The Movie Place

Group Members:

Sanyam Savla - 1811040

Shreyas More - 1811049

Parshva Shah - 1811046

Topic: The Movie Place (imdb)

Description : One place for all the movies and tv shows, get interesting facts and all the information, trailers etc. The users can create an account to rate and write reviews for a particular movie.

Github Link: **https://github.com/staticshreyas/TheMoviePLace**

Video Link: **https://drive.google.com/drive/folders/1-c3zb5kXKhfj_m_Cjw3MgCSZIpJkyFmU**

Execution Steps:

1. Clone or download zip the repository from above GitHub link.
2. Navigate to the **node-TM** folder and run the command **npm install** to install all the necessary packages for the node.
3. Once installed, run command **node index.js** to start the backend server on **port 3000**.
4. Navigate to the **angular-TM** folder and run the command **npm install** to install all the necessary packages for the angular.
5. Once installed , run the command **ng serve** to start the front end server on **port 4200.**
6. Once both the servers are started, open the browser and type the url **http://localhost:4200.**

Theory:

Axios:

- In a nutshell, Axios is a Javascript library used to make HTTP requests from node.js or XMLHttpRequests from the browser that also supports the ES6 Promise API.
- **What it aims to improve:** .fetch()

  Fetch uses a two-step process when dealing with JSON data; after making an initial request you'll then need to call the .json() method on the response in order to receive the actual data object. If we were to have a fetch request to get back an object of relevant movies for our app, I might have something like the following:

  ```
  const baseUrl = 'http://localhost:3000/api/v1'
  fetch(`${baseUrl}/movies`).then(res => res)
  ```

  This would return a pending promise and response object letting us know our request was successful, but where's our data?! We have to remember to pass our response to the .json() method as well.

  ```
  const baseUrl = 'http://localhost:3000/api/v1'
  fetch(`${baseUrl}/movies`).then(res => res.json())
  ```

  Now that's not a big deal, like many things it really comes down to syntax memorization. But as developers any kind of shorthand or time saver is worth consideration.
- **Better error handling**

  There's some extra work we need to put in on our end for the desired result when it comes to properly logging response errors with .fetch().

  A fetch() promise will be rejected with a TypeError when a network error is encountered or CORS is misconfigured on the server side, although this usually means permission issues or similar — a 404 does not constitute a network error.

  It turns out fetch() only rejects promises in relatively rare networking error situations like a DNS lookup failure. Luckily fetch() provides a simple ok flag that indicates whether or not the status code of a response is in the successful range. Generally what happens is you'll end up writing a one-size-fits-all error handling function to be used on all of your fetch() calls.

CORS:

- Cross-Origin Resource Sharing (CORS) is a protocol that enables scripts running on a browser client to interact with resources from a different origin. This is useful because, thanks to the same-origin policy followed by XMLHttpRequest and fetch, JavaScript can only make calls to URLs that live on the same origin as the location where the script is running. For example, if a JavaScript app wishes to make an AJAX call to an API running on a different domain, it would be blocked from doing so thanks to the same-origin policy.
- **Why Is It Needed? :** Most of the time, a script running in the user's browser would only ever need to access resources on the same origin (think about API calls to the same backend that served the JavaScript code in the first place). So the fact that JavaScript can't normally access resources on other origins is a good thing for security.
  In this context, "other origins" means the URL being accessed differs from the location that the JavaScript is running from, by having:
    - a different scheme (HTTP or HTTPS)
    - a different domain
    - a different port

  However, there are legitimate scenarios where cross-origin access is desirable or even necessary. For example, if you're running a React SPA that makes calls to an API backend running on a different domain. Web fonts also rely on CORS to work.

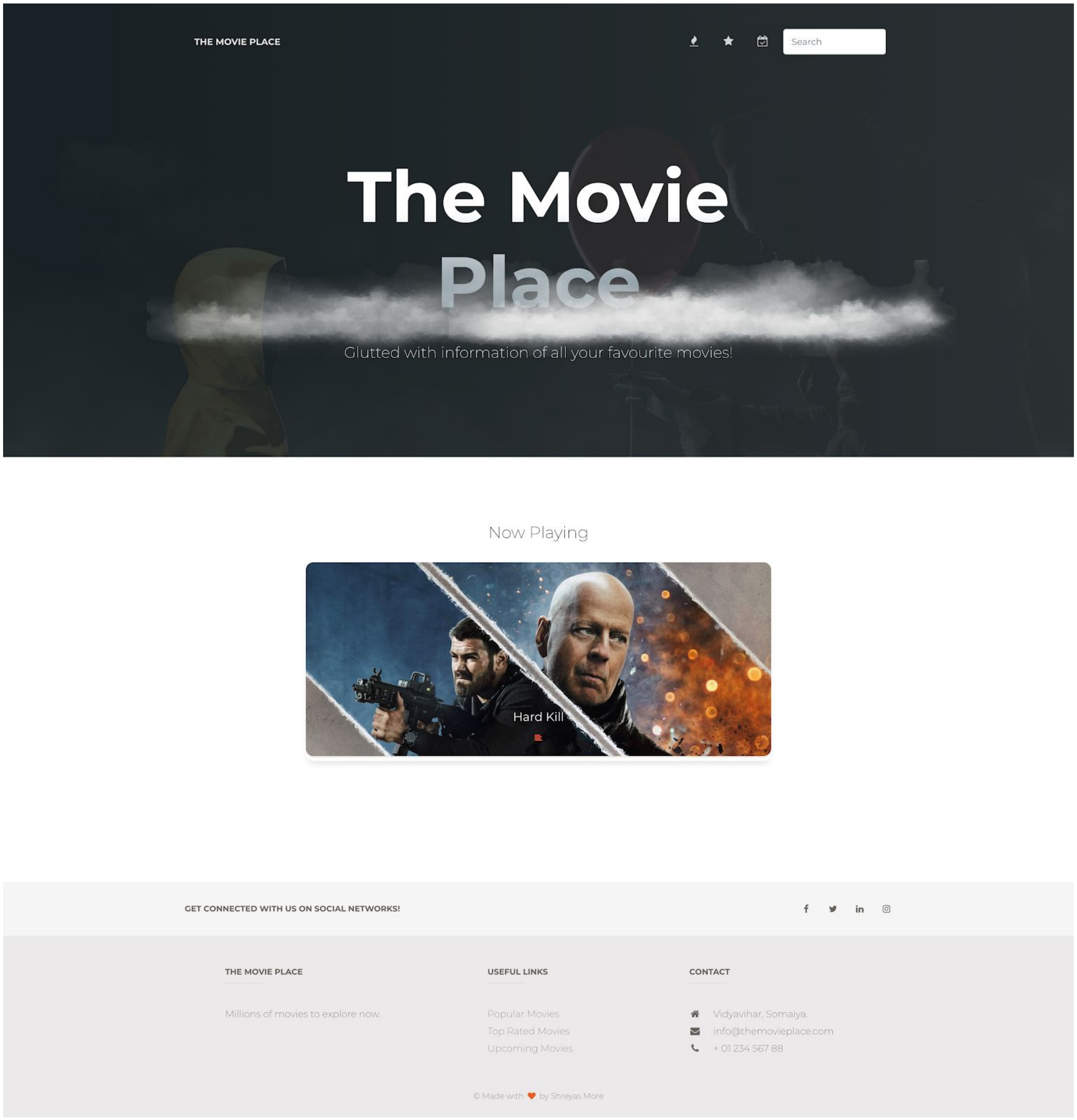
Observables:

- Observables provide support for passing messages between parts of your application. They are used frequently in Angular and are the recommended technique for event handling, asynchronous programming, and handling multiple values.
- An observable can deliver multiple values of any type—literals, messages, or events, depending on the context. The API for receiving values is the same whether the values are delivered synchronously or asynchronously. Because setup and teardown logic are both handled by the observable, your application code only needs to worry about subscribing to consume values, and

when done, unsubscribing. Whether the stream was keystrokes, an HTTP response, or an interval timer, the interface for listening to values and stopping listening is the same.

UI Modules:

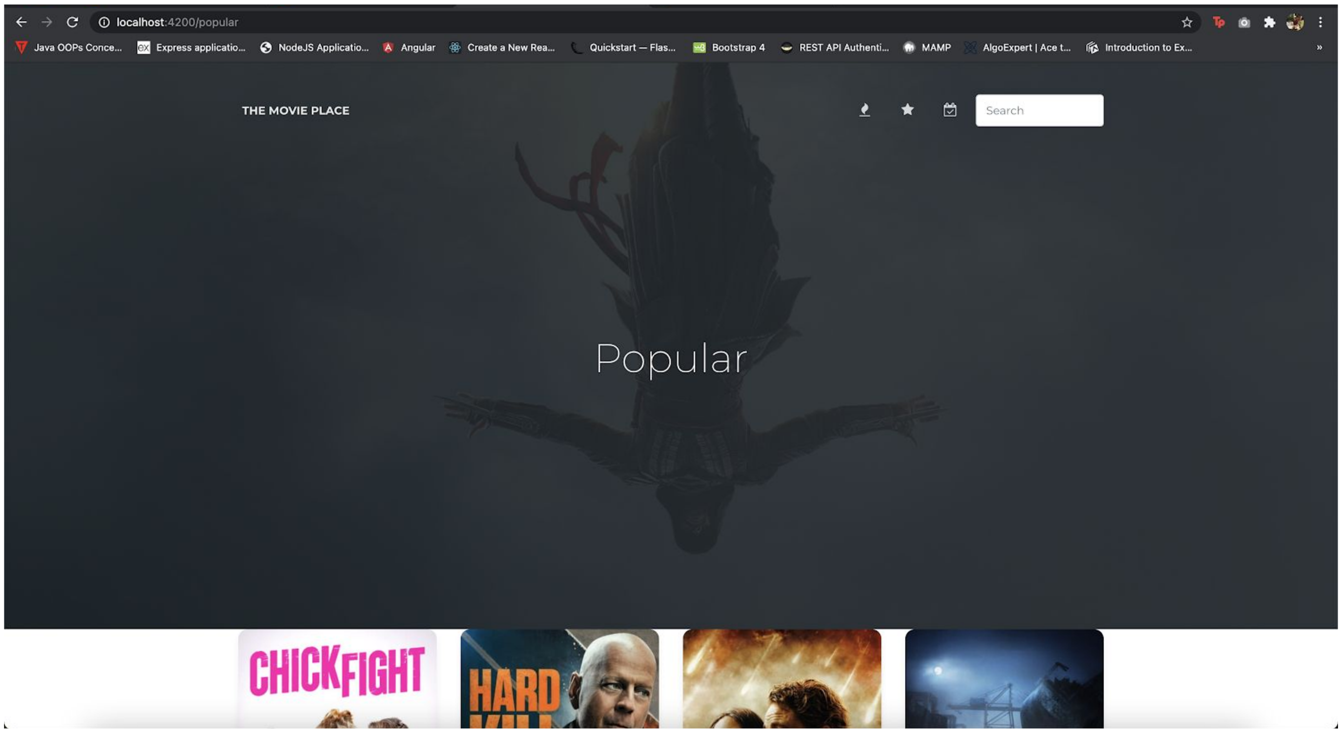
1. **Home Screen:**
   - First when we launch the website, we can see the header, which contains various categories such as **popular, top rated and upcoming.** Also, we can see the search bar where we can enter the movie name and search for movies.
   - The next part contains the **product title** and a beautiful animation.
   - When we scroll down, we can see a **carousel,** of all the movies that are currently playing in theatres. If we **hover** above the carousel we can see the illuminated buttons to scroll through the carousel and also the button at the bottom, which when clicked routes to the details of the movie.
   - Further when we scroll down, we can see an appealing footer as shown below in the image.

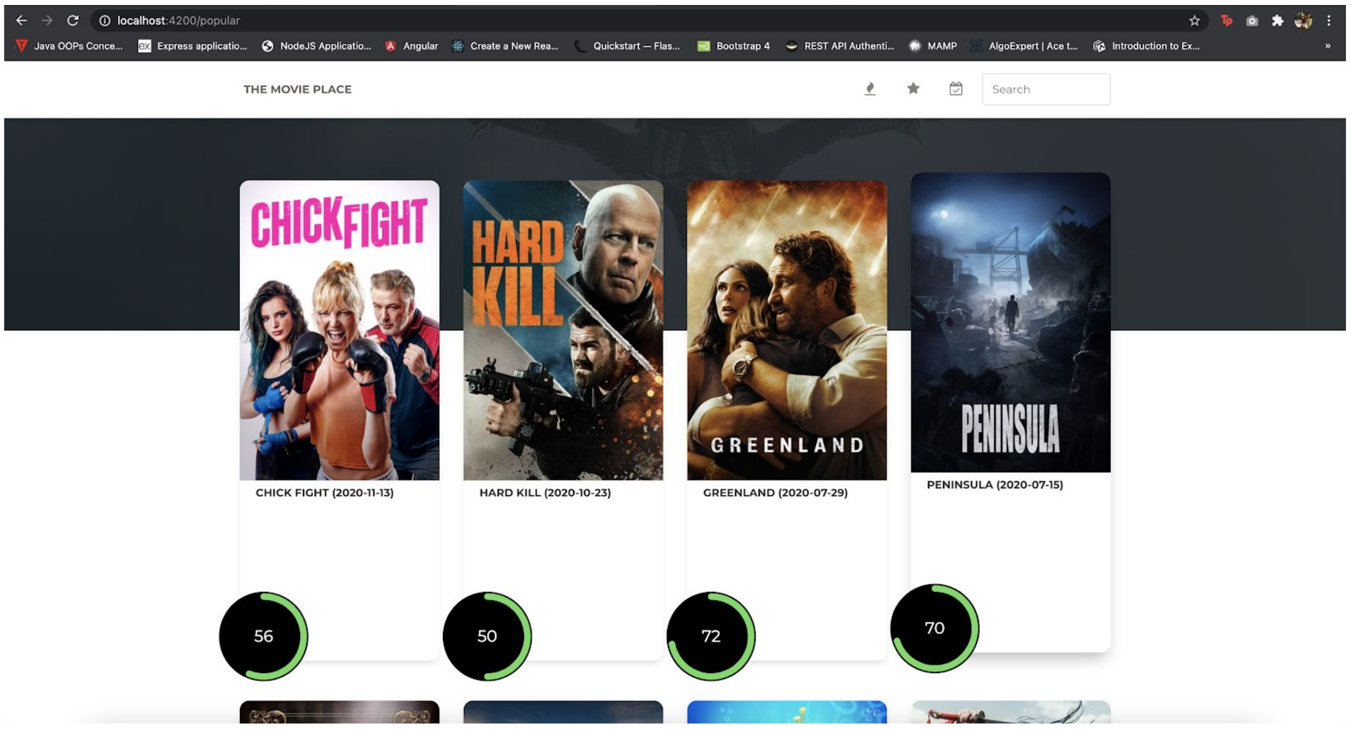

THE MOVIE PLACE
Search
The Movie Place
Glutted with information of all your favourite movies!
Now Playing
Hard Kill
GET CONNECTED WITH US ON SOCIAL NETWORKS!
THE MOVIE PLACE
Millions of movies to explore now.
USEFUL LINKS
Popular Movies
Top Rated Movies
Upcoming Movies
CONTACT
Vidyavihar, Somaiya.
info@themovieplace.com
+ 01 234 567 88
© Made with ♥ by Shreyas More

2. Categories:

- As mentioned above, there are various categories of movies to choose from.
- Each category page has **20 movie cards,** which contains the **critic rating, movie poster, movie name and the release date** of the movie.
- If we **hover** over the card, there is a beautiful animation which helps the user to understand which card he/she is one.
- If we **click** the card, we are routed to a page which shows the details of the movie.
- Below are the category pages displayed.

Popular:



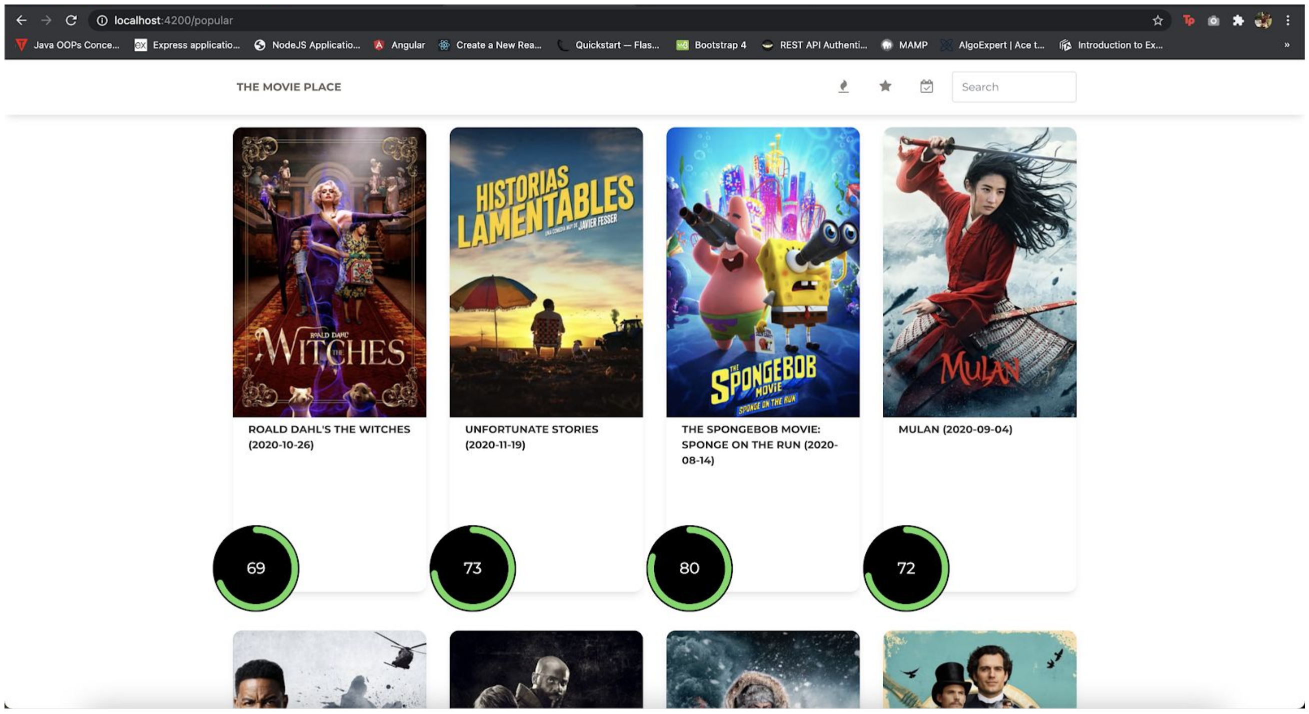


CHICK FIGHT (2020-11-13) 56

HARD KILL (2020-10-23) 50

GREENLAND (2020-07-29) 72

PENINSULA (2020-07-15) 70



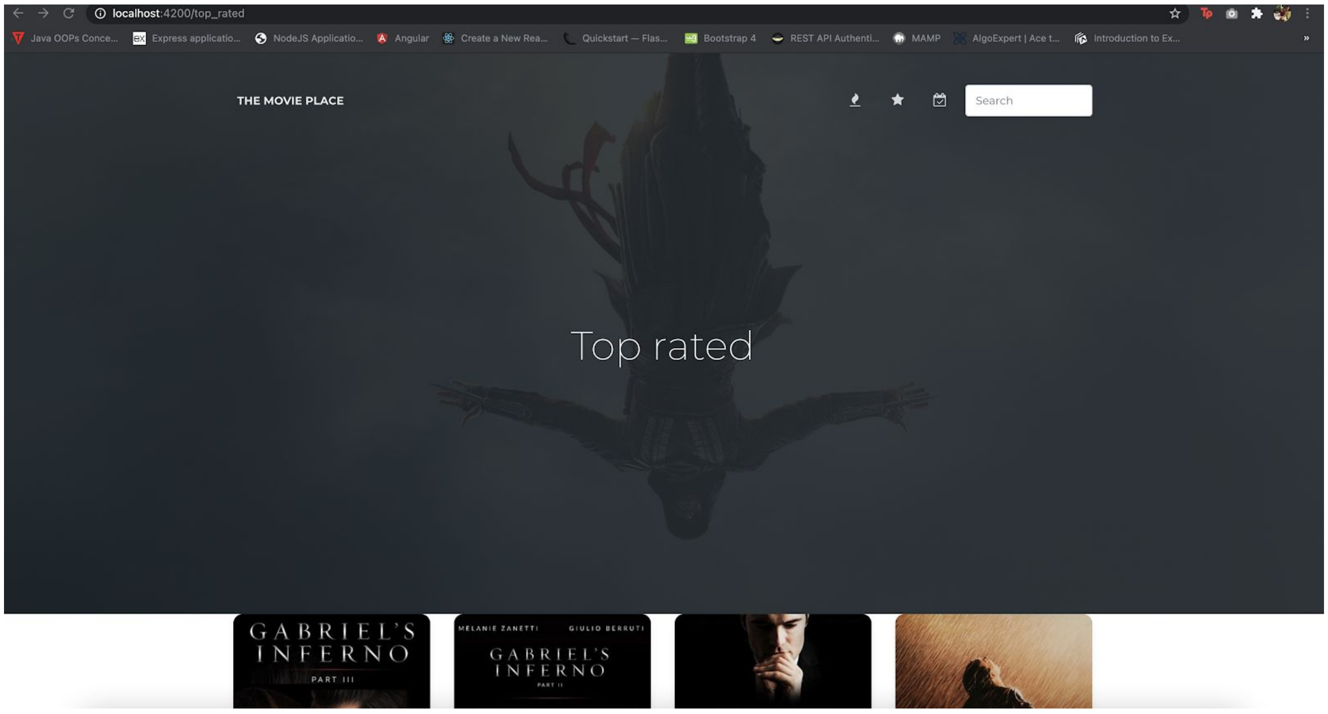
ROALD DAHL'S THE WITCHES (2020-10-26) 69

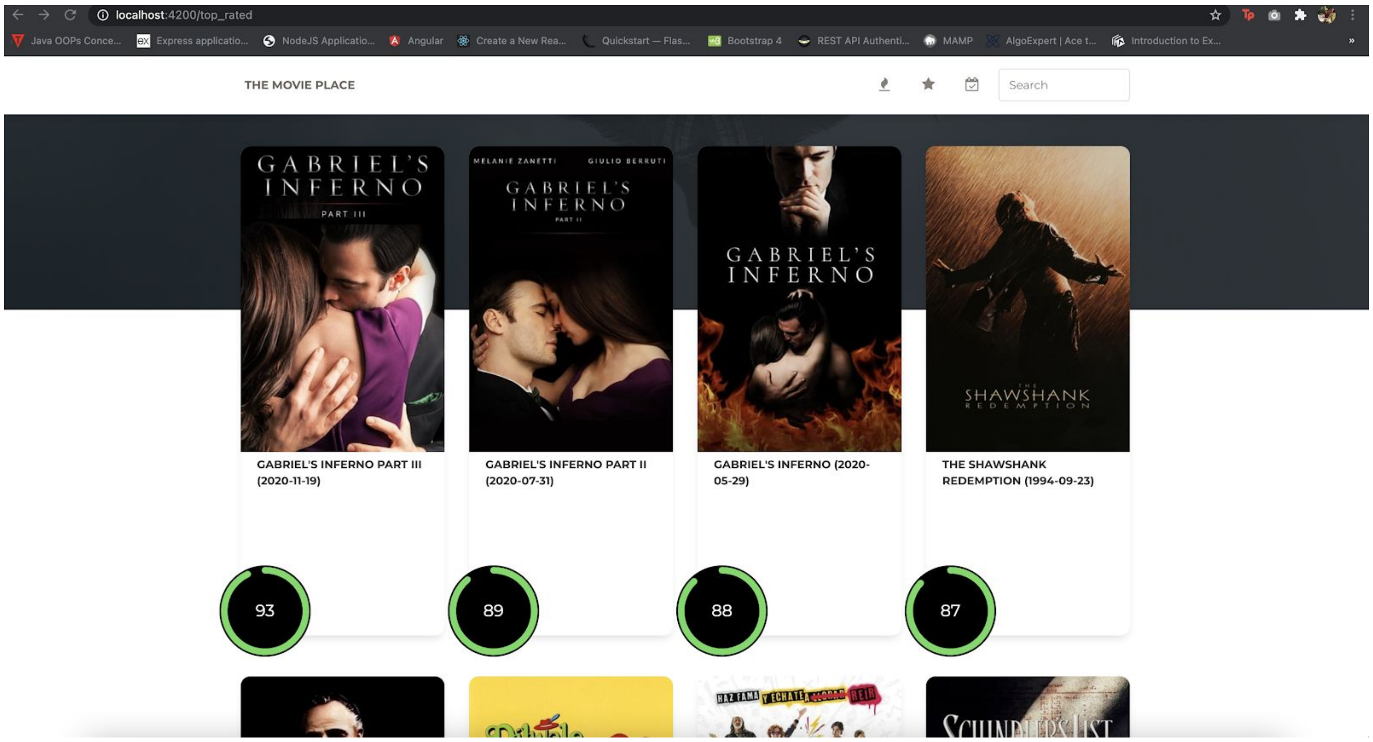
UNFORTUNATE STORIES (2020-11-19) 73

THE SPONGEBOB MOVIE: SPONGE ON THE RUN (2020-08-14) 80

MULAN (2020-09-04) 72

Top rated:


localhost:4200/top_rated
THE MOVIE PLACE
Search
Top rated
GABRIEL'S INFERNO PART III
MELANIE ZANETTI GIULIO BERRUTI GABRIEL'S INFERNO PART II


localhost:4200/top_rated
Java OOPs Conce... Express applicatio... NodeJS Applicatio... Angular Create a New Rea... Quickstart — Flas... Bootstrap 4 REST API Authenti... MAMP AlgoExpert | Ace t... Introduction to Ex...
Search

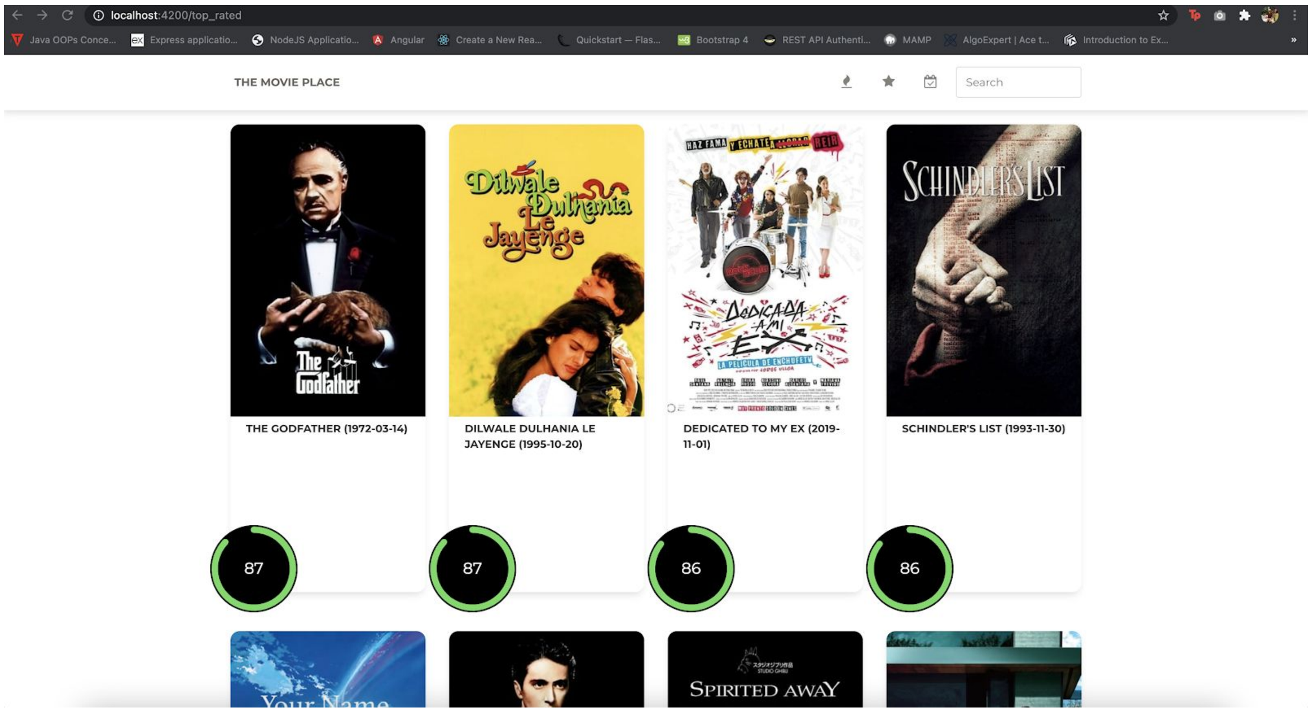
THE MOVIE PLACE

GABRIEL'S INFERNO PART III (2020-11-19) 93

GABRIEL'S INFERNO PART II (2020-07-31) 89

GABRIEL'S INFERNO (2020-05-29) 88

THE SHAWSHANK REDEMPTION (1994-09-23) 87


localhost:4200/top_rated
Java OOPs Conce... Express applicatio... NodeJS Applicatio... Angular Create a New Rea... Quickstart — Flas... Bootstrap 4 REST API Authenti... MAMP AlgoExpert | Ace t... Introduction to Ex...
Search

THE MOVIE PLACE

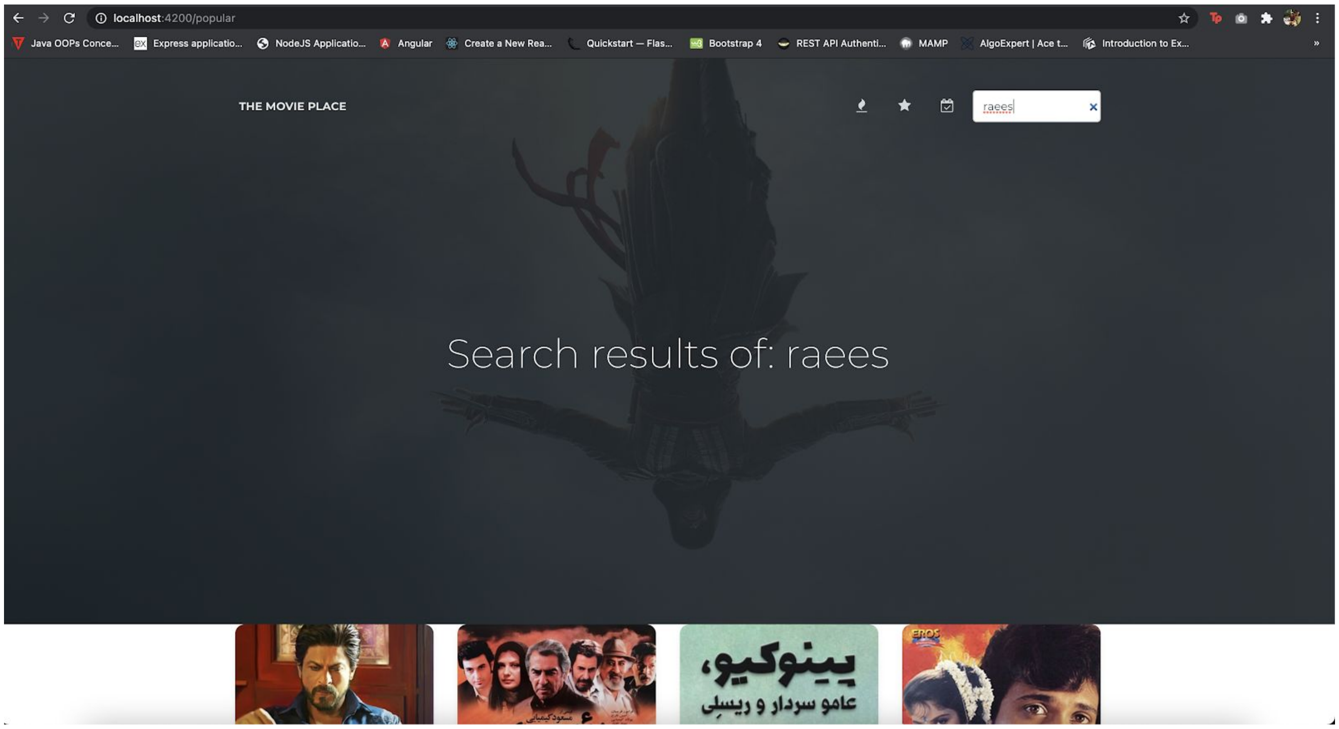
THE GODFATHER (1972-03-14) 87

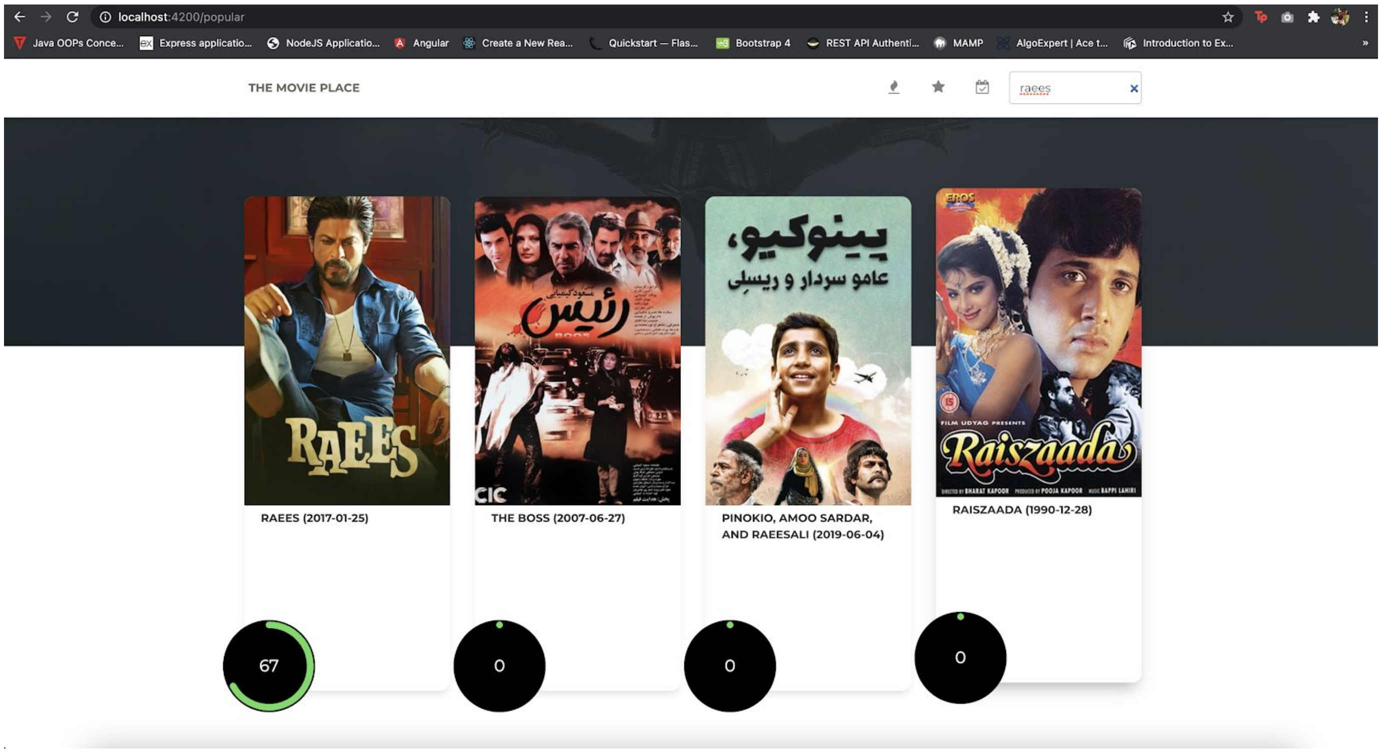
DILWALE DULHANIA LE JAYENGE (1995-10-20) 87

DEDICATED TO MY EX (2019-11-01) 86

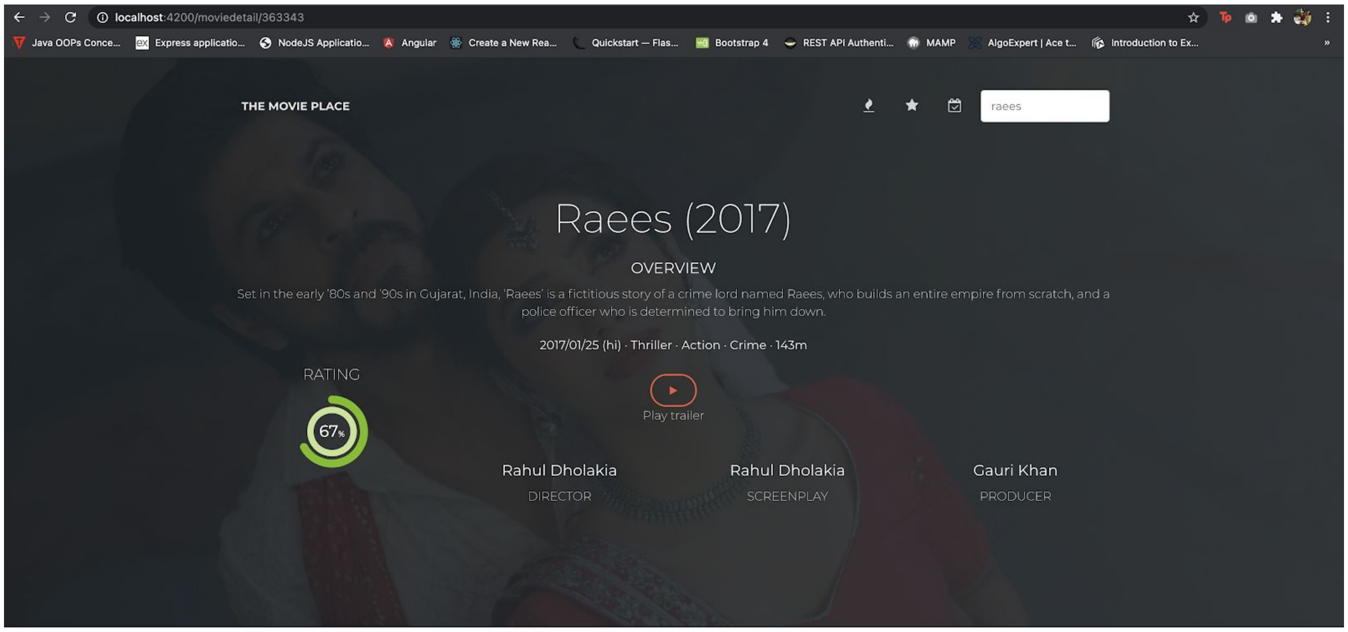
SCHINDLER'S LIST (1993-11-30) 86

3. **Search:**
   - The search bar is in the header as mentioned above.
   - If we want to search for a specific movie, we just type the movie name in the given search bar and then hit enter.
   - As shown above, the cards of the movies, related to the search query are displayed.





4. **Movie Details:**
    - As mentioned above, clicking on a card or clicking on the button on the carousel which was on the home page, we are redirected to the page which contains all the trivia of the movie.
    - This page contains a cool backdrop image of the movie, on which the movie title, the rating of the movie, the release date, the certification, genre, runtime and the technical team is displayed.
    - There is also a play trailer button, which redirects us to the trailer of the movie.
    - On scrolling below, we can see the top cast of the movie displayed stunningly.



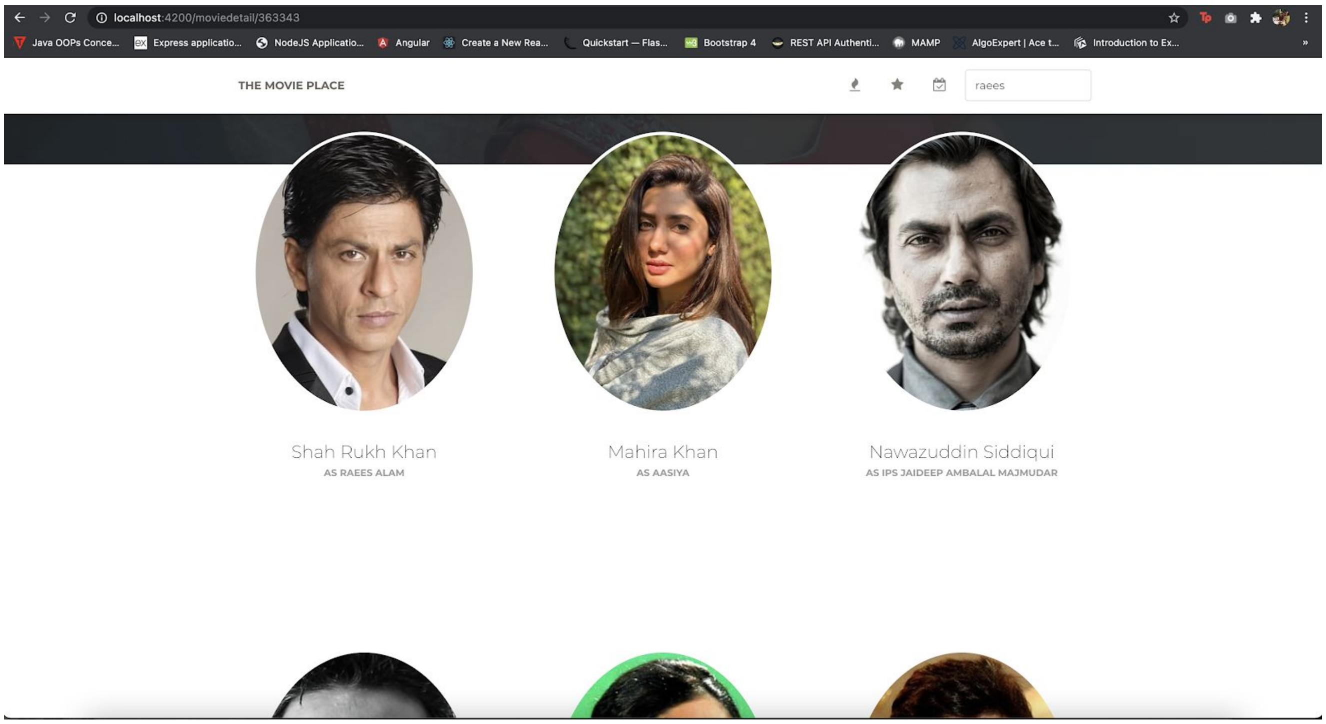
Raees (2017)

OVERVIEW

Set in the early '80s and '90s in Gujarat, India, 'Raees' is a fictitious story of a crime lord named Raees, who builds an entire empire from scratch, and a police officer who is determined to bring him down.

2017/01/25 (hi) · Thriller · Action · Crime · 143m

Rahul Dholakia DIRECTOR · Rahul Dholakia SCREENPLAY · Gauri Khan PRODUCER


localhost:4200/moviedetail/363343
Java OOPs Conce...
Express applicatio...
NodeJS Applicatio...
Angular
Create a New Rea...
Quickstart — Flas...
Bootstrap 4
REST API Authenti...
MAMP
AlgoExpert | Ace t...
Introduction to Ex...
THE MOVIE PLACE
raees
Shah Rukh Khan
AS RAEES ALAM
Mahira Khan
AS AASIYA
Nawazuddin Siddiqui
AS IPS JAIDEEP AMBALAL MAJMUDAR