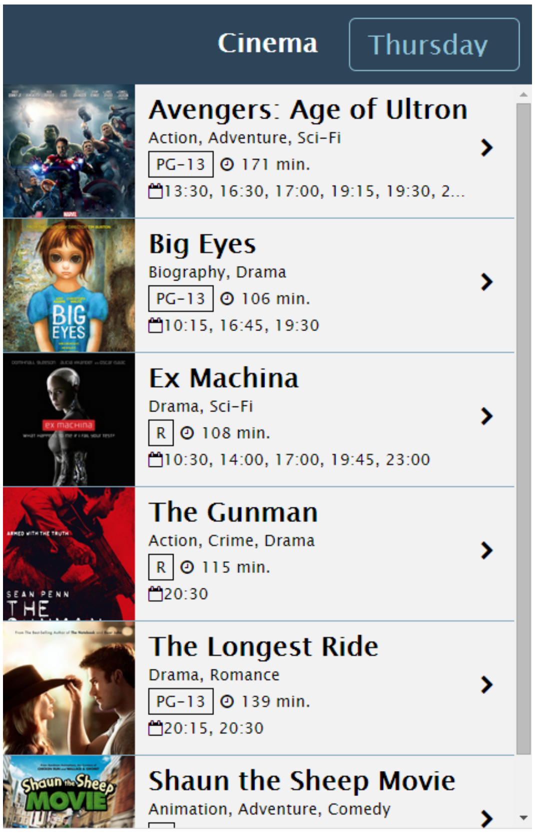
Practical Exam

During this exam, you'll create a cinema app prototype, which will allow users to reserve seats.

> **Begin**: 14:15h. Download the file "exam.zip" from Moodle, which contains a document structure, images and JS frameworks: https://www.moodle.tum.de/mod/resource/view.php?id=308591
>
> **End**: 16:15h. Upload your solution to Moodle at the end of the session: https://www.moodle.tum.de/mod/assign/view.php?id=308597
>
> **Total Points**: 107 P.

Please, make sure that the layout of your prototype corresponds to this sample below:

View 1: Cinema overview



Avengers: Age of Ultron
Action, Adventure, Sci-Fi
PG-13 ⊙ 171 min.
13:30, 16:30, 17:00, 19:15, 19:30, 2...

Big Eyes
Biography, Drama
PG-13 ⊙ 106 min.
10:15, 16:45, 19:30

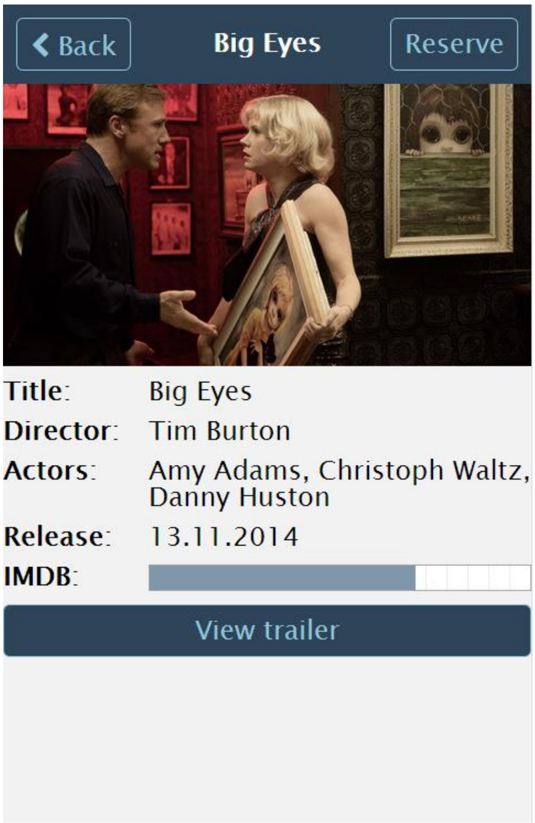
Ex Machina
Drama, Sci-Fi
R ⊙ 108 min.
10:30, 14:00, 17:00, 19:45, 23:00

The Gunman
Action, Crime, Drama
R ⊙ 115 min.
20:30

The Longest Ride
Drama, Romance
PG-13 ⊙ 139 min.
20:15, 20:30

Shaun the Sheep Movie
Animation, Adventure, Comedy

View 2: Movie detail view

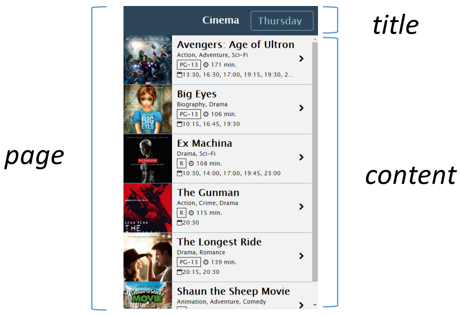


Title:	Big Eyes
Director:	Tim Burton
Actors:	Amy Adams, Christoph Waltz, Danny Huston
Release:	13.11.2014
IMDB:	

Please fulfil the following requirements.

HTML Structure (20 Points)

Requirement 1: Start with the provided “index.html” template. Create the three structural block level elements *page*, *title*, and *content* with their corresponding classes. A page can later be both the *overview* and the *details* view, start with the overview (assign an id). Use semantic elements where possible. (2 P)



Cinema Thursday

Avengers: Age of Ultron
Action, Adventure, Sci-Fi
PG-13 171 min.
13:30, 16:30, 17:00, 19:15, 19:30, 2...

Big Eyes
Biography, Drama
PG-13 106 min.
10:15, 16:45, 19:30

Ex Machina
Drama, Sci-Fi
R 108 min.
10:30, 14:00, 17:00, 19:45, 23:00

The Gunman
Action, Crime, Drama
R 115 min.
20:30

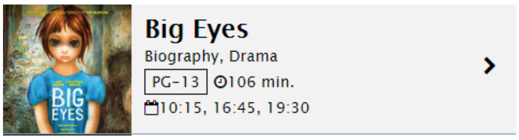
The Longest Ride
Drama, Romance
PG-13 139 min.
20:15, 20:30

Shaun the Sheep Movie
Animation, Adventure, Comedy

Requirement 2: Please, insert the “Cinema” headline into the *title*. (1 P)

Requirement 3: Insert a select dropdown into the *title*. Assign the two classes *control* and *right* to the dropdown, and give it an id. Add the three entries “Today”, “Tomorrow”, and “Thursday” to the dropdown element. (3 P)

Requirement 4: Create one block level element of the class *movie* (later containing the movie information) inside the *content* area: (1 P)



Big Eyes
Biography, Drama
PG-13 106 min.
10:15, 16:45, 19:30

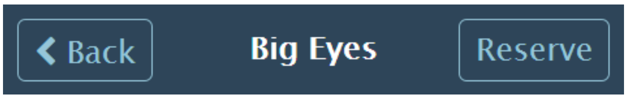
Requirement 5: Create six separate elements for *image*, *title*, *genre, info* (rating and duration), *schedule* and *more* (i.e. the arrow) inside the *movie* box. Use dummy text and one of the poster images in the folder “images/posters”. Each of the elements need to be addressable via CSS/JS (keep in mind, that there will be multiple automatically generated *movie* boxes). (3 P)

Requirement 6: Add the three icons by using the following Font-Awesome classes (present in the already included Font-Awesome library, http://fortawesome.github.io/Font-Awesome/): (2 P)

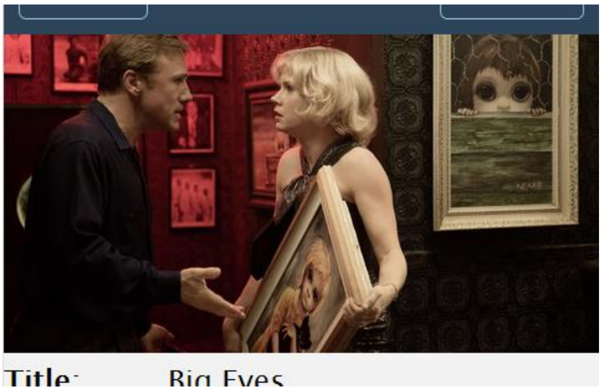
- *fa-clock-o* (for the duration icon)
- *fa-calendar-o* (for the *schedule* times)
- *fa-chevron-right* (for the *more* arrow)

Requirement 7: Your app has to consist of only one HTML document (containing the overview and the details view). Add the second details page (with corresponding class & id) in your body after the previous one (1 P)

Requirement 8: Include another title bar in the details page. Add a dummy movie headline. Add the two buttons *back* and *reserve* and assign the classes *control*, *left*, or *right* respectively. Add the back arrow using Font-Awesome. (3 P)



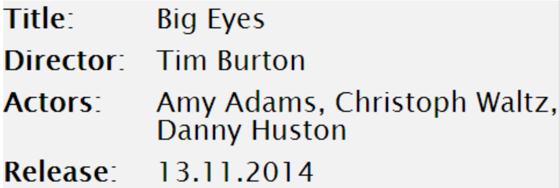
Requirement 9: Add an empty div (not an img, the div does not containin an img either) for the movie photo and assign an id: (1 P)



Title: Big Eyes

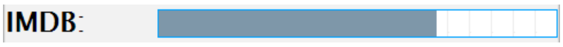
Requirement 10: Add elements for the additional information (e.g. “Big Eyes”) and the corresponding labels (e.g. “Title:”) to the content area of the details view (all need to have an id):

(2 P)

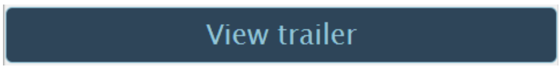


Title:	Big Eyes
Director:	Tim Burton
Actors:	Amy Adams, Christoph Waltz, Danny Huston
Release:	13.11.2014

Requirement 11: Add a progress bar for displaying the IMDB rating, set a maximum rating of 10, and apply an id. (1 P)



Bonus Point: Add a view trailer button. Reuse the class of all other control elements. (+1 P)



CSS Layout & Style (50 Points)

Colours

Use the following complementary colour theme:

dark cinema blue	*light cinema blue*	*cinema grey*
RGB: 46, 69, 89	RGB: 150, 198, 217	RGB 242, 242, 242
#2E4559	#96C6D9	#F2F2F2

Sizes

For all sizes and distances, use *multiples of 12 pixels!*

Layout

For modifying the styles, use the supplied "style.css" file!

Requirement 12: Position & style both *pages* (select their corresponding class) (5 P)

- Their position is absolute
- They are placed topmost/leftmost
- They cover the full screen size
- The background is *cinema grey*

The pages should now be overlapping (details hide overview)

Requirement 13: Position & style both *title* bars: (5 P)

- They should be bound to the top of the screen and fill the whole width
- The inner distance (to buttons/headline) is 12px
- Their overall height (including margin, border, padding) is 72px
- Background is *dark cinema blue*, title text is *white*.

Requirement 14: Layout the two *content* areas (3 P)

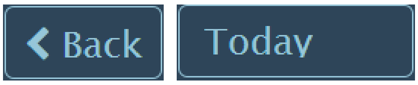
- The content should start below the title bar and fill the rest of the screen
- If the content does not fit, there should be a vertical scroll bar (only for the content area)

Style Title Bar & Controls

Requirement 15: Centre the headline text horizontally and vertically inside the title bar (2 P)

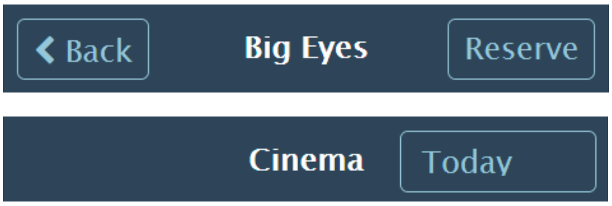
Requirement 16: Style all controls (i.e. buttons and select): (6 P)

- Add a border (1 pixel line in light cinema blue)
- Text colour (also light cinema blue)
- Background colour (dark cinema blue)
- Rounded appearance (6 pixel)
- Adjust the spacing between text and button (12 pixel)
- They have a height of 48 pixel



Requirement 17: Ensure, that the left and right title bar controls (using their classes): (3 P)

- Are displayed left/right (respectively) within their toolbar
- Have an outer distance (12 pixels) to the toolbar

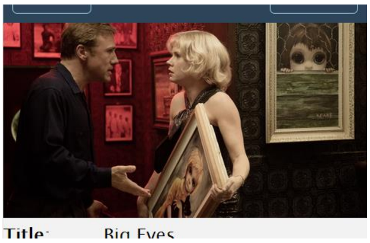


Style the Details View

Requirement 18: Add a details photo to the empty div as a background image (find photos in "images/photos"). Apply an inline style to the HTML element (as this will later on be done programmatically, it shouldn't go into the global stylesheet). (2 P)

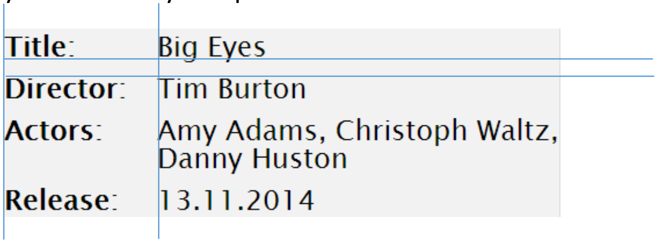
Requirement 19: Continue styling the details photo in your external stylesheet: (3 P)

- The height of the details photo should be 256 pixels
- Centre the image inside the div
- The image has to cover the div



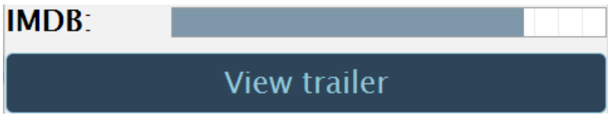
Requirement 20: Layout and style the additional information: (3 P)

- Spacing between entries is 12 pixels
- Labels are bold
- Information is always indented by 132 pixels.



Title:	Big Eyes
Director:	Tim Burton
Actors:	Amy Adams, Christoph Waltz, Danny Huston
Release:	13.11.2014

Requirement 21: Progress bar and button have to fill the full parent width (1 P)



Hide Details

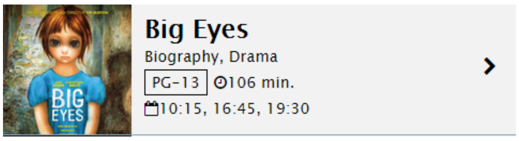
Requirement 22: Hide the *details* view (2 P)

- Select only the details view using its id
- Hide it using the display property!

The layout of both *title* and *content* areas as well as the *control* styles should now also apply for the *overview*!

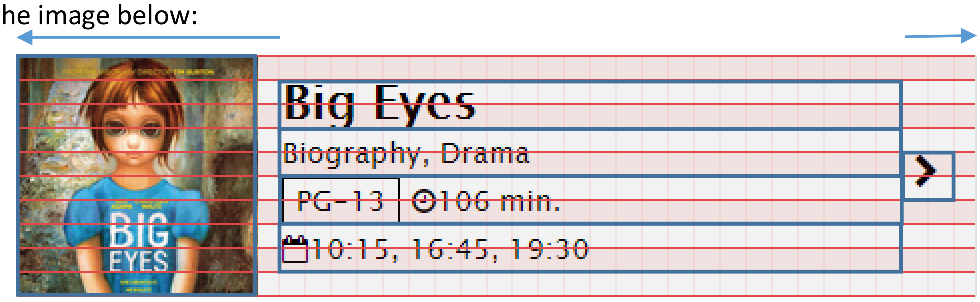
Format the movie items

Requirement 23: Style the *movie* item box to match the following requirements: (4 P)



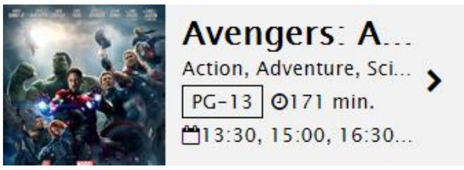
- Height: 120 pixels
- Items are separated by a *light cinema blue* line
- Elements, which overflow the box (e.g. images larger than 120 pixels) are hidden
- Add a relative positioning

Requirement 24: Align and style all elements in the movie item box according to the 12 pixels raster given in the image below:



- Directly place all six elements (*image*, *title*, *genre, info* [rating and duration]*, schedule* and *more* [i.e. the arrow]). (1 P)
- Place & align the image left; place & align the text elements left & right; place & align the more arrow right. (3 P)
- Size the image element to its width. The height should adjust automatically (and thus be cropped) (2 P)
- The title is bold (1 P)
- Genre, info, and schedule are 60% of the default font size (1 P)
- The rating has a border with a left/right distance of 6 pixel (2 P)
- There is a 6 pixel space between the rating and the duration (1 P)

Bonus Points: Format *title*, *genre*, *information* and *schedule*, to crop the content and display an ellipsis (…) if not enough space is available: (+3 P)



JS/jQuery Interaction (37 Points)

For the interaction part, you may generally use JS and/or jQuery.

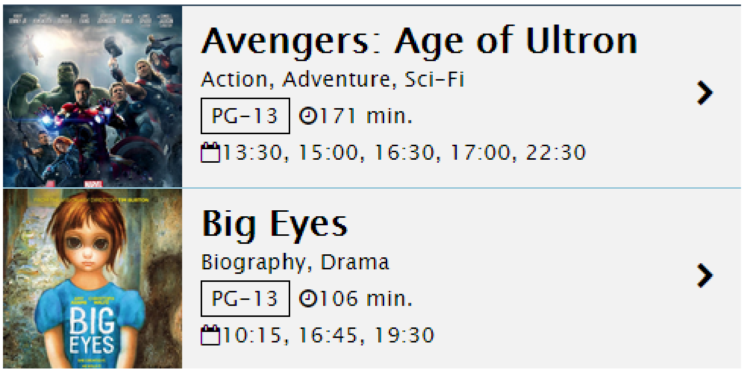
Have a look at the objects provided in "cinema.js"

Use the file "javascript.js" for your interaction code.

Requirement 25: The overlay can be opened and closed.

- Add an event handler to the *movie* item box to show the *details* view when clicked. (2 P)
- Hide the details view when the back button is clicked. (1 P)

Requirement 26: Multiple movies are written programmatically into the content area



Avengers: Age of Ultron
Action, Adventure, Sci-Fi
PG-13 171 min.
13:30, 15:00, 16:30, 17:00, 22:30
›

Big Eyes
Biography, Drama
PG-13 106 min.
10:15, 16:45, 19:30
›

- Create a `function listMovies()` which loops through all items of the global `var movies = […]` (given). Call `listMovies()`. Debug the processed movie's title to the console! (6 P)
- Call `listMovies()`, once the DOM is ready (or body has loaded, …) (1 P)
- Create a `function createMovieItem()` which accepts a `movie` as an argument. The movie will be one of the JavaScript objects in the `movies` array. (1 P)
- Create the *movie* div programmatically in `createMovieItem()` (this can be a string representation, a DOM element, or a jQuery element) and return it to the caller. (2 P)
- Call `createMovieItem()` the in `listMovies()` using the respective movie as an argument. Accumulate all the results throughout the loop. Add all movies to the content section of the overview. (4 P)
- Create the proper elements (containing the `movies` object's information) for *image, title, genre, info* (rating and duration), *more* (i.e. the arrow) inside the *movie* box (5 P)

Requirement 27: Clicking a movie item shows the corresponding details view

- Use a `function showDetails(movie)`, which, besides showing the *details* view, writes the corresponding movie information to it (passed via the parameter `movie` as an JS object):
  - Title (title bar and title information) (2 P)
  - Photo (correct source) (1 P)
  - Director (1 P)
  - **Comma-separated** actors (1 P)
  - Release date as **locale date string** (2 P)
  - The IMDB-Rating as progress-bar value (1 P)
- Add an event listener in `createMovieItem()` to the movie item, which calls `showDetails()` with the respective movie (3 P)

Requirement 28: The scheduled times are correctly applied

- Showtimes from the first array element (i.e. today) of the given global `var schedule = [..]` are correctly added to the corresponding movies in a comma-separated form. (3 P)
- If no showtimes are planned for the day, the movie item is not listed (i.e., "today" only "Avengers", "Big Eyes", "Ex Machina", and "Shaun the Sheep" are shown). (1 P)

Bonus Points The date-switcher works, selecting another day updates the movies accordingly. The listener is added programmatically (not in HTML). The day is passed to the modified `function listMovies(day)`. (+5 P)

Bonus Points: Add *slide* animations to the show/hide details functionality: (+2 P)

- Open details: details shift in from the right side, while overview shifts out to the left side
- Close details: Details shift out to the right side, while overview shifts in from the left side