امکان بروز Bottleneck و یا کاهش performance در آن وجود دارد.

3-2-5- پیشنهادها

برای بهبود Performance و همچنین تجربه کاربر، پیشنهادهایی ارائه می‌شود. می‌توان برای بهبود زمان پاسخگویی و عملکرد سرور از یک پایگاه داده برای ذخیره لینک ویدیوهای YouTube استفاده کرد، دریافت لینک ویدیو و اطلاعات Stream آن، می‌تواند زمان‌بر باشد اما در صورتی که با هر با درخواست کاربر برای نمایش ویدیو قبل از اینکه فرآیند اطلاعات Stream آغاز شود، ابتدا بررسی کنیم که آیا این اطلاعات در پایگاه داده وجود دارد یا خیر و در صورت عدم وجود آن، اقدام به ذخیره‌سازی آن کنیم، فرآیند دریافت اطلاعات Stream ویدیو می‌تواند سریع‌تر باشد تا تجربه بهتری برای کاربر رقم زند. لازم به ذکر است که مواردی مانند امکان منقضی شدن لینک‌های ویدیو و مدیریت آن باید در پیاده‌سازی این قابلیت، در نظر گرفته شود.

فصل 6

فصل 6: مراجع

مراجع

- [1] <https://support.google.com/youtube/answer/2734796?hl=en#zippy=%2Cauto-translate>
- [2] https://support.google.com/youtube/answer/6373554?hl=en&ref_topic=7296214
- [3] <https://www.faraazin.ir/about>
- [4] <https://developers.google.com/youtube/v3/docs>
- [5] https://developers.google.com/youtube/v3/guides/quota_and_compliance_audits
- [6] <https://docs.invidious.io/>
- [7] <https://github.com/iv-org/invidious>
- [8] <https://github.com/iv-org/invidious/issues/3872>
- [9] https://www.google.com/get/videoqualityreport/#what_youtube_is_doing
- [10] <https://developers.google.com/static/media/pdf/Youtube-HTML5-VP9-Case-Study.pdf>

پیوست‌ها

پیوست الف:

لینک Repository برنامه در Github : <https://github.com/alimrn001/UT-BSc-project>

Abstract:

YouTube is one of the largest video-sharing platforms and a key entertainment hub. Users worldwide engage with the platform for various purposes, watching videos across a wide range of content. Platforms like YouTube and streaming services utilize various methods to create an enjoyable user experience. Among these methods, content localization is noteworthy.

Content localization has always been a focal point, and different techniques are employed to achieve it. In the context of displaying videos on YouTube, one such approach involves creating different subtitles for the languages of the target audience. Despite advancements, there are limitations and a lack of localized content in the Persian language for many videos on YouTube in our country. Having a system capable of addressing this gap can be highly beneficial.

To achieve this goal, content localization will be carried out through subtitle translation. To implement this system, we will first examine how YouTube operates and explore methods to retrieve information from it. Subsequently, we will define the structure of our program. Based on our designed model, we will develop our web application using Django and React frameworks. Additionally, to fetch content, subtitles, and videos from YouTube, we will utilize available libraries in the Python programming language.

Keywords: Subtitles, YouTube, Web Application, Localization



University of Tehran



College of Engineering

School of Electrical and Computer Engineering

Thesis Title

A thesis submitted to the Undergraduate Studies Office

In partial fulfillment of the requirements for

The degree of Bachelor in

Computer Engineering

By:

Ali Mehrani

Supervisor:

Dr. Hesham Faili