Project 1 – Movie Search System

This project is designed to test your HTML, CSS, and JavaScript skills. You are to design a website that allows users to search and browse movies.

Here is how the system functions:

As the user types in something in the search box to search for movies, the system brings up a suggestion box that shows all the matches (Figure 1). Each suggestion item is the movie title, year, and the main actors and actresses.

When the user clicks "Search", the system shows all movies that match what's the user typed in. The default view is the grid view (Figure 2). Each item in the grid shows a picture of the movie, the movie title, year, and an icon telling whether it is an HD movie. When the user hovers a movie, the title and year go a little up, and a box emerges from the bottom showing the main actors and actresses (Figure 3).

The user can sort the results by year or rating.

When the user clicks the list view icon, the system shows up the movies in the list view (Figure 4).

Data:

The data behind the system is stored in folder "data". The main data is in "movies.js". This is a JSON table that has these fields: title, year, starring, HD (is the movie in HD?), photo (the name of the movie picture file), rating (a number out of 5). All the movie pictures are also in the data folder. You can find the data at this link:

https://www.dropbox.com/sh/97r6rxg44kirg57/AACElQPgu3DGcekGhJBumCMEa?dl=0

You can also find the images (e.g. for icons) here:

https://www.dropbox.com/sh/gqtjgpu4qn3prkm/AACYtoNif0ZRYdwJBtN_uNd0a?dl=0

You can find the Oleo font here:

https://www.dropbox.com/sh/k90br5z5w81vqbl/AACQeqEGC6SAm1z7HZ1L8vr-a?dl=0

Requirements:

1. **HTML** (Weight: 15%)**:** Design one HTML file for the project. Note that all of the HTML code will not be static in the code. For instance, the HTML code for the movies will be added dynamically by JavaScript.

2. **CSS:** Write external CSS that styles your HTML so that the appearance of your website looks like Figures 1, 2, and 4.
   Header (Weight: 10%): The header consists of the website logo, the title, and a search box. The title is defined with the Oleo font. It is positioned with fixed positioning. Figure 5 shows the dimensions that should be used for styling the header.
   Icons bar (Weight: 5%): The icons bar is located underneath the header. It consists of the list and grid icons as well as the sorting combo box.
   Suggestions box (Weight: 10%): The suggestions box shows up when the user starts typing in something in the search box. The maximum number of search results is 5. Further, if a suggestion width goes beyond the boundaries of the box, a scrollbar should show up. Figure 5 shows the styles of the suggestions box.
   Grid View (Weight: 10%): The default view of movies is the grid view (Figure 2). The movies stack next to each other. On top of each movie is the title (in uppercase letters) and the year. When the user hovers the movie, the main stars of the movie show up. Follow the styles shown in Figure 6.
   List View (Weight: 10%): The list view shows the information of movies as a list (Figure 4). The list view is activated when the user presses the list icon on top. Follow the styles shown in Figure 7.
   Responsive Design (Extra Credit: within the scope of the project +10%) Make sure your page respond to different resolutions and devices (See Figure 8).

3. **JavaScript:** Write JavaScript code that does the following:
   Live Search (Weight: 15%)**:** When the user types in something about the movie (e.g. year, title, etc.), the system will bring up the suggestions (Figure 1). When the user clicks a suggestion, the name of the movie is displayed in the search box. Write JS code that does that.
   The data the JavaScript code searches is stored in data/movie.js.
   Search (Weight: 10%)**:** Upon clicking "Search", the page displays the movies that match what the user typed in. Write JS code that does that.
   Switch Views (Weight: 5%)**:** When the user clicks the "list" icon, the system switches the view to the list view. When the user clicks the "grid" icon, the system switches to the grid view. Write JS code that does that.
   Sort Movies (Weight: 10%)**:** The user can sort the movies by rating or year. The options are chosen via the combo box on top. Write JS code that does that.

Submissions:

Zip your entire Netbeans project and submit it on Blackboard.

Notes:

Use your judgment for anything that's not explicitly mentioned in the project description. If the specified dimensions don't work for you, feel free to use your own as long as the appearance relatively matches the figures.
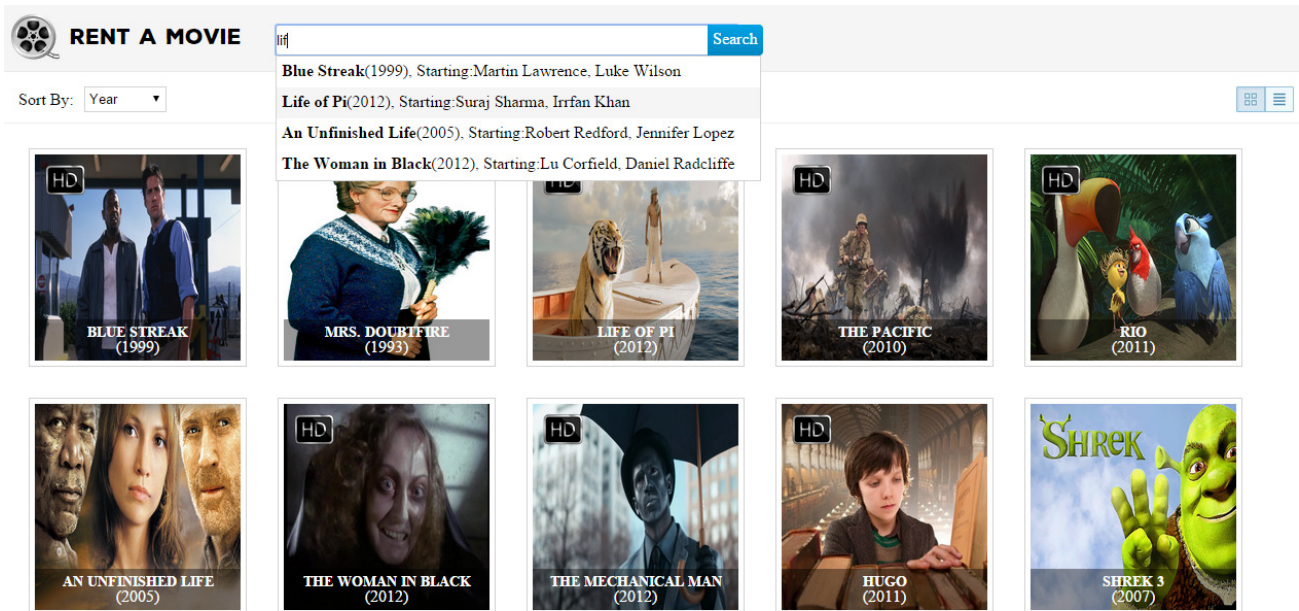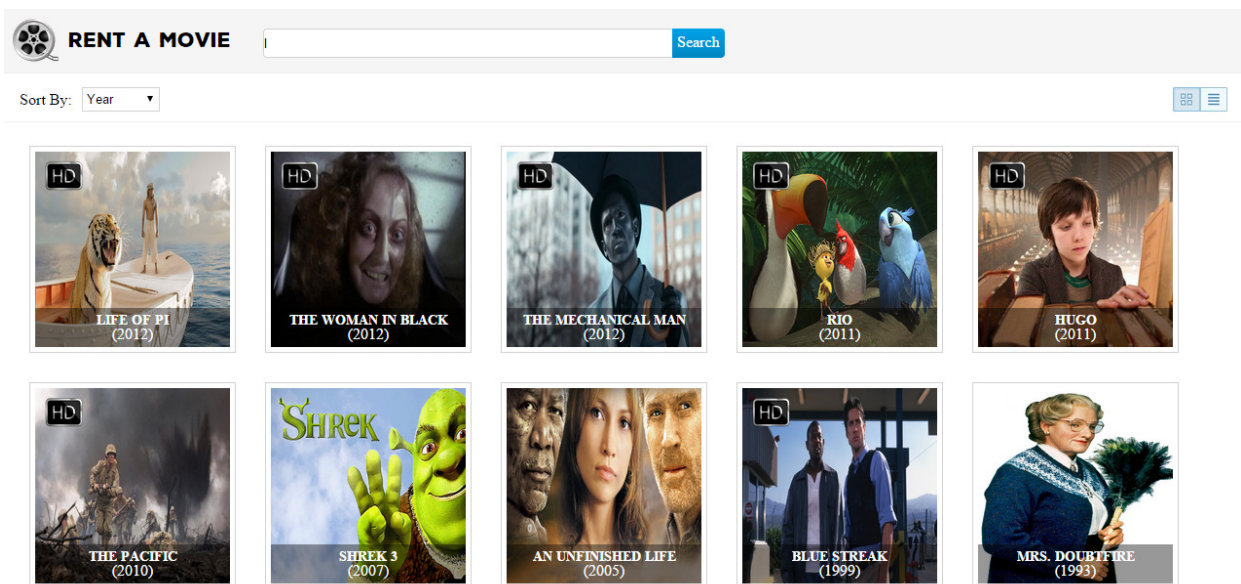
Figure 1: live search
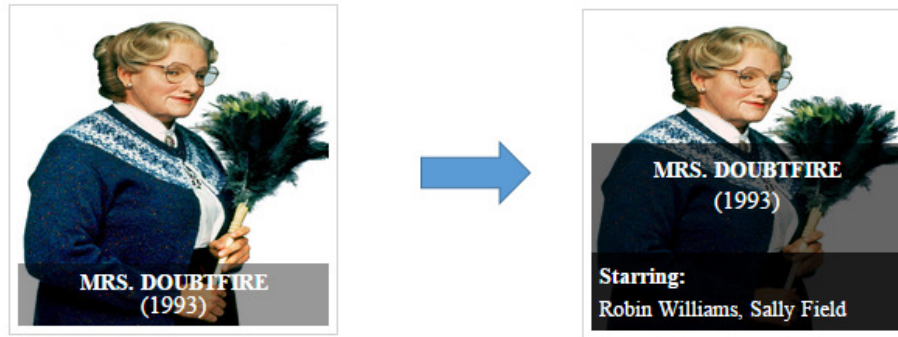

Figure 2: The grid view

Figure 3: hovering the movie brings up the starring information

Figure 4: The list view

A MOVIE

Search

Life of Pi(2012), Starting:Suraj Sharma, Irrfan Khan

The Woman in Black(2012), Starting:Lu Corfield, Daniel Radcliffe

The Mechanical Man(2012), Starting:Jenna Fischer, Chris Messina

Rio(2011), Starting:Jesse Eisenberg, Leslie Mann

Hugo(2011), Starting:Asa Butterfield, Ben Kingsley

Hovered suggestion:
Background color: rgb(245, 245, 245)

Padding: 5px

Width: 470px
Border color: light gray

Figure 5: The styles of the suggestions box

Top, Left Margins: 30px
Border color: light gray

Height: 30px
Width: 40px
Top: 15px
Left: 15px

Padding: 5px

Height: 40px
Color: Black
40% visible

HD

LIFE OF PI
(2012)

200px

200px

HD

LIFE OF PI
(2012)

Starring:
Suraj Sharma, Irrfan Khan

Height: 120px
Color: black
60% visible

Height: 40px
Color: black
70% visible
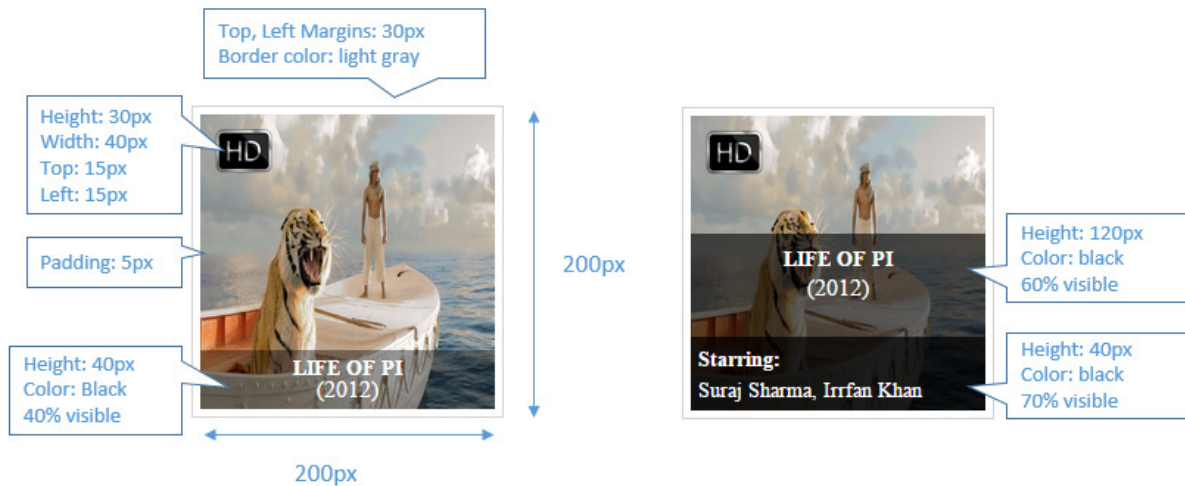
Figure 6: The styles of movie images in the grid view

Top: 15px
Left: 10px

10px

Font-size: 18px

Font-size: 15px
Color: gray

AN UNFINISHED LIFE (2005)

Starring: Robert Redford, Jennifer Lopez

Rating: ★★★☆☆

20px X 20px

An Unfinished Life is a 2005 drama film directed by Swedish director
Lasse Hallström, and based on the Mark Spragg novel of the same
name. The film stars Robert Redford, Jennifer Lopez, and Morgan
Freeman.

260px

Height: 220px
Width: 220px
Padding: 5px

10px

Line-height: 1.5
Width: 450px
Background-color: rgb(236,255,255)

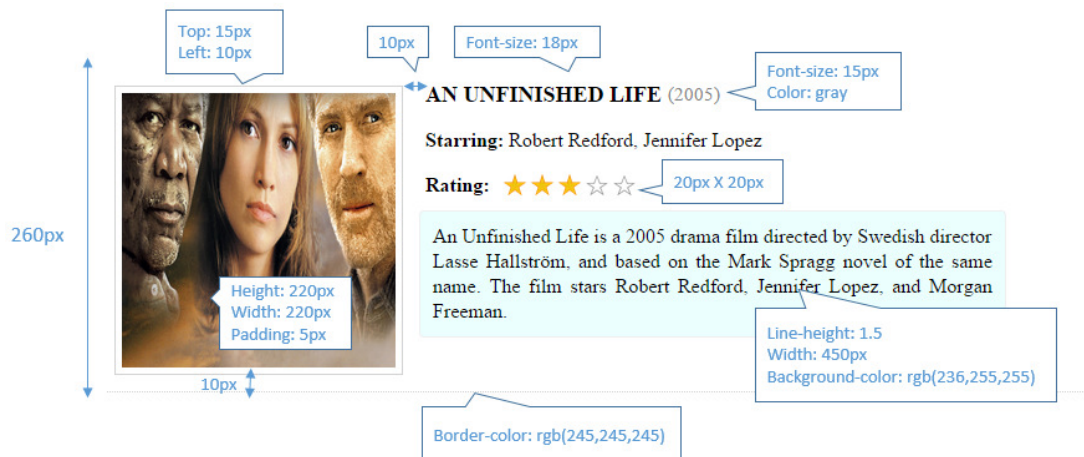Border-color: rgb(245,245,245)

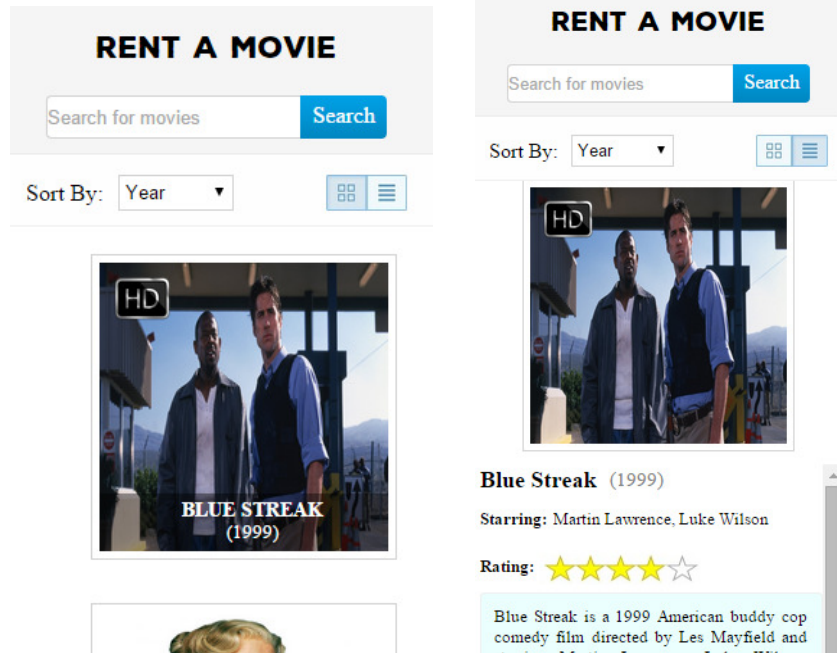Figure 7: The styles of movie images in the list view

Figure 8: The appearance of the page on a mobile screen