

## **MADS4007 FINAL EXAM**

### **(35% of overall course grade)**

### **SUBMISSION REQUIREMENTS**

1. Compress your project files and any related assets (html,js,css,images and audio files) into a zip file
2. Name your zip file: `finalexam-petersmith-10123456.zip`  
Substitute **petersmith** and **10123456** with your student name and id.
3. Upload the zip file to Blackboard by the date specified in the dropbox.

Please note that a submission will not be graded if:

1. The project is missing required files or assets
2. The submission is late
3. The submission contains academic dishonesty.

### **ACADEMIC INTEGRITY**

This assessment must be completed individually.

Providing **any help** to other students is **not permitted**.

Help includes, but is not limited to:

- Exchange of ideas, advice about how to approach the problem
- Sharing website links or reference materials
- Sharing of devices, such as laptops, physical IOS devices, or MacInCloud accounts
- Sharing of credentials to college provided or personal email accounts, Google Drives, Dropbox, Blackboard, or similar

**Communication with individuals either in the class, or outside the class, is strictly prohibited.**

Any learner suspected of breaking the above rules will not have their submission marked; and, a report will be filed with the College's Academic Integrity department.

To prevent issues of academic integrity, it is imperative that you are vigilant that your code is not duplicated, in part or whole, by another learner. Duplication can mean: identical code; or code that is semantically identical in naming and overall construction. Should the instructors discover academic dishonesty as defined above and per the College Academic Integrity Policies, **a grade of 0** will be assigned to all parties involved.

## PROBLEM DESCRIPTION

**Notes for the Student:** This Final Exam is designed to give you practical experience experience in building Client Side web applications that consume Restful APIs

**Background:** You will need to have access to a code editor and have a thorough understanding of HTML5, CSS3, Vanilla JavaScript, Event Driven programming, DOM manipulation and REST APIs

### Overview:

You have been contracted as a Front-End Web Developer to build a two (2) page movie website.

Your website does **not** have to look aesthetically pleasing, i.e, pretty. However, for the website layout, you **MUST** use CSS grids and or Flexbox. Thus, you are not allowed to use any free or paid templates, libraries and/or framework such as Bootstrap, Bulma etc. Failure to follow the above-mentioned rules will result in a 0 grade being assigned.

Your website must consist of the following two web pages :

- A Latest movie page
- A Details movie page

### Final Exam Website Demo:

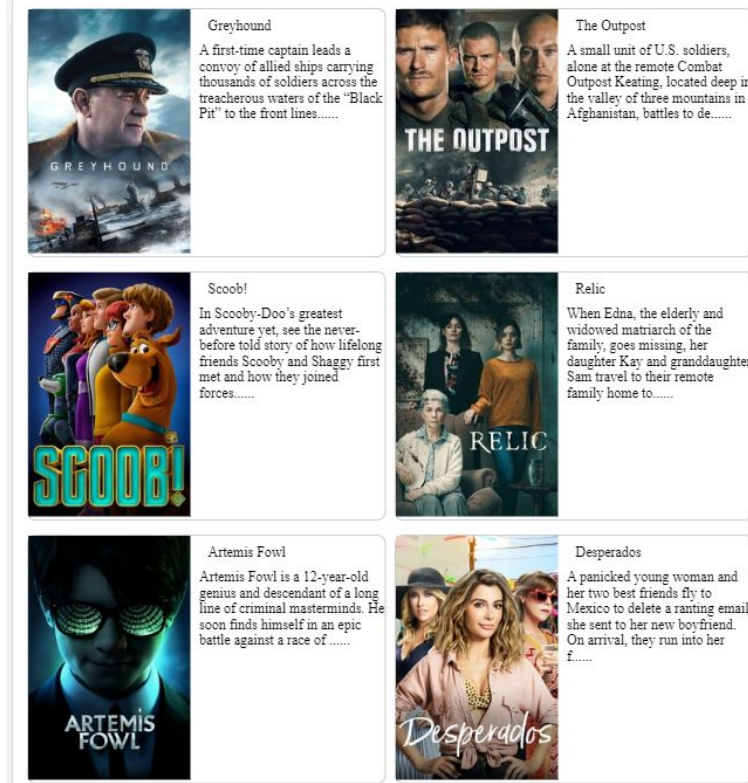
Please watch this [Video](#) of me explaining the features and the required functionality of the website.

### Rules:

The Latest movie page must have the below section:

- **Movie container section** - For this section you **must** provide a list of movies in two column grid format. **The movies must be pulled from the Now Playing API** created by TheMovieDB. Each movie must have a title, movie synopsis, image and the date the movie was released, similarly to the below image in **Figure 1.1**

**Figure 1.1**



The Latest Movie page must have the below functionality:

- Ensure that each movie poster is clickable and when the user clicks on a respective poster, the user is navigated to another webpage, specifically the **Movie details page**.
- The Movie details page **MUST** list the title, synopsis, released date, poster and the trailer of the clicked movie. For Example, if the user clicked on the **Greyhound** poster then the user should be navigated to the Movie details page, detailing the respective information for said movie.
- To obtain the endpoint for the above-mentioned API, navigate to the below URL : <https://developers.themoviedb.org/3/movies/get-now-playing>
- Click the Try it out tab (which is located next to the **Definition** tab)
- See below for example **Figure 1.2** :

**Figure 1.2**

Movies

Get Now Playing

GET /movie/now\_playing

Get a list of movies in theatres. This is a release type query that looks for all movies that have a release type of 2 or 3 within the specified date range.

You can optionally specify a `region` parameter which will narrow the search to only look for theatrical release dates within the specified country.

Definition

Try it out

Variables

api_key	Your TMDb API key	optional
---------	-------------------	----------

Query String

api_key	<<api_key>>	required
language	en-US	optional
page	1	optional
region	String	optional

SEND REQUEST

https://api.themoviedb.org/3/movie/now\_playing?api\_key=<<api\_key>>&language=en-US&page=1

- The endpoint for the Now Playing API is :

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

- Replace `<<api_key>>` with your actual api key.

Below is a sample of the results expected from the above API :

Raw

Parsed

```

{
  "results": [
    {
      "popularity": 351.193,
      "vote_count": 148,
      "video": false,
      "poster_path": "/AsdB9A2XGa1CZVjlyG9tRf03IFw.jpg",
      "id": 516486,
      "adult": false,
      "backdrop_path": "/xXBnM6uSTk6qqCf0SRZKXcga9Ba.jpg",
      "original_language": "en",
      "original_title": "Greyhound",
      "genre_ids": [
        28,
        18,
        10752
      ],
      "title": "Greyhound",
      "vote_average": 7,
      "overview": "A first-time captain leads a convoy of allied ships carrying thousands of soldiers across the treacherous waters of the 'Black Pit' to the front lines of WW2. With no air cover protection for 5 days, the captain and his convoy must battle the surrounding enemy Nazi U-boats in order to give the allies a chance to win the war.",
      "release_date": "2020-07-10"
    },
    {
      "popularity": 117.321,
      "vote_count": 126,
      "video": false,
      "poster_path": "/hPkqY2EMqWnFEoedukilIUieVG.jpg",
      "id": 531876,
      "adult": false,
      "backdrop_path": "/n1RohH2VoK1CdVI2FxcvP19dS1m.jpg",
      "original_language": "en",
      "original_title": "The Guernsey Movie"
    }
  ]
}

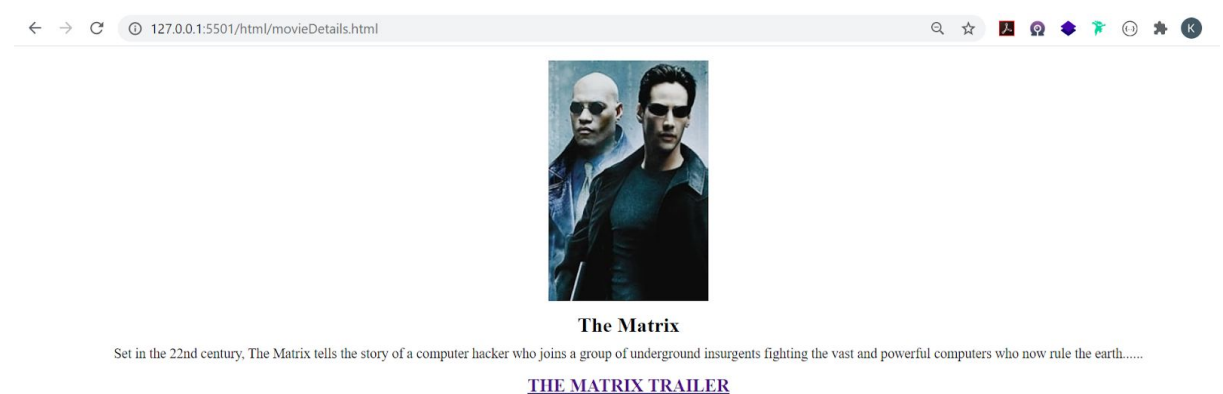
```

- Please note, the poster\_path key from the JSON results does not provide the base path for the image, it just returns the name of the image. Thus, you must concatenate the base path of the URL, i.e., ["https://image.tmdb.org/t/p/w185\\_and\\_h278\\_bestv2"](https://image.tmdb.org/t/p/w185_and_h278_bestv2) to the poster\_path for the image to be rendered.

### The Movie Details Page functionality :

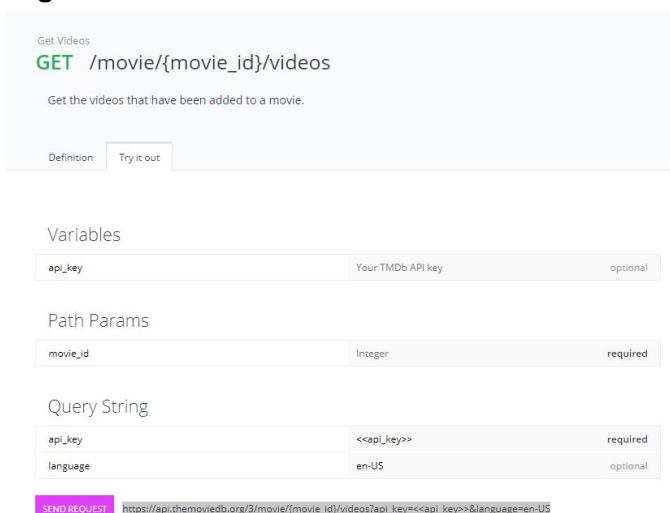
- Sample of a Movie Details page. Your page could look similar to the below **Figure 1.2**

**Figure 1.3**



- Ensure this page has the movie title, synopsis, released date and poster.
- You are also required to display a link to the trailer of the movie. To accomplish this, you have to send a request to the “**Movie Videos**” API.
- To obtain the endpoint for the “Movie Videos” API, navigate to the URL below :
- <https://developers.themoviedb.org/3/movies/get-movie-videos>
- See below for example **Figure 1.4**

**Figure 1.4**



The endpoint for the Movie Videos API is :

[https://api.themoviedb.org/3/movie/{movie\\_id}/videos?api\\_key=<<api\\_key>>&language=en-US](https://api.themoviedb.org/3/movie/{movie_id}/videos?api_key=<<api_key>>&language=en-US)

- As shown in the above, the Movie Videos API endpoint requires your API key and a movie id. The movie id passed would be the movie id of the clicked movie on the home page.
- Remember, every movie returned from the Now Playing API has a unique id, see below

**Figure 1.5 (Now Playing API showing the id for each movie)**

```
{
  "results": [
    {
      "popularity": 351.193,
      "vote_count": 148,
      "video": false,
      "poster_path": "/AsdB9A2XGa1CZVjlyG9tRf03IfW.",
      "id": 516486,
      "adult": false,
      "backdrop_path": "/xXBnM6uSTk6qqCf0SRZKXcga9B",
      "original_language": "en",
      "original_title": "Greyhound",
      "genre_ids": [
        28,
        18,
        10752
      ],
      "title": "Greyhound",
      "vote_average": 7,
      "overview": "A first-time captain leads a con",
      "lines of WW2. With no air cover protection fo",
      "chance to win the war.",
      "release_date": "2020-07-10"
    },
    {
      "popularity": 117.321,
      "vote_count": 176
```

- The below endpoint will return all the Youtube trailers for movie **516486**
- [https://api.themoviedb.org/3/movie/516486/videos?api\\_key=yourAPIKey&language=en-US](https://api.themoviedb.org/3/movie/516486/videos?api_key=yourAPIKey&language=en-US)
- See expected results :

**Figure 1.6**

```
{
  "id": 516486,
  "results": [
    {
      "id": "5e71049b4f9a9900135383b4",
      "iso_639_1": "en",
      "iso_3166_1": "US",
      "key": "eyzxu26-Wqk",
      "name": "Official Trailer",
      "site": "YouTube",
      "size": 1080,
      "type": "Trailer"
    },
    {
      "id": "5ee27c7f4c1bb0001ea5d04e",
      "iso_639_1": "en",
      "iso_3166_1": "US",
      "key": "58yeg_hyh2U",
      "name": "Greyhound - Trailer | Apple TV+",
      "site": "YouTube",
      "size": 1080,
      "type": "Trailer"
    },
    {
      "id": "5f09e3e3f48b340036fc4489",
      "iso_639_1": "en",
      "iso_3166_1": "US",
      "key": "5f09e3e3f48b340036fc4489",
      "name": "Greyhound - Trailer | Apple TV+",
      "site": "YouTube",
      "size": 1080,
      "type": "Trailer"
    }
  ]
}
```

- Based on the above results , you would realize that the movie **516486** has multiple trailers. This will not be the case for every movie.
- You will only need to provide a link that will navigate the user to the Youtube trailer of the clicked movie from the Movie Details page.
- Lastly, the **key** property that I highlighted in **Figure 1.6**, is needed to get the Youtube trailer video to be displayed.
- See a valid Youtube Trailer :
- <https://www.youtube.com/watch?v=TcMBFSGVi1c>
- **TcMBFSGVi1c** is an example of a key that would be “spat” out from the API

## THE END