Musicological Conjuror Team

For the User

We are team Musicological. This is the documentation of the project we have done. The main purpose of this project is to give idea for the viewer about the top ratings of the movies we have in our database.

Installation

Set up

1. Download the Movie-app source code and it extract to a directory.
    a. https://github.com/RodrigoFigueroaM/473-movie-app.git
2. Install MongoDB.
3. Start the MongoDB server by typing and entering into a terminal:

```
$ mkdir --p ./mongodb/data
```

```
$ ./mongodb/bin/mongod --dbpath=$HOME/mongodb/data
```

Running the app

1. Type in Terminal: npm install to install all of the dependencies

    ```
    $ npm install
    ```

2. Store movies to database

    ```
    $ node store_movies.js
    ```

3. Run the server

    ```
    $ node server.js
    ```

4. Access the website

Go to browser and type localhost:3000

Functionalities

Voting

Since this is kind of personal favorite movie blog, we don't have the option in which the viewer must have log-in account in order to vote for a movie. Hence, the viewer can up vote for the movie that he likes as much as he wants and down vote for others. However, the vote will be saved in our database and won't disappear even refreshing the page. So any guest who passes by our movie blog always knows top three trending movies by the total number of votes for top three movies and the chart. Please note this chart is rating bar that displays the average of positive votes (up votes/ total votes) because we just want to see how much people like that particular movie.

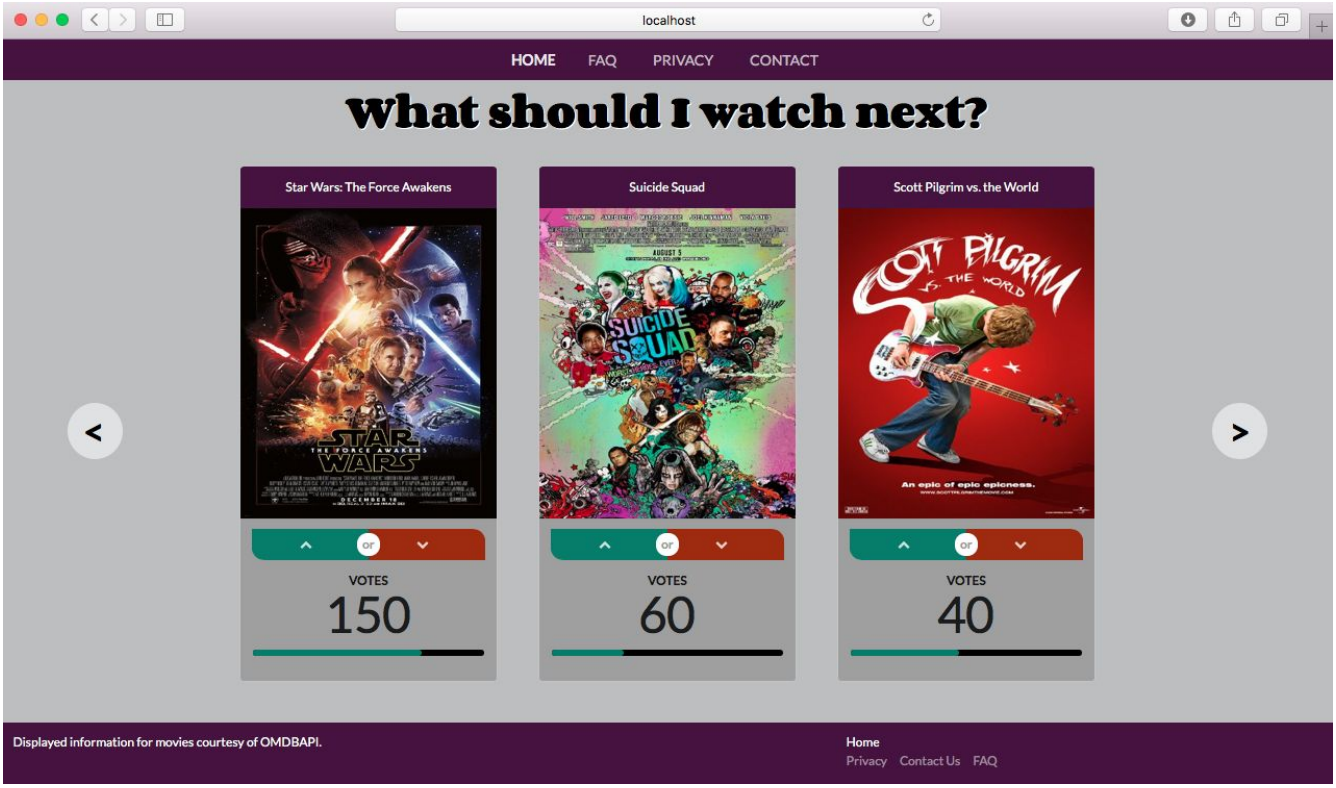

Figure 1 Top three movies. Bar below votes shows average of votes. Decreasing order average rating. Star Wars: the Force Awakens (75%), Suicide Squad(24%), Scott Pilgrim Vs. the World(50%).

Navigation

Users can navigate through the list of movies by clicking to left or right buttons arrow showed on the sides. Upon click, three new movies from the list will fade in within .5 seconds.

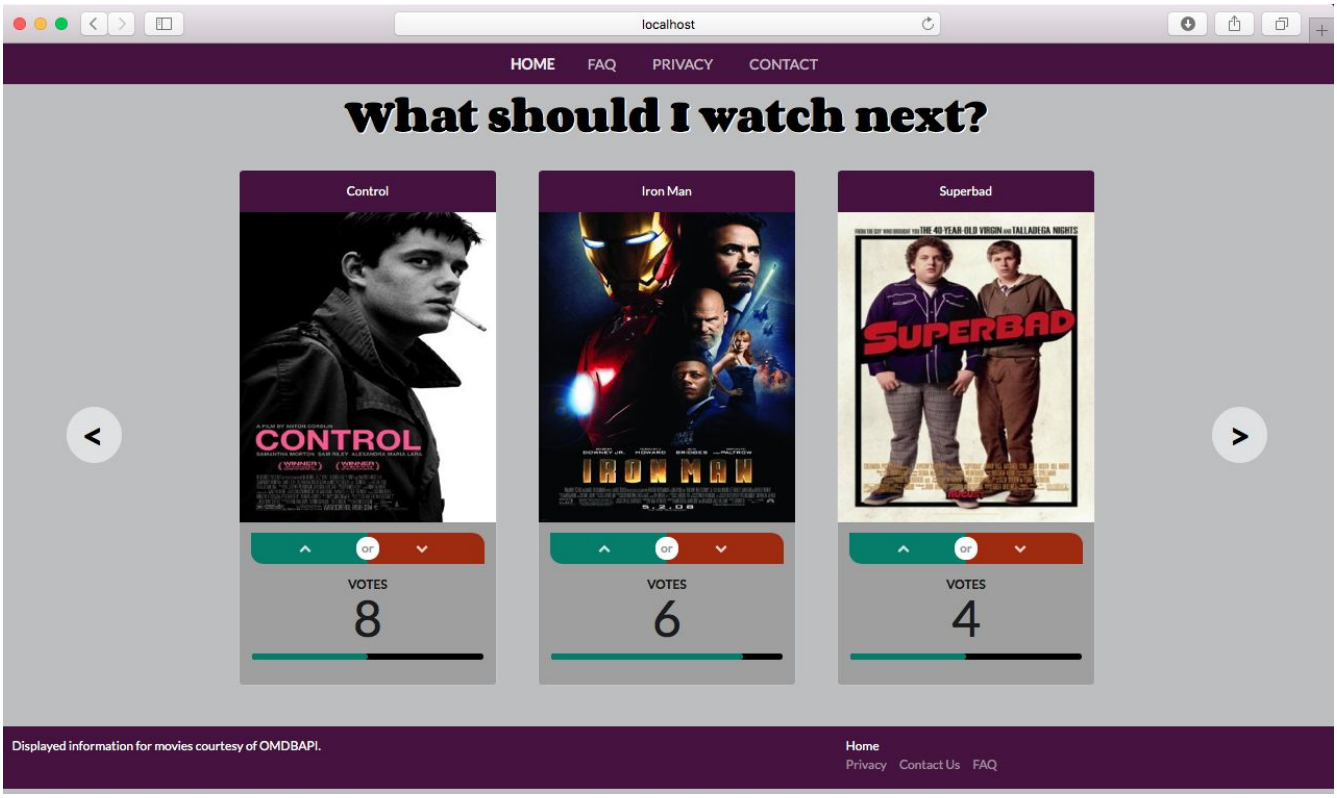

Figure 2. After clicking on right navigational button, a new set of three movies is presented to the user

Additional Details

In addition, if the viewer would like to know more information of the movie, he or she can click on the image of the movie, and there is a message info pop-up. This message includes the rating film system(PG-13, R…)duration, cast of the movie, any nomination achieved,etc...

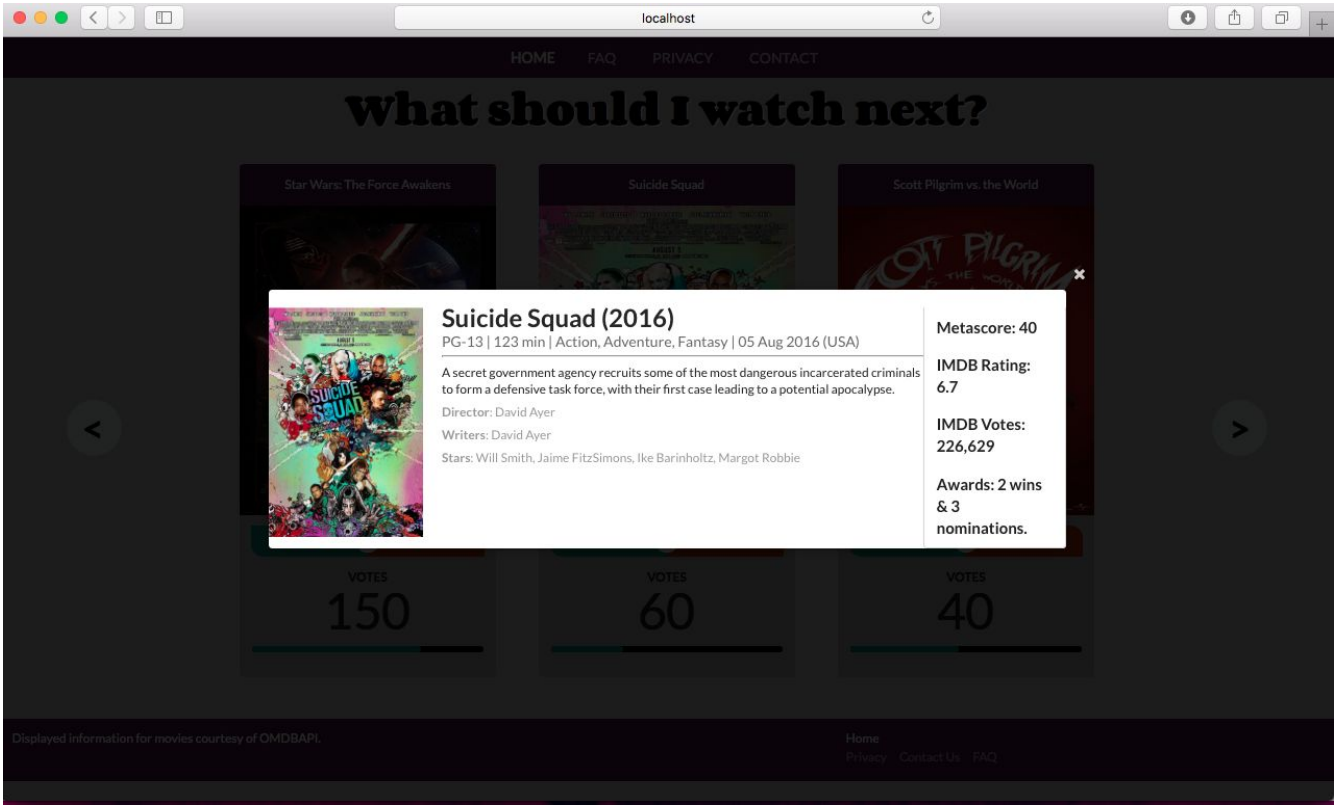

Figure 3. After clicking on movie image, more details about that specific movie are displayed.

Disclosure

All data presented in this web app, including movie posters, is provided by OMDBAPI. We only keep track of the votes on this movies to have an opinion on how good people thing a movie is.

Musicological Conjuror Team

For Developers

Frameworks, Modules, Libraries, etc:

- Semantic UI
- Node.js
- Express
- Bluebird
- Mongoose
- MongoDB
- Request

Installation

Set up

1. Download the Movie-app source code and it extract to a directory.
    a. https://github.com/RodrigoFigueroaM/473-movie-app.git
2. Install MongoDB.
3. Start the MongoDB server by typing and entering into a terminal:

```
$ mkdir --p ./mongodb/data

$ ./mongodb/bin/mongod --dbpath=$HOME/mongodb/data
```

Running the app

1. Type in Terminal: npm install to install all of the dependencies

```
$ npm install
```

2. Store movies to database

```
$ node store_movies.js
```

3. Run the server

   ```
   $ node server.js
   ```

4. Access the website

   Go to browser and type localhost:3000

File Breakdown

Server Side

Folder: ./

- movies.txt
  - Contains a list of movies to be displayed. List is in CVS style.
- store_movies.js
  - Open movies.txt file. Stores all the movies into an array.
  - Connects and retrieves specific information for a movie from OMDBAPI
- server.js
  - Creates routes to server.

Folder: modules

- movieDB.js
  - Creates a database schema using MongoDb and Mongoose

Client Side

Folder: client

- index.html
  - Main HTML page for application. Displays movie cards
- projCon.html
  - Contains contact information
- projPriv.html
  - Contains Privacy statements
- projFaq.html
  - Contains Frequently Asked Questions

Folder: client/css

- style.css
  - Modifies style attributes to give taylor application

Folder: client/js

- main.js
    - Controls the app by calling different functions.
    - In charge of:
        - Connect to server and retrieve data in JSON format
        - Initialize and control slider
        - Control elements on slider
        - Parse data
- initializeTiles.js
    - Creates an array that holds contents of tiles in this case (movie cards)
    - Formats HTML element to be appended for display
- modal.js
    - Formats an HTML modal
    - Displays and removes modal when clicked
- voteHandler.js
    - Controls post requests to server to update votes on a movie
    - Controls animation on rating bar and total number of votes

Development Process

Initial Idea

The initial idea of this app was to host a website where people could vote on a movie in to give it a rating.

Below is the the first stage of our development process.

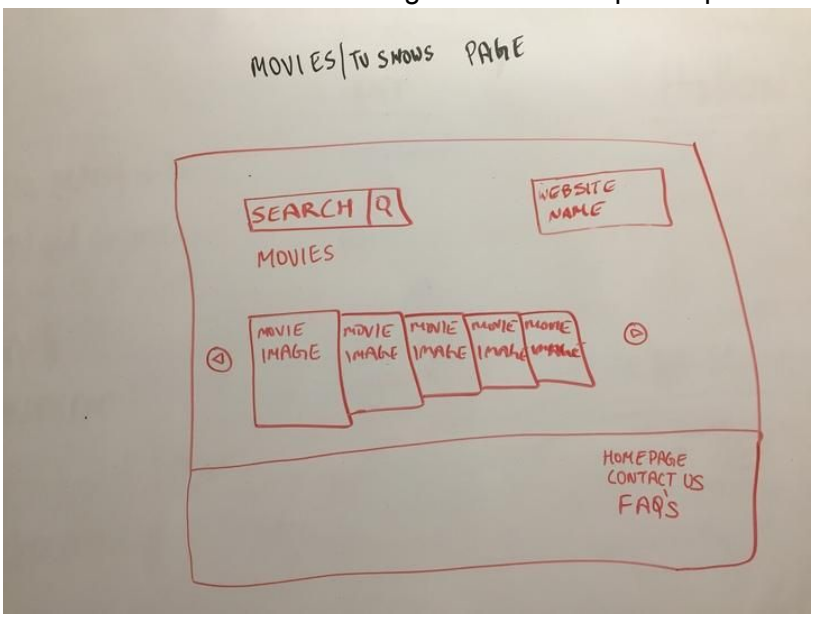

Figure 1: This is the homepage of the website. We dropped idea of searching movies. We have 12 movies instead.

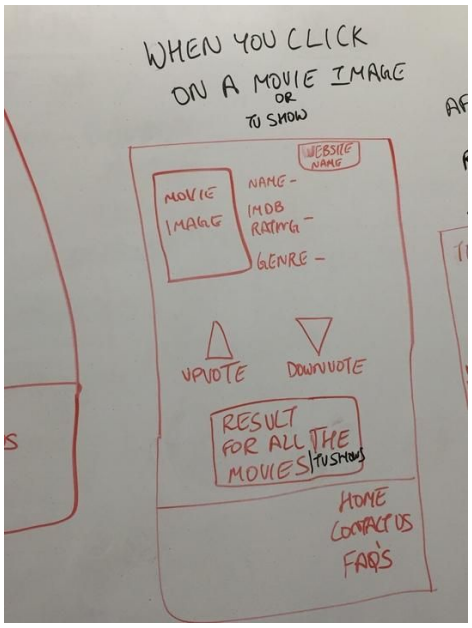

Figure 2 : Each movie image should have detail information of the movie. 2 vote buttons and show result.

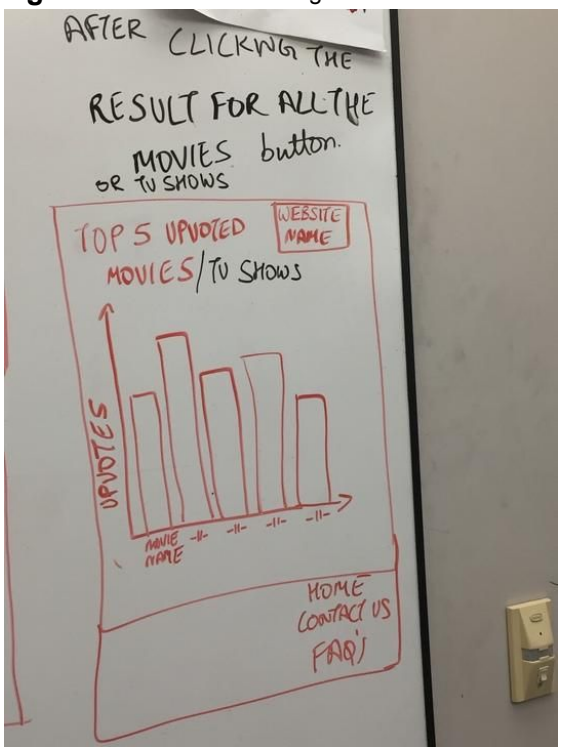

Fig 3: Actual showing the top 3 movies (instead of 5) with the up-votes only