# Homework 8: Angular, Ajax, JSON, Responsive Design, and Node.js
# CSCI571 Spring 2021

## 1. Objectives

- Get familiar with AJAX and JSON technologies
- Use a combination of HTML5, Bootstrap and Angular on client side
- Use Node.js on server side
- Get familiar with Bootstrap to enhance the user experience using responsive design
- Get hands-on experience of Cloud services hosting NodeJS/Express
- Learn to use popular APIs such as TMDB Api

## 2. Background

### 2.1 AJAX and JSON

AJAX (Asynchronous JavaScript + XML) incorporates several technologies

- Standards-based presentation using XHTML and CSS
- Result display and interaction using the Document Object Model (DOM)
- Data interchange and manipulation using JSON
- Asynchronous data retrieval using XMLHttpRequest
- JavaScript binding everything together

### 2.2 Bootstrap

Bootstrap is a free collection of tools for creating responsive websites and web applications. It contains HTML and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. To learn more details about Bootstrap please refer to the lecture material on Responsive Web Design (RWD). You should use **Bootstrap 4** in this homework.

### 2.3 Cloud Services

#### 2.3.1 Google App Engine (GAE)

Google App Engine applications are easy to create, easy to maintain, and easy to scale as your traffic and data storage needs change. With App Engine, there are no servers to maintain. You simply upload your application and it's ready to go. App Engine applications automatically scale based on incoming traffic. Load balancing, micro services, authorization, SQL and noSQL databases, memcache, traffic splitting, logging, search, versioning, roll out and roll backs, and security scanning are all supported natively and are highly customizable.
To learn more about GAE support for Node.js visit this page:

https://cloud.google.com/appengine/docs/standard/nodejs/

### 2.4 Angular

Angular is a toolset for building the framework most suited to your application development. It is fully extensible and works well with other libraries. Every feature can be modified or replaced to suit your unique development workflow and feature needs. Angular combines declarative templates, dependency injection, end to end tooling, and integrated best practices to solve development challenges. Angular empowers developers to build applications that live on the web, mobile, or the desktop.
For this homework, Angular 8+ (Angular 8, 9 or 10) can be used, but **Angular 10 is recommended**. Please note Angular 8+ will need familiarity with Typescript and component-based programming.

To learn more about Angular 8+, visit this page:

https://angular.io/

### 2.5 Node.js

Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. Node.js uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. Node.js package ecosystem, **npm**, is the largest ecosystem of open source libraries in the world.

To learn more about Node.js, visit:

https://Node.js.org/en/

Also, **Express.js** is strongly recommended. Express.js is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications. It is in fact the standard server framework for Node.js.

To learn more about Express.js, visit:

http://expressjs.com/

You can use nodemon while developing, visit:

https://www.npmjs.com/package/nodemon

**All TMDB API calls should be done through your Node.JS server otherwise will cause 4-point penalty.**

## 3. High Level Description

In this exercise, you will create a webpage that allows you to search for information regarding movies and TV shows using the TMDB API and upon selection can display results on the details page of that specific movie or TV show. You can view popular, top-rated and trending movies and TV shows on the home page along with currently playing movies. The application evolves from previous homework.

The user will first open Home page, where the navbar contains Search Bar and user can enter the movie or TV show name and select from a list of matching movie/TV show names along with images using "autocomplete". Selecting the movie or TV show name will lead to the details page of that movie or TV show.

At the top of Home page, a Carousel of currently playing movie posters is displayed (See below **Figure 1.1**). Recently visited movies and TV shows are listed under Continue Watching in the Home page. Popular, Top Rated and Trending movies and TV shows are displayed below it.

There are 4 routes for this application:

a) Home page Route ['/'] – It is a default route of this application.
b) Movie Details Route ['/watch/movie/<id>'] – It displays the details of the movie with id=<id>
c) TV show Details Route [ '/watch/tv/<id>'] – It displays the details of the TV show with id=<id>
d) Watchlist Route ['/mylist'] – It displays the watchlist of the user.

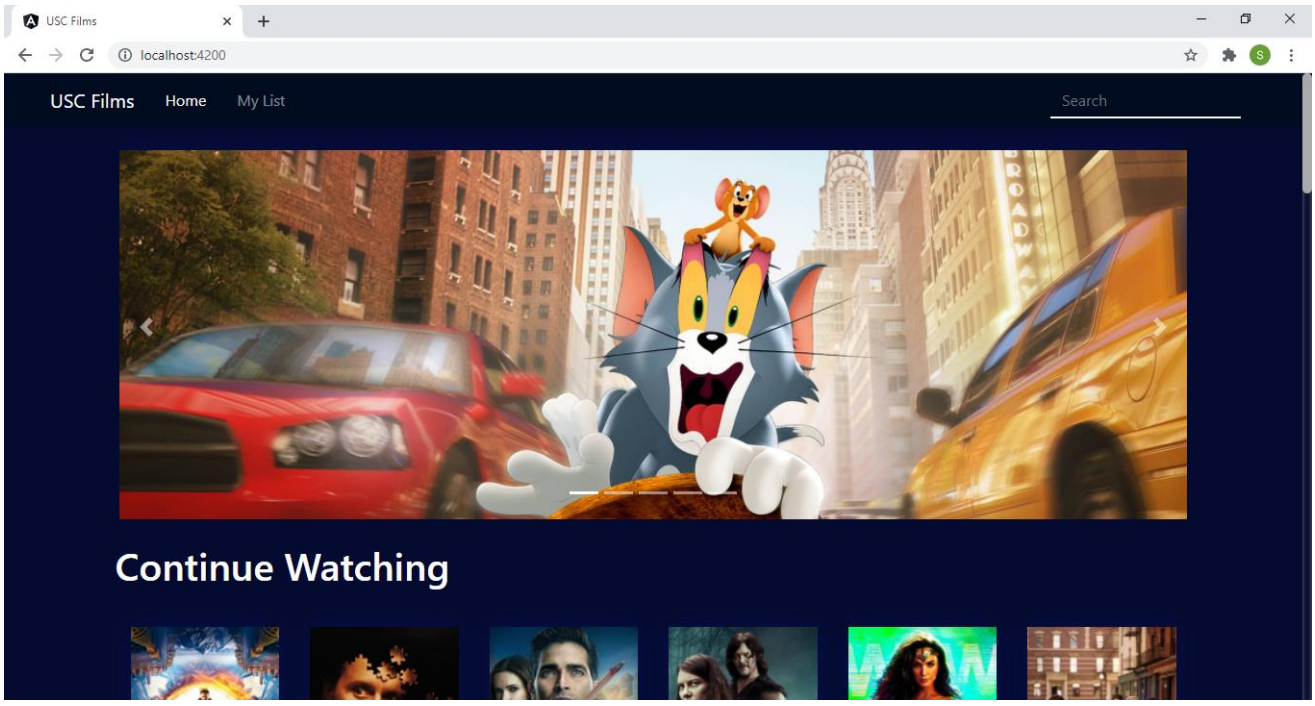



**Figure 1.1:** Initial Home Page

Instructions on how to **use the API are given in Section 4**. All implementation details and requirements will be explained in the following sections.

### ~~3.1 Navbar~~

The Navigation bar must be present on top of the page and be visible all the time. You can use Bootstrap to create a navbar. It consists of following menu options:

- Home
- MyList
- Search bar

"USC Films" upon clicking will lead to the Home page.

#### 3.1.1 Search Functionality

You must replicate the Search Bar at Top-right in the navigation bar displayed in **Figure 1.1** using **ngbTypeahead from ng-bootstrap** (**Refer section 5.3**).

The Search Bar allows user to enter a keyword (Movie name / TV show name) to retrieve information. Based on the user input, the text box should display a list of top 7 matching movies and TV shows names along with their poster images (see **Figure 1.2**). The autocomplete JSON data is retrieved from the ***TMDB Multi-Search API*** **(refer to Section 4.1.1).**
The following is an example of calling this API:
https://api.themoviedb.org/3/search/multi?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&query=game

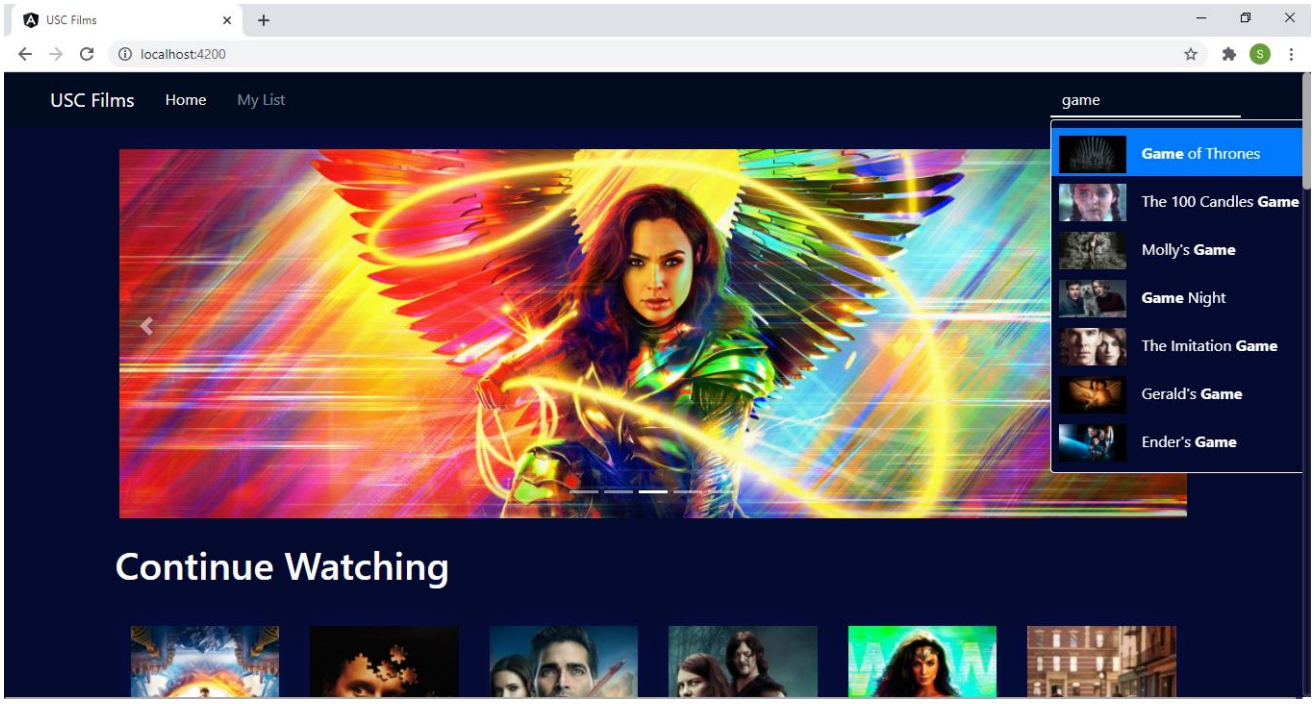



**Figure 1.2:** Autocomplete Suggestions

## 3.2 Home Page

### ~~3.2.1 Currently Playing Movies Carousel~~

- Refer to **section 4.1.4** showing sample TMDB API calls to get Currently Playing Movies related information
- Collect Top 5 results from JSON result and create a carousel consisting of 5 slides. It has auto-rotate time of 5 seconds
- Auto rotate will pause on Hover and on Focus
- Refer to **Section 5.3** for Implementation Hints for creating carousel using **ng-bootstrap**
- Each slide contains poster image of the movie and upon hovering should produce the required zoom effect and the name of the movie should appear at the bottom of the slide
- Selecting a particular slide, will take you to details page of that movie
- Refer **Figure 2.1**

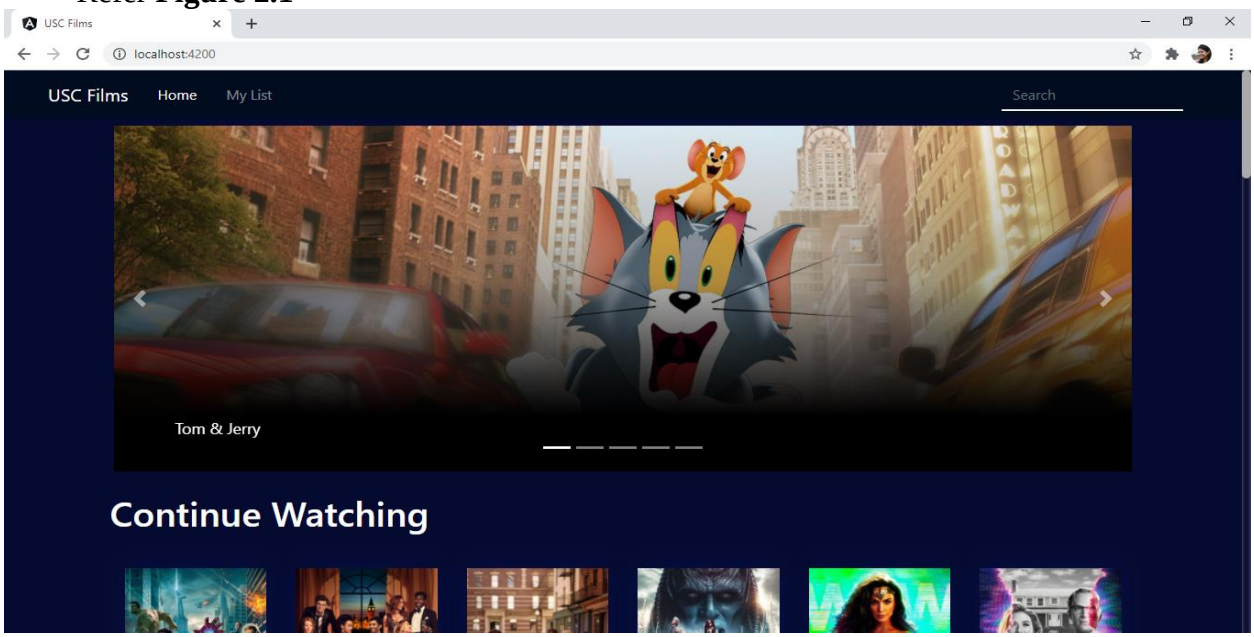



**Figure 2.1:** Currently Playing Movies Carousel

### 3.2.2 Continue Watching Section

- When the user opens Home page for the **first time**, there will be **no continue watching section**.
- Once the user searches for any movie or TV show and selects, they will visit the details page of that movie or TV show. That movie or TV show gets added to the beginning of Continue Watching section's carousel.
- This List will be maintained in **local storage** of the application. For more details on local storage, see **section 5.4**.

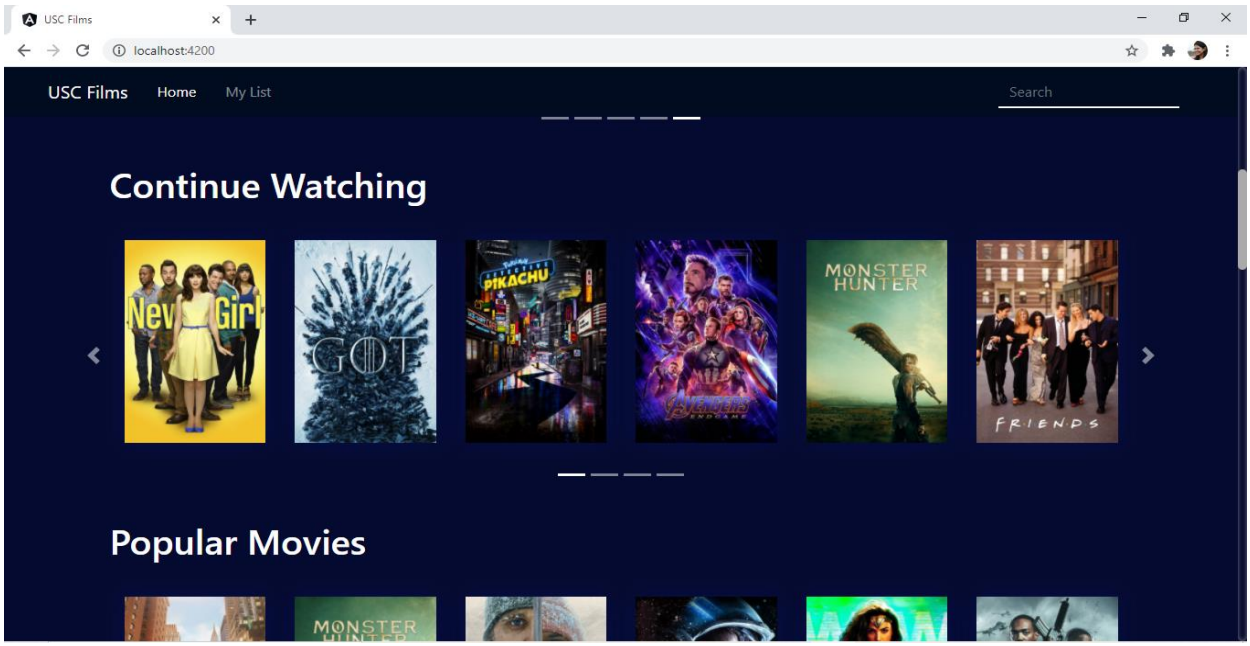



**Figure 2.2:** Continue watching

### 3.2.3 Popular Movies Section

- Popular Movies carousel displays a list of Popular Movies from Popular Movies endpoint. Refer to **section 4.1.5** for TMDB example. Following is the template of API call: https://api.themoviedb.org/3/movie/popular?api_key=<<api_key>>&language=en-US&page=1

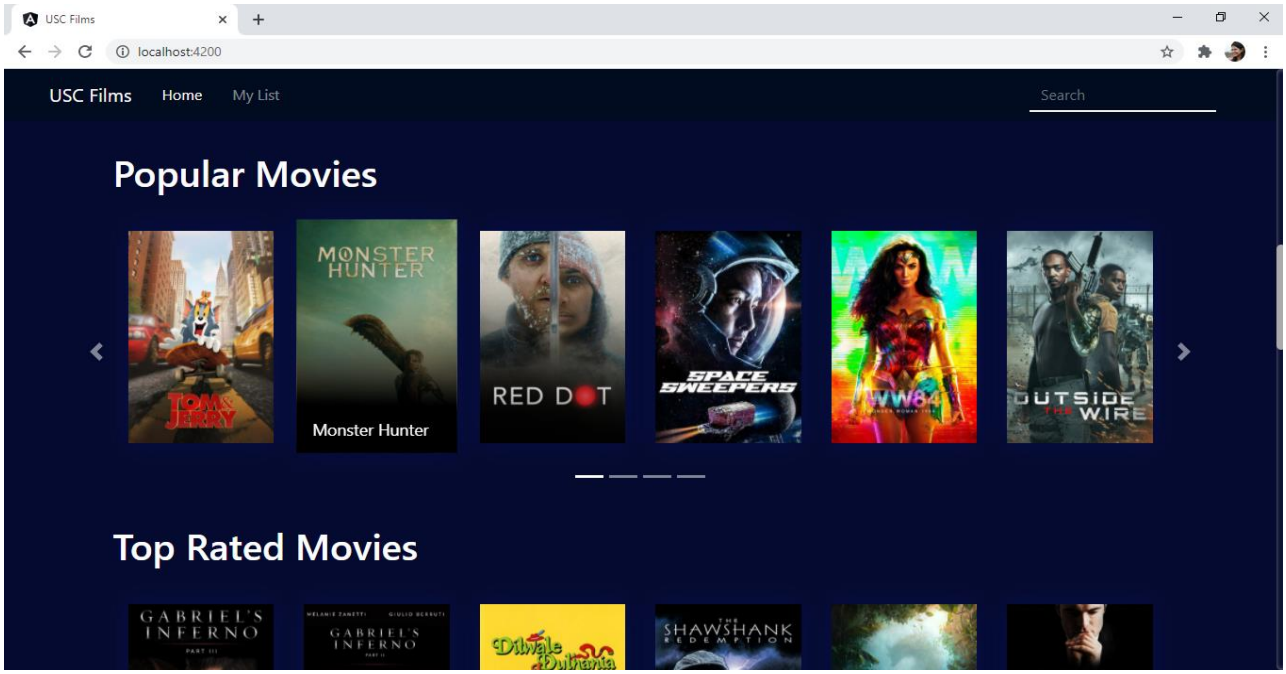



**Figure 2.3:** Popular Movies

### 3.2.4 Top Rated Movies Section

Top Rated Movies carousel displays a list of Top-Rated Movies from Top-Rated Movies endpoint. Refer to **section 4.1.3** for TMDB example. Following is the template of API call:**https://api.themoviedb.org/3/movie/top_rated?api_key=<<api_key>>&language=en-US&page=1**

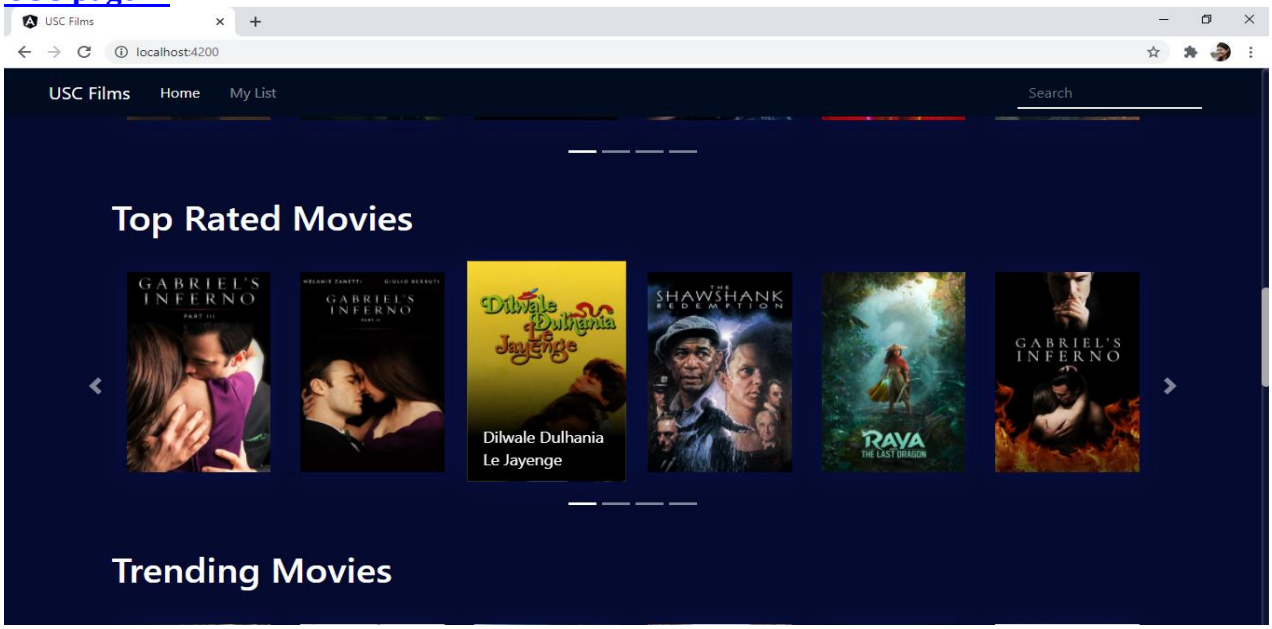



**Figure 2.4:** Top Rated Movies

### 3.2.5 Trending Movies Section

Trending Movies carousel displays a list of Trending Movies from Trending Movies endpoint. Refer to **section 4.1.2** for TMDB example. Following is the template of API call:

https://api.themoviedb.org/3/trending/movie/day?api_key=<<api_key>>

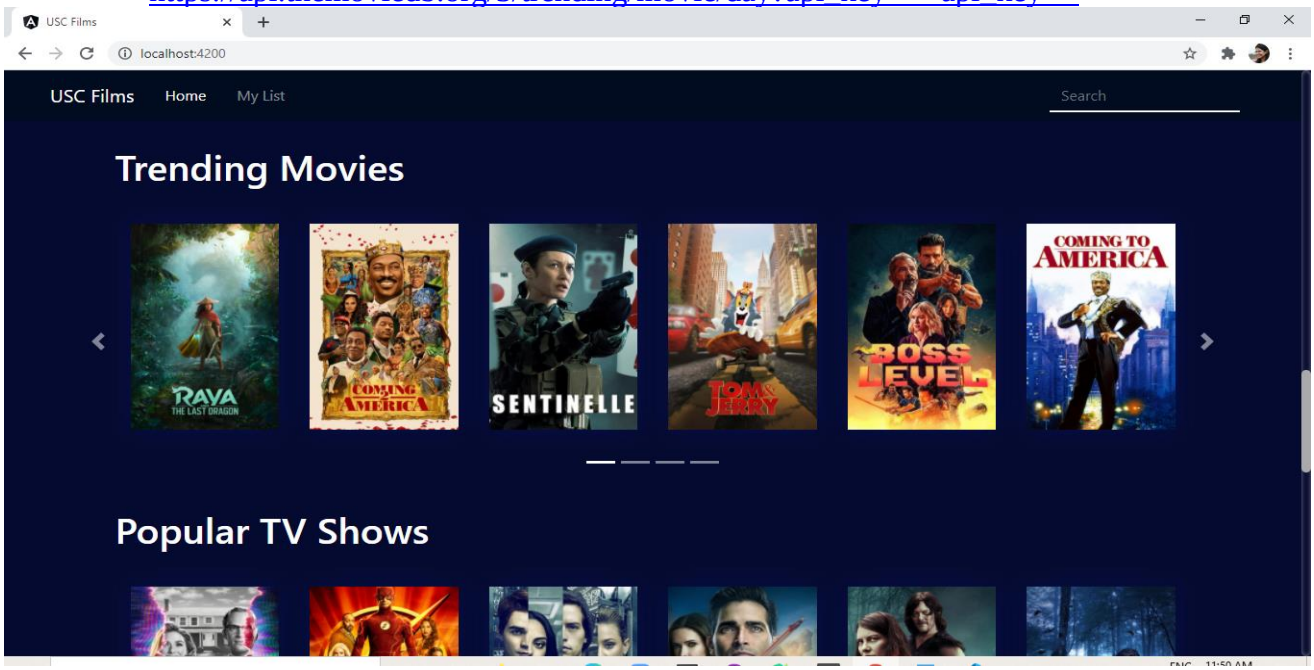



**Figure 2.5:** Trending Movies

### 3.2.6 Popular TV Shows Section

Popular TV shows carousel displays a list of Popular TV shows. Refer to **section 4.1.14** for TMDB API example. Following is the template of API call:
**https://api.themoviedb.org/3/tv/popular?api_key=<<api_key>>&language=en-US&page=1**

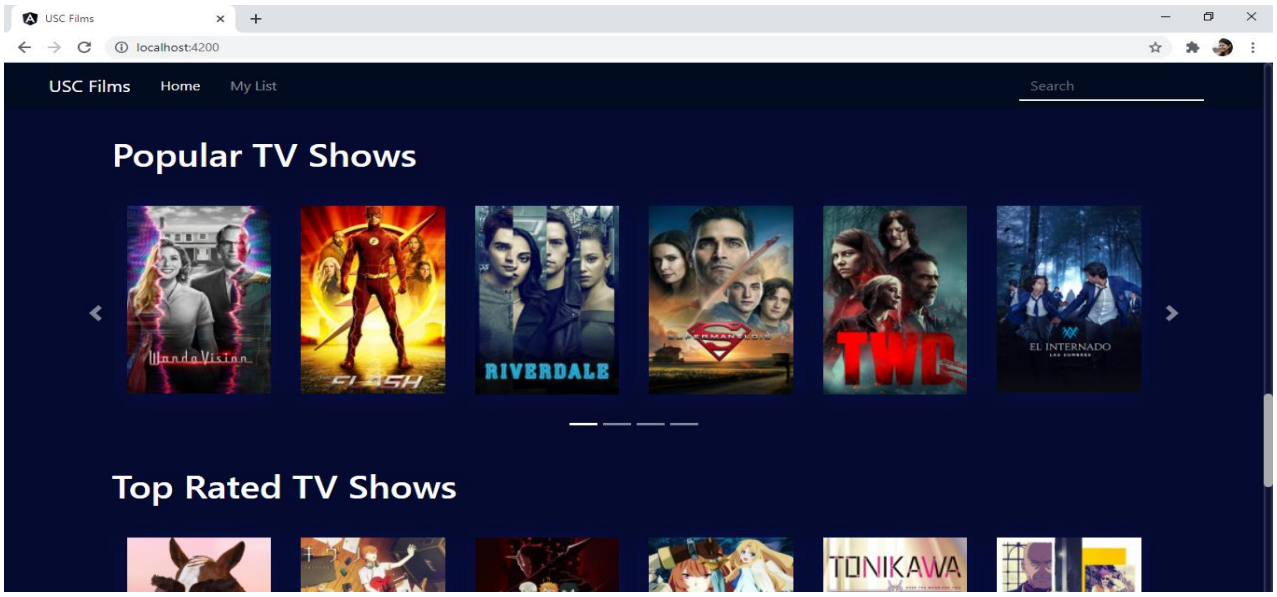



**Figure 2.6:** Popular TV shows

### 3.2.7 Top Rated TV Shows Section

Top Rated TV Shows carousel displays a list of highest Rated TV shows. Refer to **section 4.1.13** for TMDB example. Following is the template of API call:
**https://api.themoviedb.org/3/tv/top_rated?api_key=<<api_key>>&language=en-US&page=1**

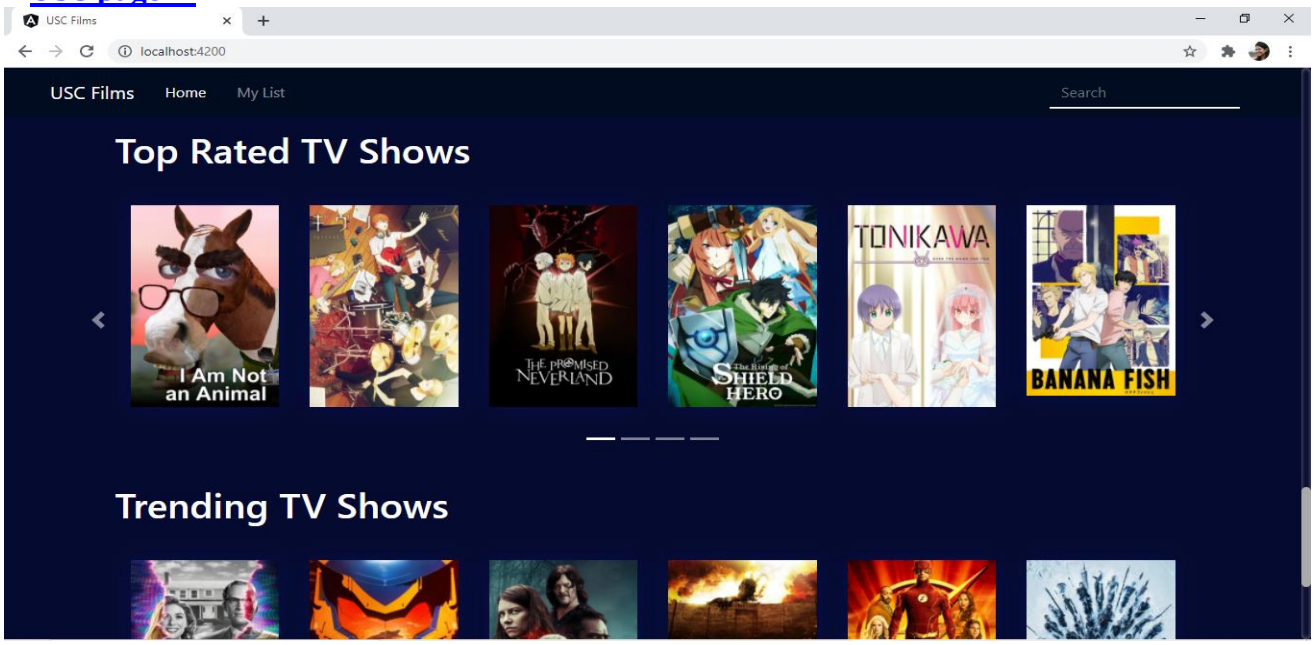



**Figure 2.7:** Top Rated TV shows

### 3.2.8 Trending TV shows Section

Trending TV shows carousel displays a list of Trending TV shows. Refer to **section 4.1.12** for TMDB example. Following is the template of API call:

https://api.themoviedb.org/3/trending/tv/day?api_key=<<api_key>>

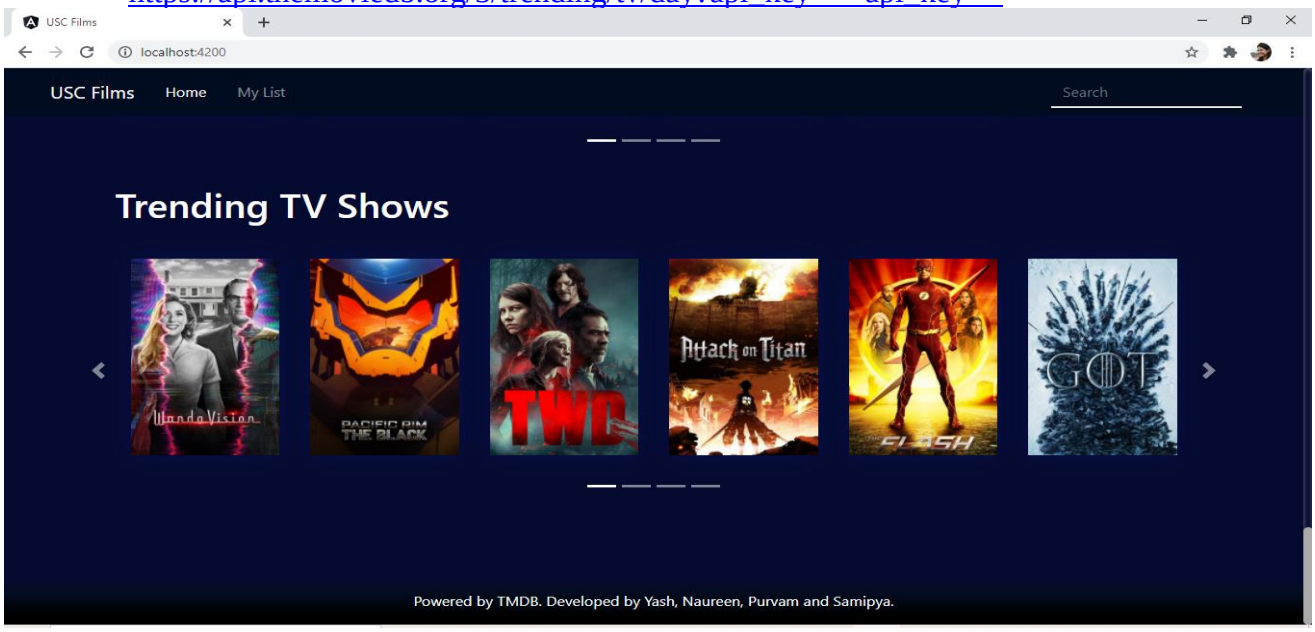



**Figure 2.7:** Trending TV shows

**Styling and Functionality for Sections from 3.2.2 to 3.2.8, 3.3.5 and 3.3.6:**

- Refer to **Section 5.3** for Implementation Hints for creating carousel using **ng-bootstrap**
- Carousel consists of 6 movie cards in a slide with left and right arrows at each end used for sliding to the next/previous set of cards
- Each Movie or TV show card, upon hovering should produce the required zoom effect and the name of TV show or movie at the bottom of the card as shown in **above Figures**.
- Selecting a particular movie's or TV show's card will take you to details page of that movie or Tv show.

## 3.3 Details Page

### 3.3.1 Details of Searched Movie/ TV show

After the user selects a Movie or TV show from autocomplete suggestions dropdown, page should route to /watch/<media_type>/<id> path where <media_type> can be movie or tv. (example: /watch/movie/299534 if selected item's <media_type> is movie or /watch/tv/1668 if selected item's <media_type> is tv).
The following components need to be displayed on successful search:

- Video of trailer (if not present, then teaser) of the movie or TV show is played using Youtube player (**refer section 5.1**). If no video is available, use default video id as **tzkWB85ULJY** https://www.youtube.com/watch?v=tzkWB85ULJY
- Title of the movie or TV show
- Tagline of the movie or TV show

- Release year, Average votes, Duration of the Movie/TV show
- Genres, spoken languages
- Overview
- Full cast and crew
- Reviews
- Recommended and Similar movies for <media_type> movie or Recommended and similar TV shows for <media_type> tv

Please refer to **Figure 3.1** and **Figure 3.2** below.

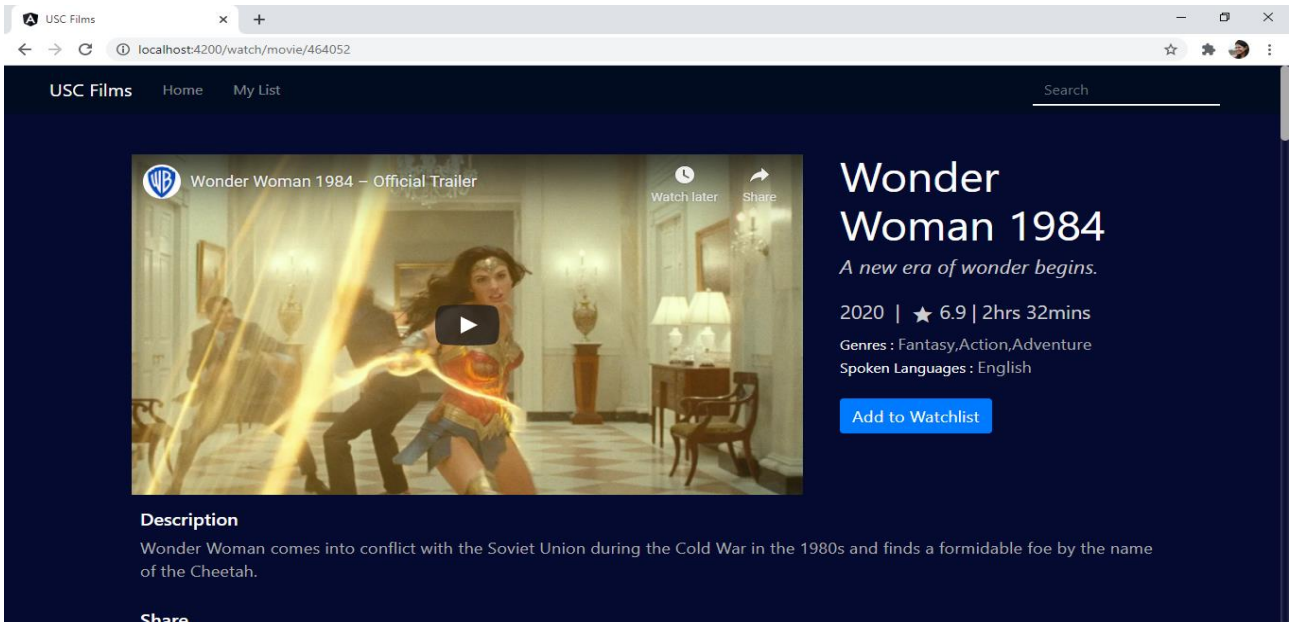



**Figure 3.1:** Details Page

Refer **to sections 4.1.8, 4.1.9, 4.1.17 and 4.1.18 for TMDB API Templates and Examples.**

- There is button “Add to Watchlist”, which when clicked adds that particular movie or TV show at the top of the watchlist. It also produces an alert saying, “Added to watchlist”. Alert closes automatically after **5 seconds or can be manually closed by user.**
- On clicking “Remove from Watchlist” button movie or TV show gets removed from watchlist. After removing from watchlist button changes back to “Add to watchlist”. Alert closes automatically after **5 seconds or can be manually closed by user.**

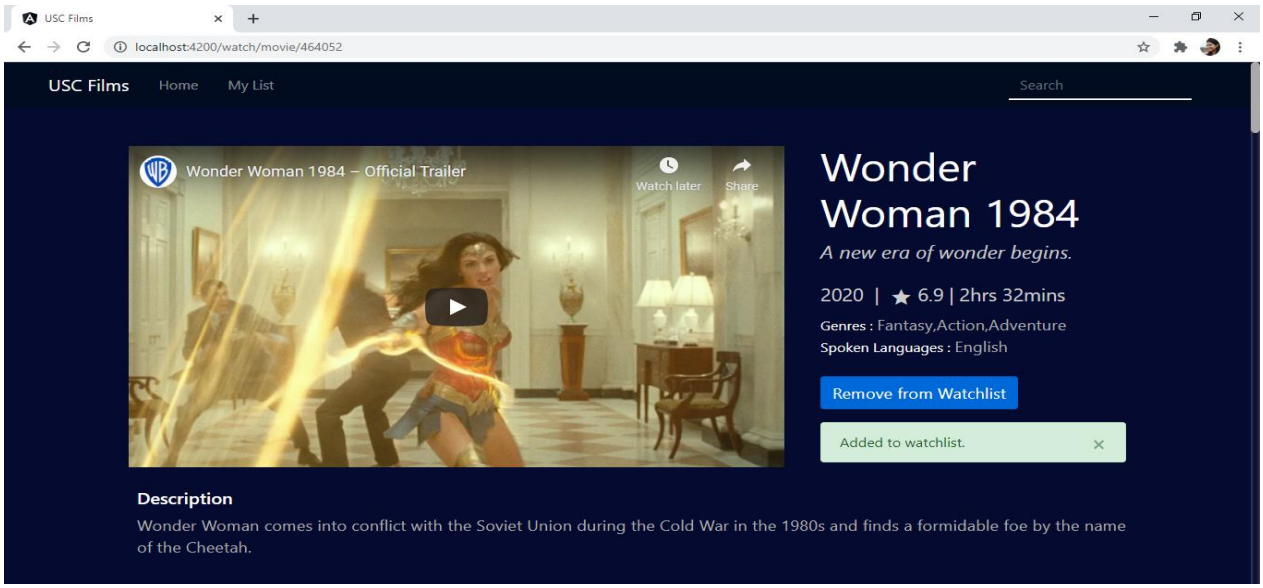



**Figure 3.2:** Added to watchlist Alert

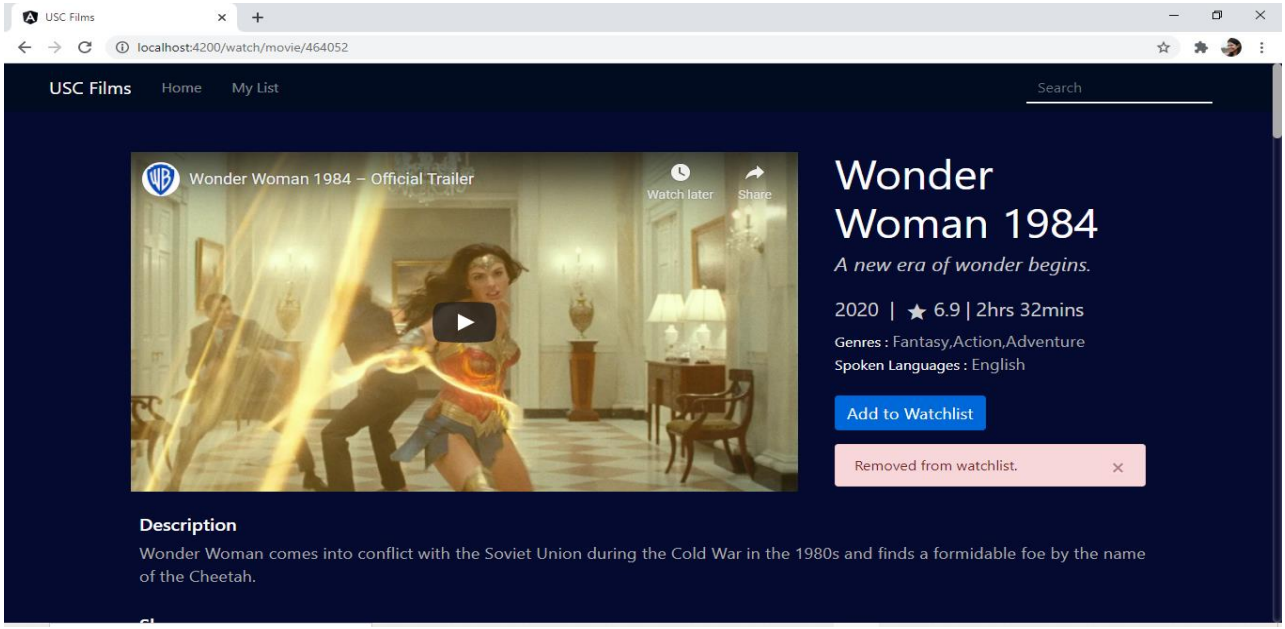



**Figure 3.3:** Removed from watchlist Alert

### 3.3.2 Share Options (Figure 3.4)

- User can share the Video (teaser or trailer links, priority is given to trailer) on Twitter and Facebook. For details on how to use it, refer **Section 4.2**.
- Twitter and Facebook should open in a new browser tab, if clicked.
- In Twitter, it should create a post having following content:
  Watch <Title of the movie/TV show>
  <Youtube link of the video>
  #USC #CSCI571 #FightOn
- In Facebook, it should create a post, which contains Youtube video link.

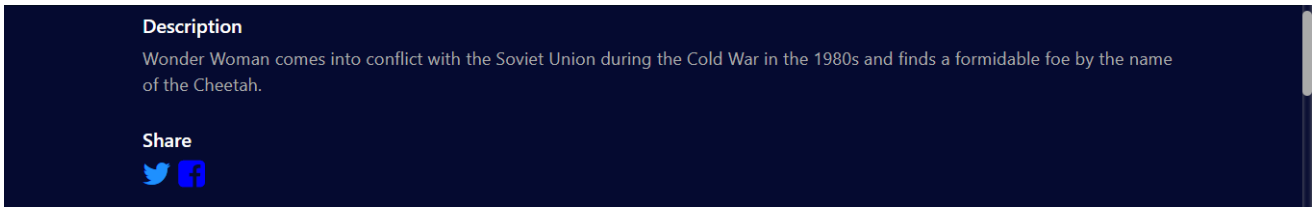



**Figure 3.4:** Share Options

### 3.3.3 Full Cast and Crew section

- Refer to **sections 4.1.11** Movie cast End point and **4.1.20** TV show Cast Endpoint for API Template and Examples
- Each cast member is displayed on a card with their image, name and character name
- There is a scroll bar below the cards, which when scrolled shows the cards for rest of the cast

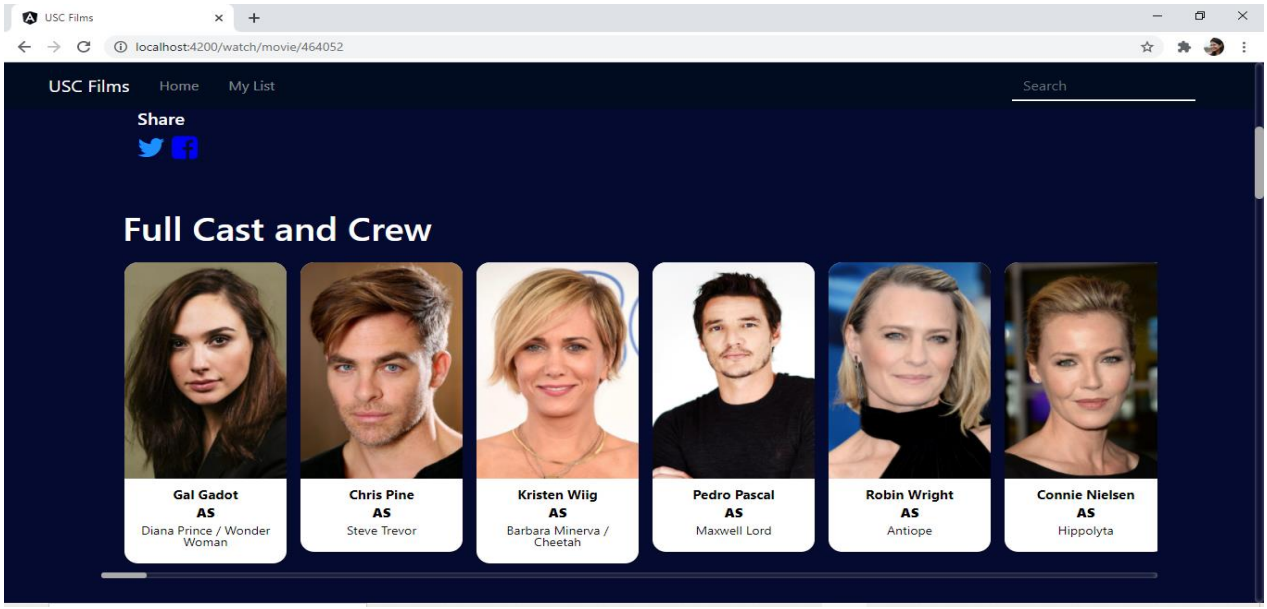



**Figure 3.5:** Cast and Crew

- When clicked on one of the cards, open a Modal window as shown in **Figure 3.6**. For details regarding implementing a **Modal refer section 5.3**.
- Modal contains the following fields:
  - Name of the Cast member
  - Image of the cast member
  - Date of birth
  - Place of birth
  - Gender
  - Known for
  - Other names of the cast member
  - External ids of cast member - upon hovering, a small tooltip appears
    Example: Visit IMDB, Visit Instagram

On clicking, it should take the user to that social media handle of that cast member

- Biography of the cast member

Refer **sections 4.1.21 and 4.1.22** to find example API calls to get external ids and details about cast members.

If any of the required fields are null, they should not appear in the modal. Ignore them.

On clicking the cross symbol or anywhere outside of the modal view, modal should close.


**Figure 3.6:** Modal View on clicking cast member's card

### 3.3.4 Reviews

- Refer to **sections 4.1.10 and 4.1.19** to find the sample API calls to TMDB to get reviews for a particular movie or TV show using their id and API key.
- We display the number of reviews retrieved for the Movie or TV show. A maximum of 10 reviews should be displayed
- Each review will contain author's name, profile picture, rating given, written date, time and content of the review
- "Read the rest" text will open the full review in new tab
- If the content of review is more than 3 lines, it should end with ellipsis the end of 3rd line with "…"

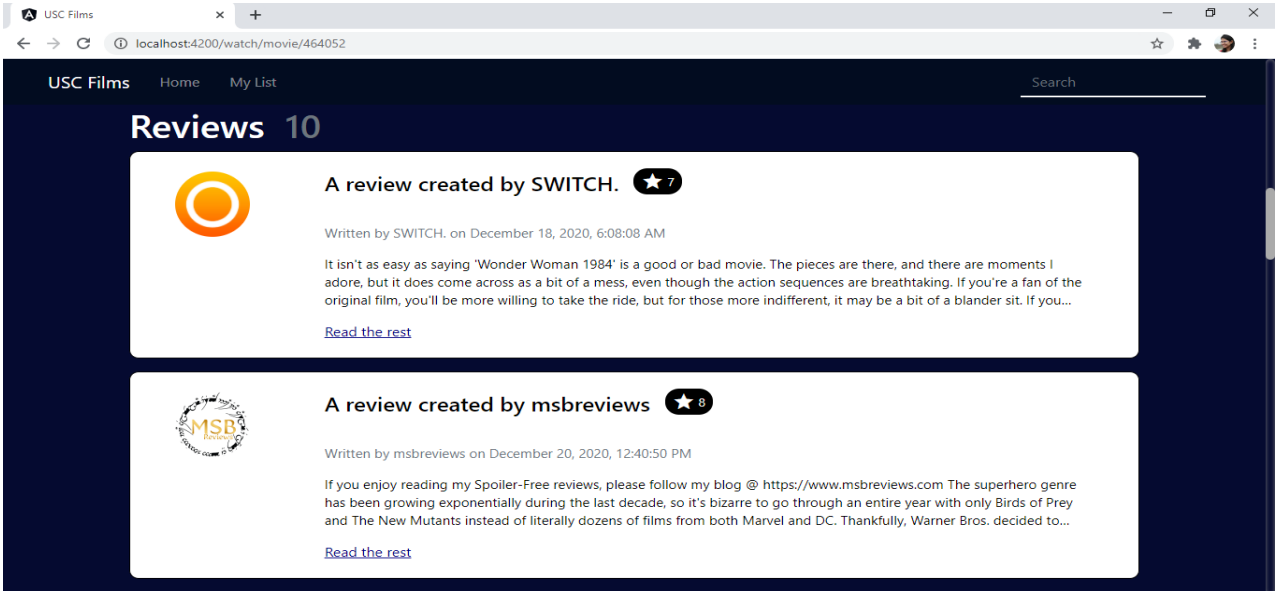



**Figure 3.7:** Reviews

### 3.3.5 Recommended Movies/ TV shows

- Refer to **sections 4.1.6 and 4.1.15** to find the example TMDB API calls to get recommended movies/TV shows
- We give recommended movies or TV shows depending on the media_type of the item displayed on details page
-

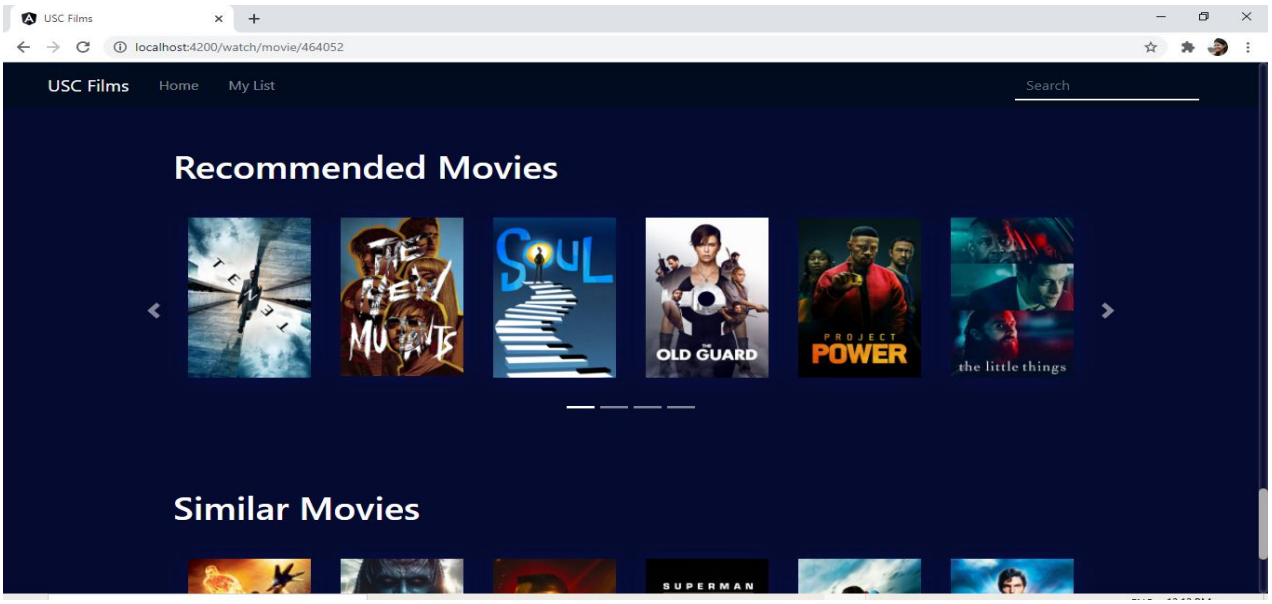



**Figure 3.8:** Recommended Movies/ TV Shows

### 3.3.6 Similar Movies/TV shows

- Refer to **sections 4.1.7 and 4.1.16** to find the example TMDB API calls to get recommended movies/TV shows

- We give similar movies or TV shows depending on the media_type of the item displayed on details page

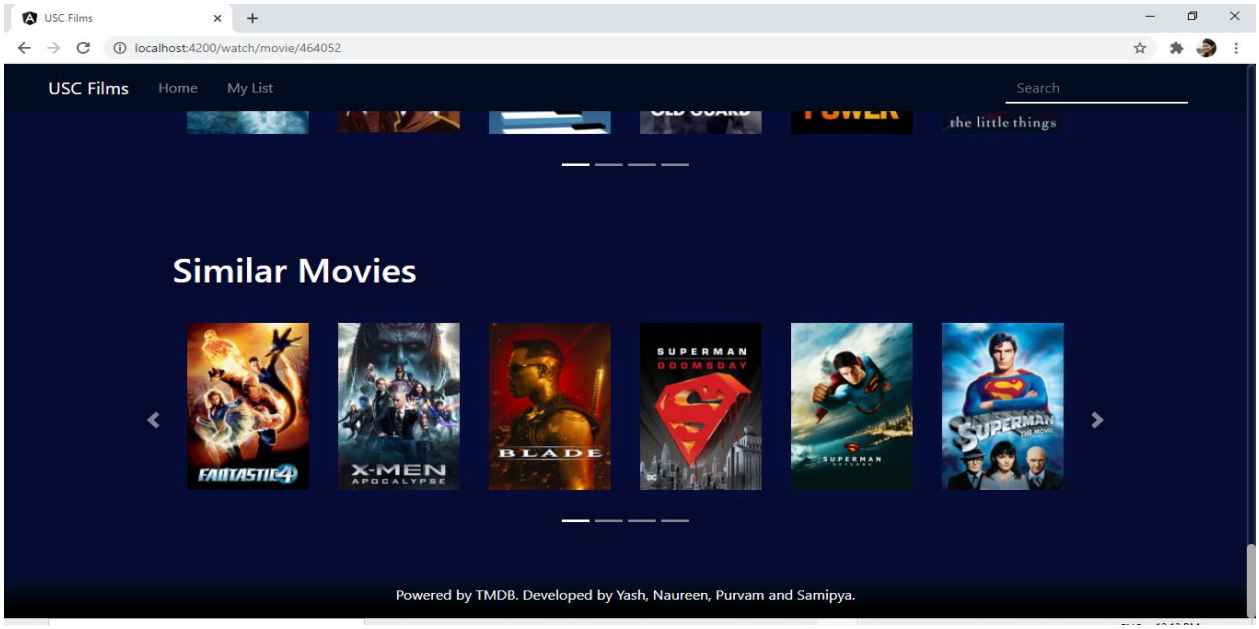



**Figure 3.9:** Similar Movies/TV shows

## 3.4 Watchlist Menu

This menu will display all the Movies and TV shows that are added to the watchlist by the user. This watchlist will be maintained in local storage of the application. For more details on **local storage, see section 5.4**.

- Each Movie or TV show is displayed in the form of a card
- Upon hovering, it produces zooming effect and name of the Movie or TV show should appear at the bottom of the card
- Most Recently added Movie or TV show should be added at the beginning of the list. Once the Movie or TV show which is in the watchlist is visited again, it shifts to the beginning of the watchlist
- On clicking Movie or TV show card, it should take user to the details page of that Movie or TV show
- If watchlist is empty, it should display the alert as shown in **Figure 3.11**.

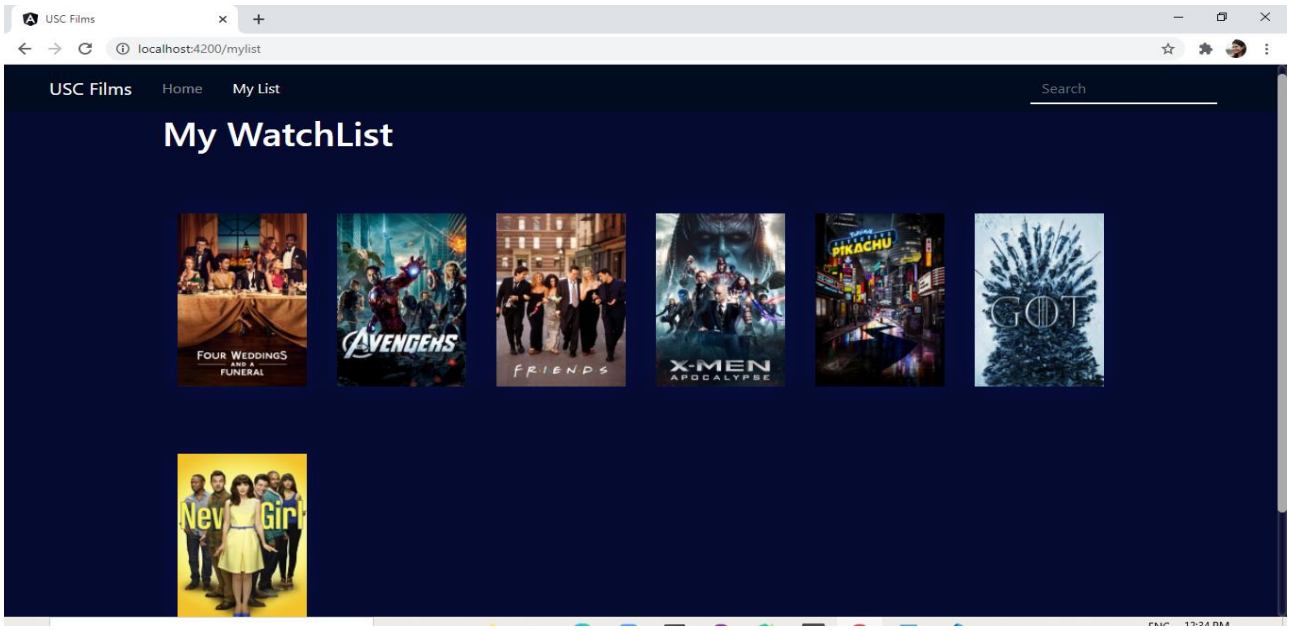



**Figure 3.10:** Watchlist menu page


**Figure 3.11:** Watchlist Empty Alert

## 3.5 Custom Scroll Bar

There is customized scroll bar for the whole web application.

## 3.6 Footer

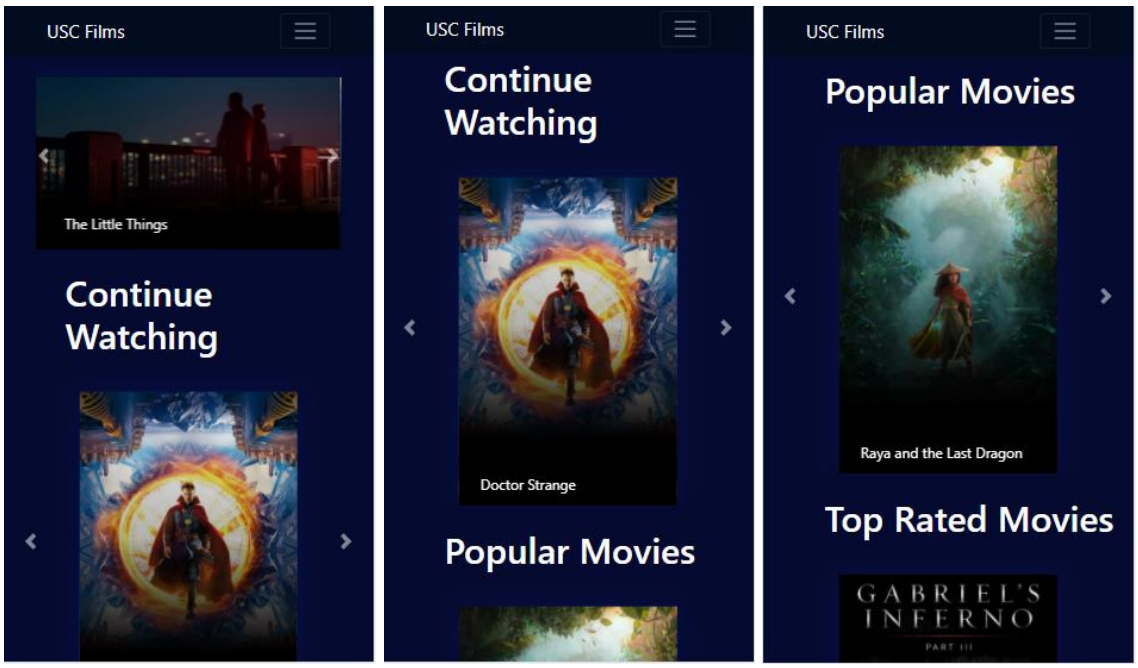
The Footer must be present at the end of each page, as shown in above figures. It should contain following line: **"Powered by TMDB. Developed by <student's name>"**. Here <student's name> must be replaced with your name.

## 3.7 Responsive Design

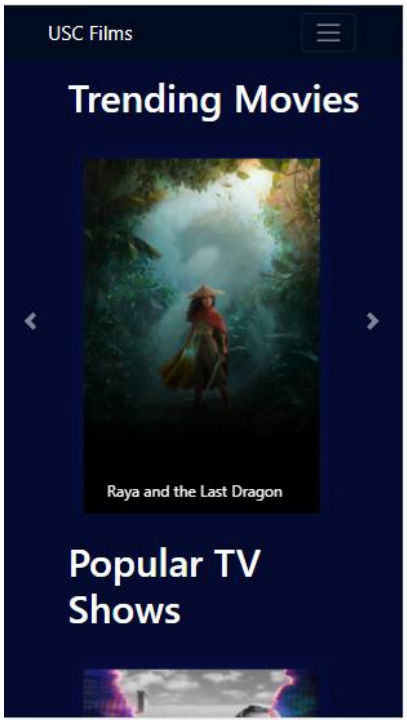
You must watch the video carefully to see how each page looks like on mobile devices. All functions must work on mobile devices like iphone 6/7/9 plus.

**Please note that for Sections from 3.2.2 to 3.2.8, 3.3.5, 3.3.6, the indicators at bottom of carousel should be disabled and the Hover effect should be enabled by default in mobile view.**

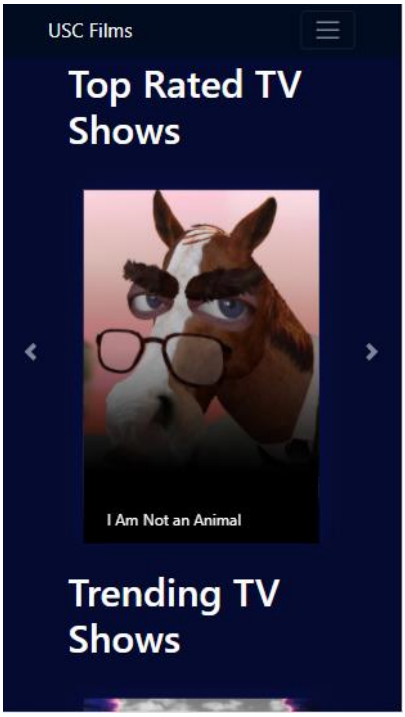
**One easy way to test for mobile devices is to use Google Chrome Responsive Design Mode**

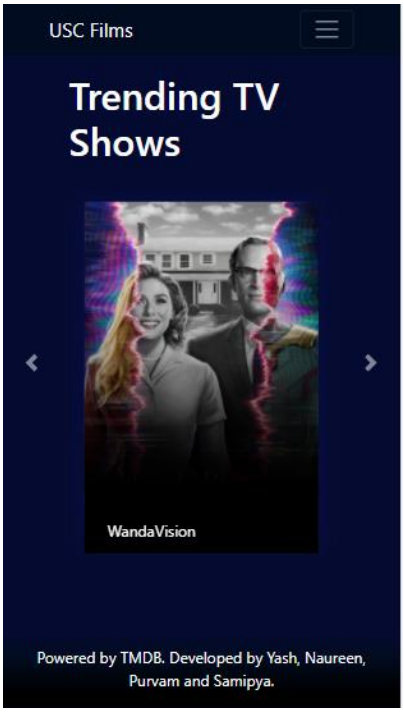


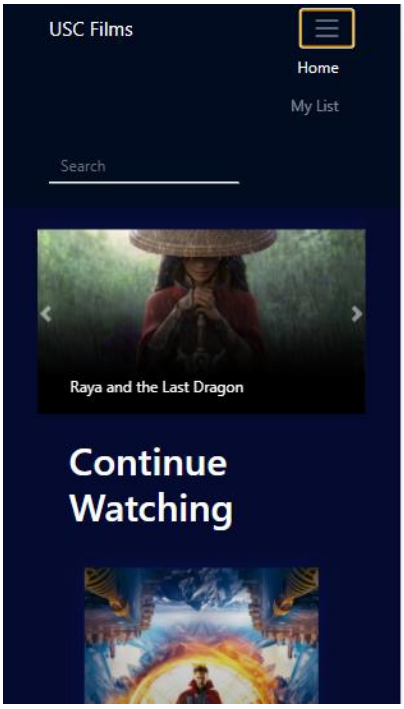

USC Films
Top Rated Movies
GABRIEL'S INFERNO
PART III
Gabriel's Inferno Part III
Trending Movies

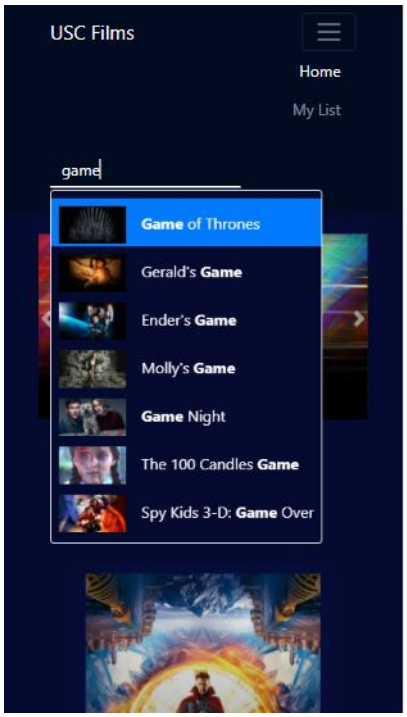

USC Films
Trending Movies
Raya and the Last Dragon
Popular TV Shows


USC Films
Popular TV Shows
WandaVision
Top Rated TV Shows

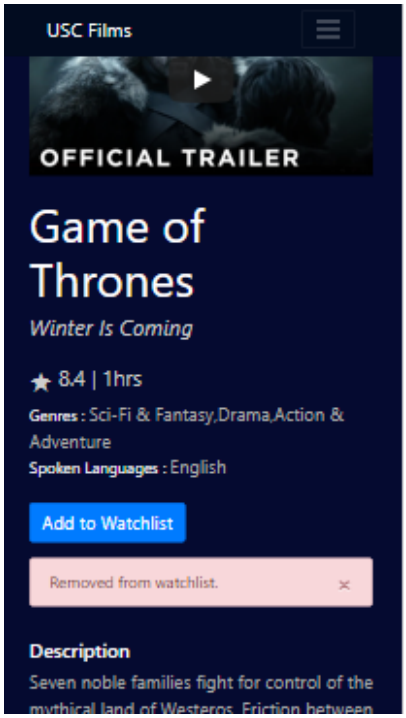

USC Films
Top Rated TV Shows
I Am Not an Animal
Trending TV Shows

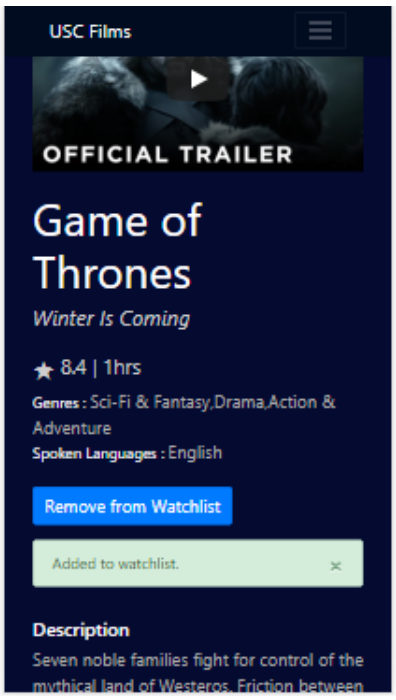

USC Films
Trending TV Shows
WandaVision
Powered by TMDB. Developed by Yash, Naureen, Purvam and Samipya.

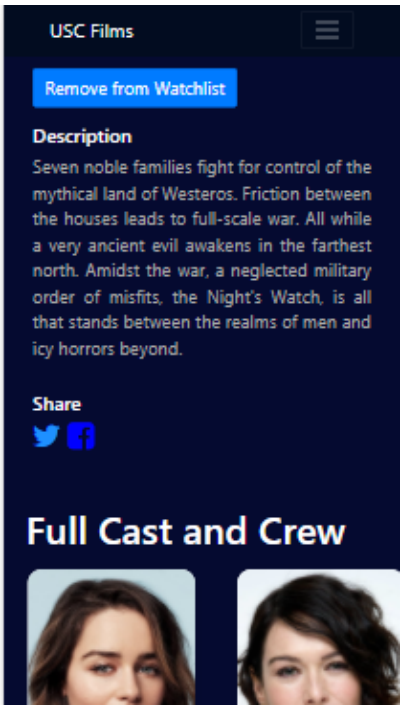

USC Films
Home
My List
Search
Raya and the Last Dragon
Continue Watching

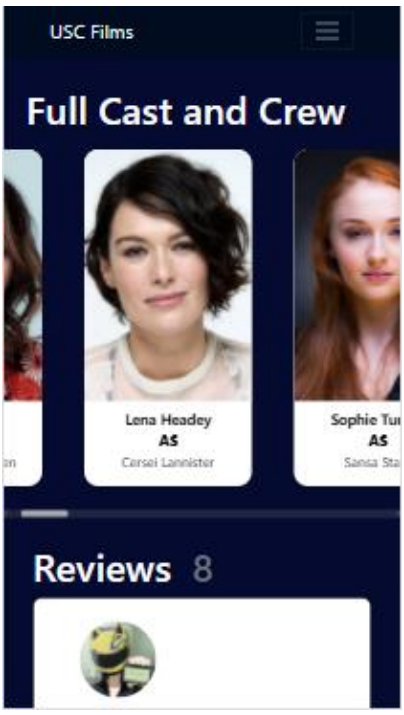

USC Films
Home
My List
game
Game of Thrones
Gerald's Game
Ender's Game
Molly's Game
Game Night
The 100 Candles Game
Spy Kids 3-D: Game Over

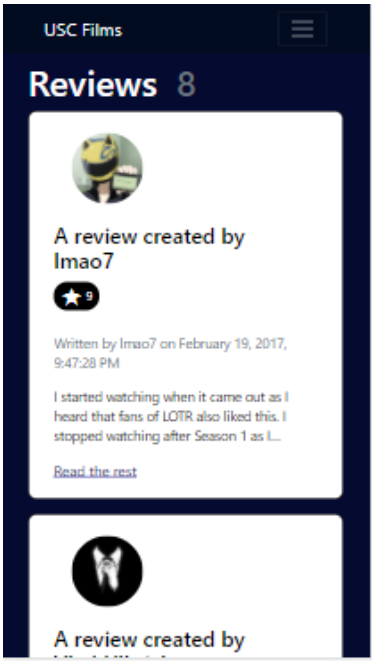

USC Films
GAME OF THRONES - SEAS
OFFICIAL TRAILER
Game of Thrones
Winter Is Coming
8.4 | 1hrs
Genres : Sci-Fi & Fantasy,Drama,Action & Adventure
Spoken Languages : English
Remove from Watchlist
Description
Seven noble families fight for control of the


USC Films
OFFICIAL TRAILER
Game of Thrones
Winter Is Coming
8.4 | 1hrs
Genres : Sci-Fi & Fantasy,Drama,Action & Adventure
Spoken Languages : English
Add to Watchlist
Removed from watchlist.
Description
Seven noble families fight for control of the

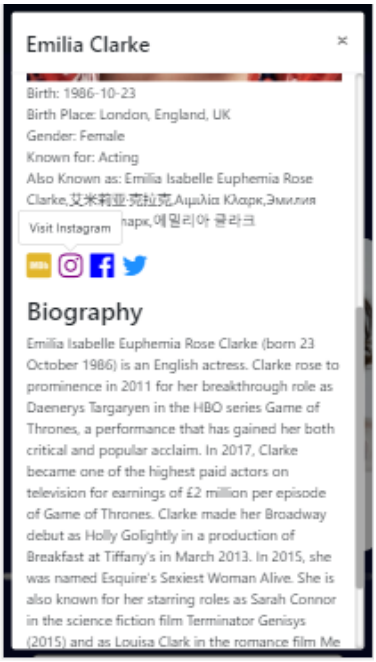

USC Films
OFFICIAL TRAILER
Game of Thrones
Winter Is Coming
8.4 | 1hrs
Genres : Sci-Fi & Fantasy,Drama,Action & Adventure
Spoken Languages : English
Remove from Watchlist
Added to watchlist.
Description
Seven noble families fight for control of the

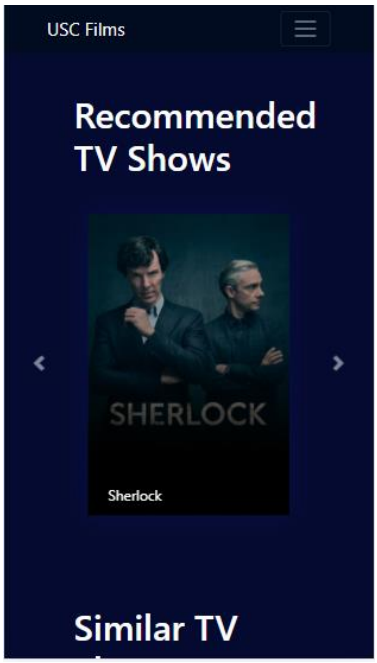

USC Films
Remove from Watchlist
Description
Seven noble families fight for control of the mythical land of Westeros. Friction between the houses leads to full-scale war. All while a very ancient evil awakens in the farthest north. Amidst the war, a neglected military order of misfits, the Night's Watch, is all that stands between the realms of men and icy horrors beyond.
Share
Full Cast and Crew


USC Films
Full Cast and Crew
Lena Headey
AS
Cersei Lannister
Reviews 8

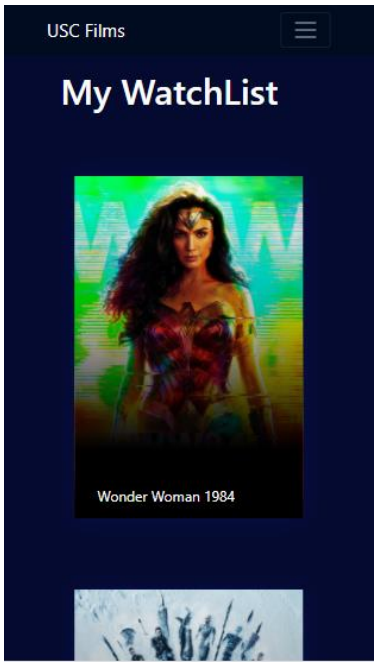

USC Films
Reviews 8
A review created by lmao7
Written by lmao7 on February 19, 2017, 9:47:28 PM
I started watching when it came out as I heard that fans of LOTR also liked this. I stopped watching after Season 1 as I...
Read the rest
A review created by

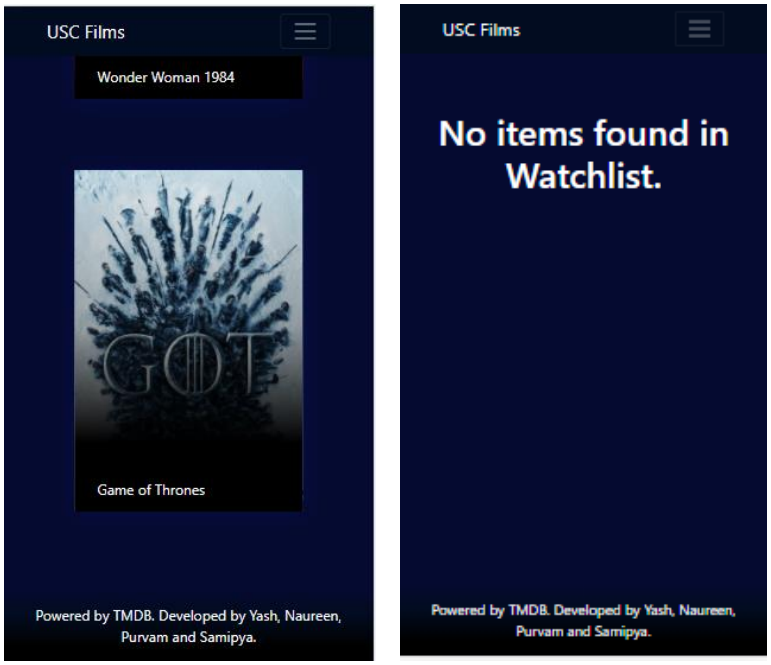

Emilia Clarke
Birth: 1986-10-23
Birth Place: London, England, UK
Gender: Female
Known for: Acting


Emilia Clarke
Birth: 1986-10-23
Birth Place: London, England, UK
Gender: Female
Known for: Acting
Also Known as: Emilia Isabelle Euphemia Rose
Visit Instagram
Biography
Emilia Isabelle Euphemia Rose Clarke (born 23 October 1986) is an English actress. Clarke rose to prominence in 2011 for her breakthrough role as Daenerys Targaryen in the HBO series Game of Thrones, a performance that has gained her both critical and popular acclaim. In 2017, Clarke became one of the highest paid actors on television for earnings of £2 million per episode of Game of Thrones. Clarke made her Broadway debut as Holly Golightly in a production of Breakfast at Tiffany's in March 2013. In 2015, she was named Esquire's Sexiest Woman Alive. She is also known for her starring roles as Sarah Connor in the science fiction film Terminator Genisys (2015) and as Louisa Clark in the romance film Me


USC Films
Recommended TV Shows
SHERLOCK
Sherlock
Similar TV


USC Films
Similar TV Shows
EARTHSEA
Legend of Earthsea
Powered by TMDB. Developed by Yash, Naureen, Purvam and Samipya.


USC Films
My WatchList
Wonder Woman 1984




# 4. API's description

## 4.1 TMDB API calls similar to Homework 6

In this homework, we will use TMDB API.

### ~~4.1.1 Multi-Search Endpoint to search for both Movies and TV shows~~

From this endpoint, you will get a lot of information about the shows, movies, and people that are related to your search query. For each object that is returned, find out if it is a movie, show, or a person through the **'media_type'** property returned for each object. If it is a movie or a tv show, collect it, otherwise, ignore it.

**API Template:**
https://api.themoviedb.org/3/search/multi?api_key=<<api_key>>&language=enUS&query=<<search_query>>

**URL parameters to be provided in API Call:**

- query: search query
- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.

**API Example:**
https://api.themoviedb.org/3/search/multi?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&query=game

For each TV show, you will only need these fields:

- id - the ID of the TV Show
- name - the name of the TV Show
- backdrop_path – You need to construct image url like this: https://image.tmdb.org/t/p/w500 + backdrop_path
- Media_type – tv

For each Movie, you will only need these fields:

- id - the ID of the Movie
- name - the name of the Movie
- backdrop_path – You need to construct image url like this: https://image.tmdb.org/t/p/w500 + backdrop_path
- Media_type - movie


```
// https://api.themoviedb.org/3/search/multi?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1&query=game

{
  "page": 1,
  "results": [
    {
      "backdrop_path": "/suopoADq0k8YZr4dQXcU6pToj6s.jpg",
      "first_air_date": "2011-04-17",
      "genre_ids": [
        10765,
        18,
        10759
      ],
      "id": 1399,
      "media_type": "tv",
      "name": "Game of Thrones",
      "origin_country": [
        "US"
      ],
      "original_language": "en",
      "original_name": "Game of Thrones",
      "overview": "Seven noble families fight for control of the mythical land of Westeros. Friction between the houses leads to full-scale war. All while a very ancient evil awakens in the farthest north. Amidst the war, a neglected military order of misfits, the Night's Watch, is all that stands between the realms of men and icy horrors beyond.",
      "popularity": 391.095,
      "poster_path": "/u3bZgnGQ9T01sWNhyveQz0wH0Hl.jpg",

    {
      "adult": false,
      "backdrop_path": "/1MJC1iOVPnmikmTTeE7EX3nDM5Z.jpg",
      "genre_ids": [
        27
      ],
      "id": 751394,
      "media_type": "movie",
      "original_language": "en",
      "original_title": "The 100 Candles Game",
      "overview": "A group of friends must confront their fears in a terrifying game. They must sit by the other players in a circle made of a hundred candles, take one of them and tell a horror story. As stories are told and candles blown out, strange events will start to happen. They will feel strange presences around them, lurking in the shadows. But they MUST NOT leave the game or else a terrible curse will fall upon them...",
      "popularity": 38.486,
      "poster_path": "/9qpZrNVruotM2oC1QVvjd3fSfK5.jpg",
      "release_date": "2020-11-13",
      "title": "The 100 Candles Game",
      "video": false,
      "vote_average": 6.1,
      "vote_count": 43
    },
```


**Figure 4.1:** The JSON response received from the endpoint for the query “game

### ~~4.1.2 Trending Movies Endpoint~~

This endpoint will be used to get information about trending movies

**API Template:**
https://api.themoviedb.org/3/trending/movie/day?api_key=<<api_key>>

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.

**API Example:**
https://api.themoviedb.org/3/trending/movie/day?api_key=97588ddc4a26e3091152aa0c9a40de22

You will only need these fields:

- id - the ID of the Movie
- title - the name of the Movie
- poster_path – You need to construct image url like this: https://image.tmdb.org/t/p/w500 + poster_path


```
// https://api.themoviedb.org/3/trending/movie/day?api_key=97588ddc4a26e3091152aa0c9a40de22

{
  "page": 1,
  "results": [
    {
      "original_title": "Raya and the Last Dragon",
      "poster_path": "/lPsD10PP4rgUGiGR4CCXA6iY0QQ.jpg",
      "video": false,
      "vote_average": 8.5,
      "overview": "Long ago, in the fantasy world of Kumandra, humans and dragons lived together in harmony. But when an evil force
threatened the land, the dragons sacrificed themselves to save humanity. Now, 500 years later, that same evil has returned and it's up
to a lone warrior, Raya, to track down the legendary last dragon to restore the fractured land and its divided people.",
      "release_date": "2021-03-03",
      "vote_count": 141,
      "id": 527774,
      "backdrop_path": "/7prYzufdIOy1KCTZKVWpjBFqqNr.jpg",
      "title": "Raya and the Last Dragon",
      "genre_ids": [
        16,
        12,
        14,
        10751
      ],
      "adult": false,
      "original_language": "en",
```


**Figure 4.2: JSON returned for Trending Movies Endpoint**

### 4.1.3 Top-Rated Movies Endpoint

This endpoint will be used to get information about highest rated movies

**API Template:**

https://api.themoviedb.org/3/movie/top_rated?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.

**API Example:**

https://api.themoviedb.org/3/movie/top_rated?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

You will only need these fields:

- id - the ID of the Movie
- title - the name of the Movie
- poster_path – You need to construct image url like this: https://image.tmdb.org/t/p/w500 + poster_path


```
// https://api.themoviedb.org/3/movie/top_rated?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

{
  "page": 1,
  "results": [
    {
      "adult": false,
      "backdrop_path": "/fQq1FWp1rC89xDrRMuyFJdFUdMd.jpg",
      "genre_ids": [
        10749,
        35
      ],
      "id": 761053,
      "original_language": "en",
      "original_title": "Gabriel's Inferno Part III",
      "overview": "The final part of the film adaption of the erotic romance novel Gabriel's Inferno written by an anonymous Canadian author under the pen name Sylvain Reynard.",
      "popularity": 31.555,
      "poster_path": "/fYtHxTxlhzD4QWfEbrC1rypysSD.jpg",
      "release_date": "2020-11-19",
      "title": "Gabriel's Inferno Part III",
      "video": false,
      "vote_average": 8.8,
      "vote_count": 734
    },
    {
```


**Figure 4.3: JSON returned for Top_Rated Movies Endpoint**

### ~~4.1.4 Currently playing Movies Endpoint~~

This endpoint will be used to get information about currently playing movies

**API Template:**

https://api.themoviedb.org/3/movie/now_playing?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.

**API Example:**

https://api.themoviedb.org/3/movie/now_playing?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

You will only need these fields:

- id - ID of the Movie
- title - name of the Movie
- poster_path – You need to construct image url like this: https://image.tmdb.org/t/p/w500 + ~~poster_path~~. Ignore those movies which don't have ~~poster_path~~. backdrop_path


**Figure 4.4: JSON returned for Latest Movies Endpoint**

### ~~4.1.5 Popular Movies Endpoint~~

This endpoint will be used to get information about popular movies

**API Template:**

https://api.themoviedb.org/3/movie/popular?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.

**API Example:**

https://api.themoviedb.org/3/movie/popular?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

You will only need these fields:

- id - the ID of the Movie
- title - the name of the Movie
- poster_path – You need to construct image url like this: https://image.tmdb.org/t/p/w500 + poster_path


```
// https://api.themoviedb.org/3/movie/popular?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

{
  "page": 1,
  "results": [
    {
      "adult": false,
      "backdrop_path": "/fev8UFNFFYsD5q7AcYS8LyTzqwl.jpg",
      "genre_ids": [
        28,
        35,
        10751,
        16,
        12
      ],
      "id": 587807,
      "original_language": "en",
      "original_title": "Tom & Jerry",
      "overview": "Tom the cat and Jerry the mouse get kicked out of their home and relocate to a fancy New York hotel, where a scrappy employee named Kayla will lose her job if she can't evict Jerry before a high-class wedding at the hotel. Her solution? Hiring Tom to get rid of the pesky mouse.",
      "popularity": 4502.599,
      "poster_path": "/6KErczPBROQty7QoIsaa6wJYXZi.jpg",
      "release_date": "2021-02-11",
      "title": "Tom & Jerry",
      "video": false,
```


**Figure 4.4: JSON returned for Popular Movies Endpoint**

### 4.1.6 Recommended Movies Endpoint

This endpoint will be used to get information about recommended movies

**API Template:**

https://api.themoviedb.org/3/movie/<<movie_id>>/recommendations?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.
- Movie_id: id of the movie

**API Example:**

https://api.themoviedb.org/3/movie/464052/recommendations?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

You will only need these fields:

- id - the ID of the Movie
- title - the name of the Movie
- poster_path – You need to construct image url like this:
  https://image.tmdb.org/t/p/w500 + poster_path

```
// https://api.themoviedb.org/3/movie/464052/recommendations?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

{
  "page": 1,
  "results": [
    {
      "video": false,
      "vote_average": 7.3,
      "overview": "Armed with only one word - Tenet - and fighting for the survival of the entire world, the Protagonist journeys
through a twilight world of international espionage on a mission that will unfold in something beyond real time.",
      "release_date": "2020-08-22",
      "adult": false,
      "backdrop_path": "/wzJRB4MKi3yK138bJyuL9nx47y6.jpg",
      "vote_count": 4578,
      "genre_ids": [
        28,
        53,
        878
      ],
      "id": 577922,
      "original_language": "en",
      "original_title": "Tenet",
      "poster_path": "/k68nPLbIST6NP96JmTxmZijEvCA.jpg",
      "title": "Tenet",
      "popularity": 539.091
    },
    {
```

**Figure 4.5: JSON returned for Recommended Movies Endpoint**

### ~~4.1.7 Similar Movies Endpoint~~

This endpoint will be used to get information about movies similar to the selected movie

**API Template:**

https://api.themoviedb.org/3/movie/<<movie_id>>/similar?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.
- Movie_id: id of the movie

**API Example:**

https://api.themoviedb.org/3/movie/464052/similar?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

You will only need these fields:

- id - the ID of the Movie

- title - the name of the Movie
- poster_path – You need to construct image url like this:
  https://image.tmdb.org/t/p/w500 + poster_path


```
// https://api.themoviedb.org/3/movie/464052/similar?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

{
  "page": 1,
  "results": [
    {
      "adult": false,
      "backdrop_path": "/isDbx0OEFgYKevzbgZwBJZmQgbM.jpg",
      "genre_ids": [
        28,
        12,
        14,
        878
      ],
      "original_language": "en",
      "original_title": "Fantastic Four",
      "poster_path": "/8HLQLILZLhDQWO6JDpvY6XJLH75.jpg",
      "id": 9738,
      "video": false,
      "vote_average": 5.8,
      "vote_count": 7272,
      "overview": "During a space voyage, four scientists are altered by cosmic rays: Reed Richards gains the ability to stretch his body; Sue Storm can become invisible; Johnny Storm controls fire; and Ben Grimm is turned into a super-strong … thing. Together, these \"Fantastic Four\" must now thwart the evil plans of Dr. Doom and save the world from certain destruction.",
      "release_date": "2005-06-29",
      "title": "Fantastic Four",
      "popularity": 29.23
```


**Figure 4.6: JSON returned for Similar Movies Endpoint**

### ~~4.1.8 Movies Video Endpoint~~

This endpoint will be used to get video related to the selected movie

**API Template:**
https://api.themoviedb.org/3/movie/<<movie_id>>/videos?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.
- Movie_id: id of the movie

**API Example:**
https://api.themoviedb.org/3/movie/464052/videos?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

```
// https://api.themoviedb.org/3/movie/464052/videos?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

{
  "id": 464052,
  "results": [
    {
      "id": "5ded686932489b0016c2760a",
      "iso_639_1": "en",
      "iso_3166_1": "US",
      "key": "sfM7_JLk-84",
      "name": "Official Trailer",
      "site": "YouTube",
      "size": 1080,
      "type": "Trailer"
    },
    {
      "id": "5f41589c028f1400315a70b8",
      "iso_639_1": "en",
      "iso_3166_1": "US",
      "key": "XW2E2Fnh52w",
      "name": "Wonder Woman 1984 - Official Main Trailer",
      "site": "YouTube",
      "size": 1080,
      "type": "Trailer"
    },
    {
      "id": "5f45fe1dcee4810036a9893d",
```


**Figure 4.7: JSON returned for Movies Video Endpoint**

You will only need these fields:

- site – the website where video is available
- type – the type of video available
- name – the caption for video
- key – You need to construct video link like this:
  "https://www.youtube.com/watch?v=" +key

### ~~4.1.9 Movie Details Endpoint~~

This endpoint will be used to get information about Movie details

**API Template:**
https://api.themoviedb.org/3/movie/<<movie_id>>?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.
- Movie_id: id of the movie

**API Example:**
https://api.themoviedb.org/3/movie/464052?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

```
// https://api.themoviedb.org/3/movie/464052?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

{
  "adult": false,
  "backdrop_path": "/srYya1ZlI97Au4jUYAktDe3avyA.jpg",
  "belongs_to_collection": {
    "id": 468552,
    "name": "Wonder Woman Collection",
    "poster_path": "/8AQRfTuTHeFTddZN4IUAqprN8Od.jpg",
    "backdrop_path": "/n9KlvCOBFDmSyw3BgNrkUkxMFva.jpg"
  },
  "budget": 200000000,
  "genres": [
    {
      "id": 14,
      "name": "Fantasy"
    },
    {
      "id": 28,
      "name": "Action"
    },
    {
      "id": 12,
      "name": "Adventure"
    }
  ],
      "name": "DC Comics",
      "origin_country": "US"
    }
  ],
  "production_countries": [
    {
      "iso_3166_1": "US",
      "name": "United States of America"
    }
  ],
  "release_date": "2020-12-16",
  "revenue": 159533000,
  "runtime": 152,
  "spoken_languages": [
    {
      "english_name": "English",
      "iso_639_1": "en",
      "name": "English"
    }
  ],
  "status": "Released",
  "tagline": "A new era of wonder begins.",
  "title": "Wonder Woman 1984",
  "video": false,
  "vote_average": 6.9,
  "vote_count": 4067
```


**Figure 4.8: JSON returned for Movie Details Endpoint**

You will only need these fields:

- title – title of the movie
- genres – movie's genres
- spoken_languages – Languages in which movie is available
- release_date – release date of the movie
- runtime – duration of movie
- overview – movie's synopsis
- vote_average – movie rating votes
- tagline – movie's tag line

### ~~4.1.10 Movie Reviews Endpoint~~

This endpoint will be used to get reviews related to the selected movie

**API Template:**

https://api.themoviedb.org/3/movie/<<movie_id>>/reviews?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.
- Movie_id: id of the movie

**API Example:**

https://api.themoviedb.org/3/movie/464052/reviews?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1


```
{
  "id": 464052,
  "page": 1,
  "results": [
    {
      "author": "SWITCH.",
      "author_details": {
        "name": "SWITCH.",
        "username": "maketheSWITCH",
        "avatar_path": "/klZ9hebmc8biG1RC4WmzNFnciJN.jpg",
        "rating": 7.0
      },
      "content": "It isn't as easy as saying 'Wonder Woman 1984' is a good or bad movie. The pieces are there, and there are moments I adore, but it does come across as a bit of a mess, even though the action sequences are breathtaking. If you're a fan of the original film, you'll be more willing to take the ride, but for those more indifferent, it may be a bit of a blander sit. If you can and are planning to watch it, the theatrical experience is the way to go - there is nothing like seeing these stunning sets, fun action scenes and hearing Zimmer's jaw-dropping score like on the big screen.\r\n- Chris dos Santos\r\n\r\nRead Chris' full article...\r\nhttps://www.maketheswitch.com.au/article/review-wonder-woman-1984-a-new-era-of-wonder-occasionally",
      "created_at": "2020-12-18T14:08:08.440Z",
      "id": "5fdcb7c82efe4e0040d7237c",
      "updated_at": "2020-12-20T16:46:29.704Z",
      "url": "https://www.themoviedb.org/review/5fdcb7c82efe4e0040d7237c"
    },
    {
      "author": "msbreviews",
      "author_details": {
        "name": "",
```


**Figure 4.9: JSON returned for Movie Reviews Endpoint**

You will need a list of the following fields:

- author – the author of the review
- content – review's content
- created_at – date at which review was created
- url – link to the review
- rating – rating value if not null, take 0 otherwise
- avatar_path – profile picture of author

When you get the id from avatar_path, you can append https://image.tmdb.org/t/p/original at the beginning of it, otherwise use the link provided.
If avatar_path is not present, use the default image source as "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRHnPmUvFLjjmoYWAbLTEmLLIRCPpV_OgxCVA&usqp=CAU" or download it from the website.

### ~~4.1.11 Movie Cast Endpoint~~

This endpoint will be used to get cast related information of the selected movie

**API Template:**

https://api.themoviedb.org/3/movie/<<movie_id>>/credits?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.
- Movie_id: id of the movie

**API Example:**

https://api.themoviedb.org/3/movie/464052/credits?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1


```
// https://api.themoviedb.org/3/movie/464052/credits?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

{
  "id": 464052,
  "cast": [
    {
      "adult": false,
      "gender": 1,
      "id": 90633,
      "known_for_department": "Acting",
      "name": "Gal Gadot",
      "original_name": "Gal Gadot",
      "popularity": 18.872,
      "profile_path": "/fysvehTvU6bE3JgxaOTRfvQJzJ4.jpg",
      "cast_id": 0,
      "character": "Diana Prince / Wonder Woman",
      "credit_id": "595686e4c3a368382e050da4",
      "order": 0
    },
    {
      "adult": false,
      "gender": 2,
      "id": 62064,
      "known_for_department": "Acting",
      "name": "Chris Pine",
      "original_name": "Chris Pine",
      "popularity": 10.94,
```


**Figure 4.10: JSON returned for Movie Cast Endpoint**

You will need a list of the following fields:

- Id – id of the cast member
- name – name of the cast member
- character – character's name
- profile_path – Construct image url: https://image.tmdb.org/t/p/w500/+ profile_path

If profile picture is not available, don't display those cast members.

### 4.1.12 Trending TV Shows Endpoint

This endpoint will be used to get information about trending TV shows

**API Template:**
https://api.themoviedb.org/3/trending/tv/day?api_key=<<api_key>>

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.

**API Example:**
https://api.themoviedb.org/3/trending/tv/day?api_key=97588ddc4a26e3091152aa0c9a40de22

You will only need these fields:

- id - the ID of the TV show
- name - the name of the TV show
- poster_path – You need to construct image url like this: https://image.tmdb.org/t/p/w500 + poster_path


```
// https://api.themoviedb.org/3/trending/tv/day?api_key=97588ddc4a26e3091152aa0c9a40de22

{
  "page": 1,
  "results": [
    {
      "overview": "Wanda Maximoff and Vision—two super-powered beings living idealized suburban lives—begin to suspect that everything
is not as it seems.",
      "first_air_date": "2021-01-15",
      "id": 85271,
      "vote_count": 6761,
      "genre_ids": [
        10765,
        9648,
        18
      ],
      "original_language": "en",
      "original_name": "WandaVision",
      "origin_country": [
        "US"
      ],
      "name": "WandaVision",
      "poster_path": "/glKDfE6btIRcVB5zrjspRIs4r52.jpg",
      "backdrop_path": "/57vVjteucIF3bGnZj6PmaoJRScw.jpg",
      "vote_average": 8.5,
      "popularity": 3770.795,
      "media_type": "tv"
    }
```


**Figure 4.11: JSON returned for Trending TV shows Endpoint**

### ~~4.1.13 Top-Rated TV shows Endpoint~~

This endpoint will be used to get information about top-rated TV shows

**API Template:**
https://api.themoviedb.org/3/tv/top_rated?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.

**API Example:**

https://api.themoviedb.org/3/tv/top_rated?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

You will only need these fields:

- id - the ID of the TV show
- name - the name of the TV show
- poster_path – You need to construct image url like this: https://image.tmdb.org/t/p/w500 + poster_path


```
// https://api.themoviedb.org/3/tv/top_rated?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

{
  "page": 1,
  "results": [
    {
      "backdrop_path": null,
      "first_air_date": "2004-05-10",
      "genre_ids": [
        16,
        35
      ],
      "id": 100,
      "name": "I Am Not an Animal",
      "origin_country": [
        "GB"
      ],
      "original_language": "en",
      "original_name": "I Am Not an Animal",
      "overview": "I Am Not An Animal is an animated comedy series about the only six talking animals in the world, whose cosseted
existence in a vivisection unit is turned upside down when they are liberated by animal rights activists.",
      "popularity": 11.602,
      "poster_path": "/qG59J1Q7rpBc1dvku4azbzcqo8h.jpg",
      "vote_average": 9.4,
      "vote_count": 599
    },
```


**Figure 4.13: JSON returned for Top_Rated TV shows Endpoint**

### ~~4.1.14 Popular TV shows Endpoint~~

This endpoint will be used to get information about popular TV shows

**API Template:**

https://api.themoviedb.org/3/tv/popular?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.

**API Example:**

**https://api.themoviedb.org/3/tv/popular?api_key=97588ddc4a26e3091152aa0c9a40de2**

**2&language=en-US&page=1**

You will only need these fields:

- id - the ID of the Tv show
- name- the name of the Tv show
- poster_path – You need to construct image url like this: https://image.tmdb.org/t/p/w500 + poster_path


**Figure 4.14: JSON returned for Popular TV shows Endpoint**

### 4.1.15 Recommended TV shows Endpoint

This endpoint will be used to get information about recommended TV shows

**API Template:**

https://api.themoviedb.org/3/tv/<<tvshow_id>>/recommendations?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.
- tvshow_id: id of the TV show

**API Example:**

https://api.themoviedb.org/3/tv/85271/recommendations?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

You will only need these fields:

- id - the ID of the TV show
- name - the name of the TV show
- poster_path – You need to construct image url like this:

https://image.tmdb.org/t/p/w500 + poster_path


```
// https://api.themoviedb.org/3/tv/85271/recommendations?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

{
  "page": 1,
  "results": [
    {
      "backdrop_path": "/iqmJfU6DkqJra9PqIepaOJbzfa1.jpg",
      "first_air_date": "2021-01-08",
      "genre_ids": [
        99
      ],
      "id": 114695,
      "name": "Marvel Studios: Legends",
      "origin_country": [
        "US"
      ],
      "original_language": "en",
      "original_name": "Marvel Studios: Legends",
      "overview": "Revisit the epic heroes, villains and moments from across the MCU in preparation for the stories still to come. Each
dynamic segment feeds directly into the upcoming series – setting the stage for future events. This series weaves together the many
threads that constitute the unparalleled Marvel Cinematic Universe.",
      "poster_path": "/EpDuYIK81YtCUT3gH2JDpyj8Qk.jpg",
      "vote_average": 7.6,
      "vote_count": 307,
      "networks": [
        {
          "id": 2739,
```


**Figure 4.15: JSON returned for Recommended TV shows Endpoint**

### ~~4.1.16 Similar TV shows Endpoint~~

This endpoint will be used to get information about TV shows similar to the selected TV show

**API Template:**

https://api.themoviedb.org/3/tv/<<tvshow_id>>/similar?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.
- tvshow_id: id of the TV show

**API Example:**

https://api.themoviedb.org/3/tv/85271/similar?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

You will only need these fields:

- id - the ID of the Movie
- name - the name of the Movie
- poster_path – You need to construct image url like this:
  https://image.tmdb.org/t/p/w500 + poster_path

```
// https://api.themoviedb.org/3/tv/85271/similar?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

{
  "page": 1,
  "results": [
    {
      "original_name": "Marvel's The Defenders",
      "name": "Marvel's The Defenders",
      "vote_average": 7.2,
      "overview": "Daredevil, Jessica Jones, Luke Cage and Iron Fist join forces to take on common enemies as a sinister conspiracy
threatens New York City.",
      "vote_count": 716,
      "poster_path": "/49XzINhH4LFsgz7cx6TOPcHUJUL.jpg",
      "id": 62285,
      "backdrop_path": "/7eV2vDrj1AwlTffUud66v9o0Ytq.jpg",
      "first_air_date": "2017-08-18",
      "genre_ids": [
        10765,
        10759,
        80
      ],
      "origin_country": [
        "US"
      ],
      "original_language": "en",
      "popularity": 44.325
    },
```


**Figure 4.16: JSON returned for Similar TV shows Endpoint**

### ~~4.1.17 TV show Video Endpoint~~

This endpoint will be used to get video related to the selected TV show

**API Template:**

https://api.themoviedb.org/3/tv/<<tvshow_id>>/videos?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.
- Tvshow_id: id of the TV show

**API Example:**

https://api.themoviedb.org/3/tv/85271/videos?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1


**Figure 4.17: JSON returned for TV show Video Endpoint**

You will only need these fields:

- site – the website where video is available
- type – the type of video available
- name – the caption for video
- key – You need to construct video link like this:
  "https://www.youtube.com/watch?v=" +key

### 4.1.18 TV show Details Endpoint

This endpoint will be used to get information about TV show details

**API Template:**

https://api.themoviedb.org/3/tv/<<tvshow_id>>?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.
- tvshow_id: id of the TV show

**API Example:**

https://api.themoviedb.org/3/tv/85271?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1


**Figure 4.118: JSON returned for TV show Details Endpoint**

You will only need these fields:

- title – title of the TV show
- genres – TV show's genres
- spoken_languages – Languages in which TV show is available
- first_air_date – release date of the TV show
- episode_run_time – duration of TV show's episode
- overview – TV show synopsis
- vote_average – TV show rating votes
- tagline – TV show tag line

### 4.1.19 TV show Reviews Endpoint

This endpoint will be used to get reviews related to the selected TV show

**API Template:**
https://api.themoviedb.org/3/tv/<<tvshow_id>>/reviews?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.
- tvshow_id: id of the TV show

**API Example:**

https://api.themoviedb.org/3/tv/85271/reviews?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1


```
// https://api.themoviedb.org/3/tv/85271/reviews?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

{
  "id": 85271,
  "page": 1,
  "results": [
    {
      "author": "msbreviews",
      "author_details": {
        "name": "",
        "username": "msbreviews",
        "avatar_path": "/https://secure.gravatar.com/avatar/992eef352126a53d7e141bf9e8707576.jpg",
        "rating": null
      },
      "content": "If you enjoy reading my Spoiler-Free reviews, please follow my blog @\r\nhttps://www.msbreviews.com\r\n\r\n(Based on
the season's first three episodes handed to press)\r\n\r\nMarvel delivered an unprecedentedly successful superhero universe that
culminated in the record-breaking Avengers: Endgame (Spider-Man: Far from Home is actually the last movie of the Infinity Saga, but let's
ignore that). With the first overarching story ending after twenty-three films, the MCU now begins a new Era. Basically, Phase Four is
truly another Phase One, where viewers will meet new heroes to root for. However, there are dozens of Avengers still in the game,
including the astonishingly powerful Wanda. \r\n\r\nTelevision will play a massive role in the upcoming saga, with miniseries starring
key superheroes having a significant impact in the movies. Obviously, this raises the question: do viewers need to watch the TV shows in
order to understand and follow the films? I believe so. Some series might not have as much impact as others, but WandaVision will
undoubtedly affect Wanda's path in the MCU. From being one of the most powerful Avengers to the possibility of Wanda being the next big
villain, this character has limitless potential. In addition to this, Elizabeth Olsen is arguably one of the best actresses in the
franchise, and honestly, of her entire generation.\r\n\r\nTherefore, even though I always try to keep my expectations as moderate as
possible, I couldn't help but feel extremely hyped for WandaVision. The series is as weird and mysterious as I expected it to be. In the
first three episodes, the viewer is placed inside a black-and-white sitcom, where it becomes super clear that everything that's happening
```


**Figure 4.19: JSON returned for TV show Reviews Endpoint**

You will need a list of the following fields:

- author – the author of the review
- content – review's content
- created_at – date at which review was created
- url – link to the review
- rating – rating value if not null, take 0 otherwise
- avatar_path – avatar of author

When you get the id from avatar_path, you can append "https://image.tmdb.org/t/p/original" at the beginning of it, otherwise use the link provided.
If avatar_path is not present, use the default image source as "https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRHnPmUvFLjjmoYWAbLTEmLLIRCPpV_OgxCVA&usqp=CAU" or download it from the website.

### ~~4.1.20 TV show Cast Endpoint~~

This endpoint will be used to get cast related information of the selected TV show

**API Template:**

https://api.themoviedb.org/3/tv/<<tvshow_id>>/credits?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.
- tvshow_id: id of the TV show

**API Example:**

https://api.themoviedb.org/3/tv/85271/credits?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1


```
// https://api.themoviedb.org/3/tv/85271/credits?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

{
  "cast": [
    {
      "adult": false,
      "gender": 1,
      "id": 550843,
      "known_for_department": "Acting",
      "name": "Elizabeth Olsen",
      "original_name": "Elizabeth Olsen",
      "popularity": 31.704,
      "profile_path": "/wIU675y4dofIDVuhaNWPizJNtep.jpg",
      "character": "Wanda Maximoff / The Scarlet Witch",
      "credit_id": "5cb05ac50e0a2626c8c41b0d",
      "order": 0
    },
    {
      "adult": false,
      "gender": 2,
      "id": 6162,
      "known_for_department": "Acting",
      "name": "Paul Bettany",
      "original_name": "Paul Bettany",
      "popularity": 14.847,
      "profile_path": "/vcAVrAOZrpqmi37qjFdztRAv1u9.jpg",
      "character": "Vision / The Vision",
```


**Figure 4.20: JSON returned for TV show Cast Endpoint**

You will need a list of the following fields:

- Id – id of the cast member
- name – name of the cast member
- character – character's name
- profile_path – Construct image url:
  https://image.tmdb.org/t/p/w500/+ profile_path

If profile picture is not available, don't display those cast members.

### 4.1.21 Cast Details Endpoint

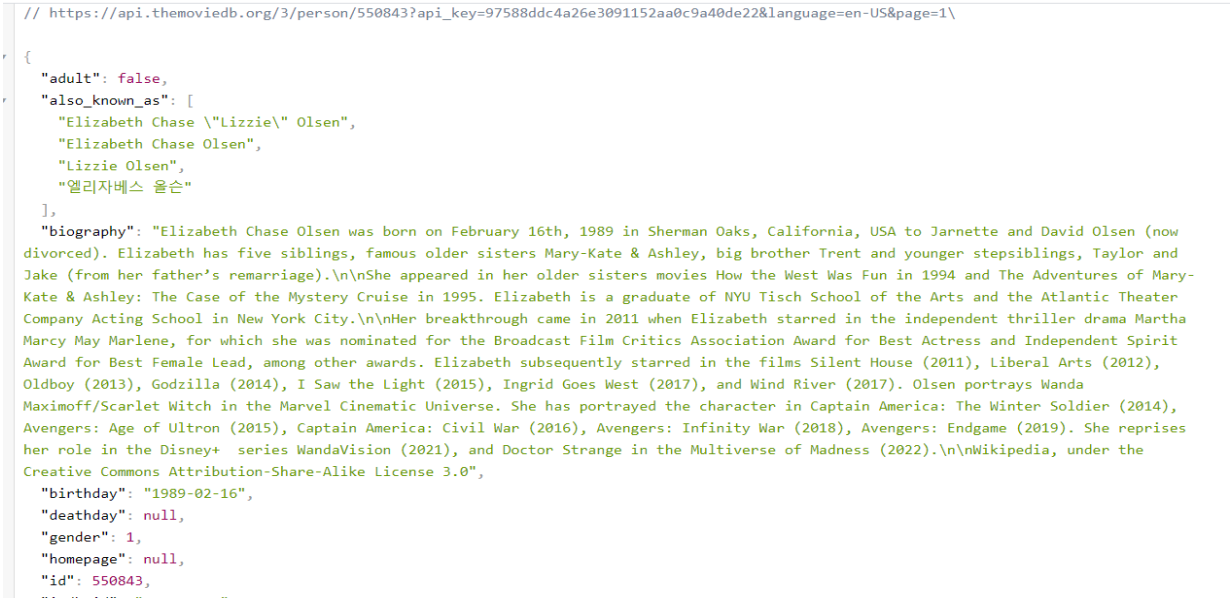
This endpoint will be used to get information related to cast member using person id

**API Sample:**
https://api.themoviedb.org/3/person/<<person_id>>?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.
- person_id: id of the cast member, we can get this using 4.1.11 for movie cast members and 4.1.20 for TV show cast members

**API Example:**
https://api.themoviedb.org/3/person/550843?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1



```
// https://api.themoviedb.org/3/person/550843?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1\

{
  "adult": false,
  "also_known_as": [
    "Elizabeth Chase \"Lizzie\" Olsen",
    "Elizabeth Chase Olsen",
    "Lizzie Olsen",
    "엘리자베스 올슨"
  ],
  "biography": "Elizabeth Chase Olsen was born on February 16th, 1989 in Sherman Oaks, California, USA to Jarnette and David Olsen (now divorced). Elizabeth has five siblings, famous older sisters Mary-Kate & Ashley, big brother Trent and younger stepsiblings, Taylor and Jake (from her father's remarriage).\n\nShe appeared in her older sisters movies How the West Was Fun in 1994 and The Adventures of Mary-Kate & Ashley: The Case of the Mystery Cruise in 1995. Elizabeth is a graduate of NYU Tisch School of the Arts and the Atlantic Theater Company Acting School in New York City.\n\nHer breakthrough came in 2011 when Elizabeth starred in the independent thriller drama Martha Marcy May Marlene, for which she was nominated for the Broadcast Film Critics Association Award for Best Actress and Independent Spirit Award for Best Female Lead, among other awards. Elizabeth subsequently starred in the films Silent House (2011), Liberal Arts (2012), Oldboy (2013), Godzilla (2014), I Saw the Light (2015), Ingrid Goes West (2017), and Wind River (2017). Olsen portrays Wanda Maximoff/Scarlet Witch in the Marvel Cinematic Universe. She has portrayed the character in Captain America: The Winter Soldier (2014), Avengers: Age of Ultron (2015), Captain America: Civil War (2016), Avengers: Infinity War (2018), Avengers: Endgame (2019). She reprises her role in the Disney+  series WandaVision (2021), and Doctor Strange in the Multiverse of Madness (2022).\n\nWikipedia, under the Creative Commons Attribution-Share-Alike License 3.0",
  "birthday": "1989-02-16",
  "deathday": null,
  "gender": 1,
  "homepage": null,
  "id": 550843,
```


**Figure 4.21: JSON returned for Cast Details Endpoint**

You will need a list of the following fields:

- birthday – date of birth of cast member
- gender – 1 for female and 2 for male, undefined if 0
- name – name of the cast member
- homepage – website link for cast member
- also known as – other names of cast members
- known for department – talents
- biography – biography of cast member

### 4.1.22 Cast external ids Endpoint

This endpoint will be used to get information related to external ids of cast member using person id

**API Sample:**
https://api.themoviedb.org/3/person/<<person_id>>/external_ids?api_key=<<api_key>>&language=en-US&page=1

**URL parameter in API Call:**

- api_key: The API access Key. It is private, please do not share with anyone. See Homework 6.
- person_id: id of the cast member, we can get this using 4.1.11 for movie cast members and 4.1.21 for TV show cast members

**API Example:**
https://api.themoviedb.org/3/person/550843/external_ids?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1


```
// https://api.themoviedb.org/3/person/550843/external_ids?api_key=97588ddc4a26e3091152aa0c9a40de22&language=en-US&page=1

{
  "id": 550843,
  "freebase_mid": null,
  "freebase_id": null,
  "imdb_id": "nm0647634",
  "tvrage_id": null,
  "facebook_id": "OfficiallyElizabethOlsen",
  "instagram_id": "",
  "twitter_id": null
}
```


**Figure 4.22: JSON returned for Cast External ids Endpoint**

You will need a list of the following fields:

- imdb_id – imdb page for cast member (imdb.com/name/imdb_id)
- facebook_id – facebook id of cast member (facebook.com/facebook_id)
- Instagram_id – Instagram id if of cast member (instagram.com/Instagram_id)
- Twitter_id – twitter id if of cast member (twitter.com/Twitter_id)

## 4.2 Socials

### 4.2.1 Twitter

Refer the following link for details:

https://developer.twitter.com/en/docs/twitter-for-websites/tweet-button/overview

### 4.2.2 Facebook

Refer the following link for details:
https://developers.facebook.com/docs/plugins/share-button/

## 5. Implementation Hints

## 5.1 Useful Libraries

To get started with the Bootstrap toolkit, please see:
https://getbootstrap.com/docs/4.0/getting-started/introduction/.

Bootstrap can be imported to Angular in couple of ways. See:
https://www.techiediaries.com/angular-bootstrap/

Angular Youtube player:
https://www.npmjs.com/package/@angular/youtube-player

NG Bootstrap:
https://ng-bootstrap.github.io/#/getting-started

## 5.2 Bootstrap UI Components

Bootstrap provides a complete mechanism to make Web pages responsive for different mobile devices. In this exercise, you will get hands-on experience with responsive design using the Bootstrap Grid System.
https://getbootstrap.com/docs/4.0/layout/grid/

At a minimum, you will need to use Bootstrap Forms, Alerts, Cards and Buttons to implement the required functionality.

| | |
|---|---|
| Bootstrap Forms | https://getbootstrap.com/docs/4.0/components/forms/ |
| Bootstrap Alerts | https://getbootstrap.com/docs/4.0/components/alerts/ |
| Bootstrap Cards | https://getbootstrap.com/docs/4.0/components/card/ |
| Bootstrap Buttons | https://getbootstrap.com/docs/4.0/components/buttons/ |

## 5.3 Ng-Bootstrap

- Carousel - https://ng-bootstrap.github.io/#/components/carousel/examples
- Alerts - https://ng-bootstrap.github.io/#/components/alert/examples
- Modal -https://ng-bootstrap.github.io/#/components/modal/examples
- ngbTypeahead - https://ng-bootstrap.github.io/#/components/typeahead/examples
- Tooltip - https://ng-bootstrap.github.io/#/components/tooltip/examples

## 5.4 Local Storage

Refer the following documentation for detailed understanding on Local Storage and how to use it: https://developer.mozilla.org/en-US/docs/Web/API/Window/localStorage

### 5.5 Icons

Use font-awesome icons: https://fontawesome.com/icons?d=gallery&p=2

### 5.6 Deploy Node.js application on Cloud Services

Since Homework #8 is implemented with Node.js on Cloud Services, you should **select Nginx as your proxy server (if available)**, which should be the default option.

## 6. Files to Submit

In your course homework page, you should update the Homework #8 link to refer to your new initial web page for this exercise. Additionally, you need to provide an additional link to the URL of the cloud service where the AJAX call is made with sample parameter values.
Zip the project folder and submit on DEN.

**IMPORTANT**:
All videos are part of the homework description. All discussions and explanations in Piazza related to this homework are part of the homework description and will be accounted into grading. So please review all Piazza threads before finishing the assignment.