Class: IS219 – Advanced Website Development

Assignment: Project 3

## Starting and Submitting your Files

1. Download the Project 3 (P3) ZIP file from Moodle.
2. Move the contents of the ZIP file to your computer's local web directory.
    a. If you don't have a web directory, you can install an "AMP" stack for Apache (web server, MySQL and PHP. I recommend AMPPS (www.ampps.com).
    b. Once you get your AMP stack running, you can visit your site using http://127.0.0.1/
    c. You should be testing your code on a real web server such as http://127.0.0.1/, NOT on a local file such as file:///Applications/AMPPS/www/p3/index.html or on Heroku (you should only be pushing good, tested code to Heroku).
    d. Please do not edit your files directly on github.com. You should be using a separate code editor (e.g., Sublime Text) to edit your code. You should be committing using the Github Desktop application or the command line tools.
3. Using your Github Desktop application, create a *new* project using your P3 web directory from Step 2. Please do not reuse a past directory and do not make multiple repositories for this project. I recommend using the following project name: *is219lastname-p3*
4. Connect your Github repository to a new Heroku app. Please do not reuse a past Heroku app name. You will submit this link on Moodle. I recommend using the following name: *is219lastname-p3.herokuapp.com*

## Notes

- Please commit to your Github repository often to keep track of your code and to clearly show your progress. **YOU WILL BE GRADED ON THIS (if there are insufficient or unclear commits, a large percentage will be deducted from your grade).** This is to make sure you are making progress on your own in incremental steps. I recommend committing to Github after testing and finishing each step in the instructions (and not for untested code). When committing changes to Github, use the "best practices for commit messages" (http://chris.beams.io/posts/git-commit/). You should use the command line tools or Github Desktop application to commit your files. You should *not* be editing files directly from the github.com website.
- You will **not** receive any credit for implementing anything using *jQuery*. Everything should be implemented using AngularJS where possible.
- Please do not commit updates to Github and/or push to Heroku *after* the project deadline. Timestamps on Github will count as the submission time, so updating after the deadline will be considered late.

## Legend

- To guide you through the assignment, the following colors correspond to file you should be editing in the "Instructions" sections.
  - index.html – text is in green
  - HomeController.js – text is blue
  - style.css – text is in orange

## Examples

- You can see example images of how things should look in the examples/ folder

## Instructions

1. Include the AngularJS javascript into the <HEAD> area of your file.
   (Your controller and module files are already loaded at the bottom of index.html).
2. Add "ng-app" and "ng-controller" attributes to the correct HTML tags so that your data-binding and expressions will work (so your Angular expressions show correctly on your page).
3. Create the following variables inside your HomeController controller:
   A. *title* – the value should be "IMDB + *YourName's* Top 8 Movies".
      Example: IMDB + Michael's Top 8 Movies
   B. *owner* – the value should be set to your name.
      Example: Michael
   C. github – the value should be set to your Github repository for P3.
      Example: http://www.github.com/mjslee/is219lee-p3

4. Fix the syntax for the data-bindings you created in the prior step using Angular code
   (Change the %something% placeholders to use double curly braces).
   A. #app_title should display the controller's *title* variable's value.
   B. #github should display the github icon, the controller's *owner* variable, and create an link to the controller's *github* variable's value.
5. Fix the syntax for the data-bindings to display a movie's information using Angular code
   (Fix the %something% placeholders & refer to HomeController.js for correct variable names to use).
   A. .poster should display the poster using Angular and the value from movie[0].poster[0]
   B. .imageInfo should display the value from movie[0].posterIndex
   C. .title should display value from movie[0].title and movie[0].country on the same line, with the country enclosed in parentheses: Example: The Shawshank Redemption (USA)
   D. .score should display a star icon and the value from movie[0].iscore
   E. .runtime should display the value from movie[0].runtime
   F. .released should display the value from movie[0].released and movie[0].rating on the same line. movie[0].released should be sent through the Angular "date" filter.
      Example: Nov 14, 1994 | Rated: R
   G. .websites should be Angular links using movie[0].imdb and movie[0].website on the same line, with the text IMDB and Official, respectively: Example: IMDB | Official
   H. .likes should display a thumbs-up icon and the value from movie[0].likes
   I. .dislikes should display a thumbs-down icon and the value from movie[0].dislikes
6. Create a loop using Angular code to display all of the movies in your HomeController.js file. You will have to modify the code slightly from Step 5 .
7. Create the following functions inside your HomeController controller:
   A. *like(index)* – a function called "like" that accepts the parameter "index" (Number). When called, it should increment the variable "like" variable by one, for the movie at the given "index". (These do not have to save when you refresh the page).
   B. *dislike(index)* – a function called "dislike" that accepts the parameter "index". when called, it should increment the variable "dislike" variable by one, for the movie at the given "index" (Number). (These do not have to save when you refresh the page).
   C. *posterClick*(index) – a function called "posterClick" that accepts the parameter "index" (Number). When called, it should increment posterindex, for the movie at the given "index", and show the poster at the updated posterindex index. **Once it reaches the last image of the posters array, it should loop back to the first index (index 0).**
   D. *timeText(minutes)* – a function called "timeText" that accepts the parameter "minutes" (Number). It should convert "minutes" (an integer) to a String of hours and minutes. For example, "timeText(61)" should return "1h 1m", "timeText(145)" should return "2h 25m", and "timeText(180)" should return "3h 0m".
8. Add a Angular click handler to:
   A. .likes to call the like() function in HomeController.js, passing $index as an argument ($index is a built-in variable that is used by Angular loops). This should change the displayed value in .likes

   B. .dislikes to call the dislike() function in HomeController.js, passing $index as an argument ($index is a built-in variable that is used by Angular loops). This should change the displayed value in .dislikes
   C. .poster to call the posterClick() function in HomeController.js, passing $index as an argument ($index is a built-in variable that is used by Angular loops). This should change the displayed .poster for the movie (if there is more than one poster).
9. Update .imageInfo to display the number of the *current poster* and the *total number of posters* for each movie. This should change every time a poster changes (if there is more then one poster for that movie). Example: Poster 2 of 3
10. Update .score to show as a circle instead of a square.
11. Add a green background color (#008700) and white text (#ffffff) to your .likes button.
12. Add a light-green background (#9edd87) and green text (#008700) to your .likes button on mouseover/hover.
13. Add a red background color (#f65a5b) and white text (#ffffff) to your .dislikes button.
14. Add a light-red background (#fdd6d6) and red text (#F65A5B) to your .dislikes button on mouseover/hover.
15. Add a smooth transition (500ms) when hovering on and off .likes and .dislikes
16. Add THREE of *your* favorite movies to the list and make sure they all display correctly on your page (so a total of 8 movies will show on your page: 5 original movies + 3 new movies).
   A. Each of your three favorite movies should have all of the necessary metadata and they should be accurate (you can probably get the real information from IMDB.com). The "like" and "dislike" numbers can be fictitious (you can choose any number for these).
   B. You should have at least TWO posters for each of your movies. They should be large enough to fit nicely into your existing movie listing. The recommended size is: 500x750 pixels. Add posters to the img/ folder.

| | **Project 3 - Rubric - Note, steps *must* be implemented using AngularJS where possible** | **Points** | **Your Points** |
|---|---|---|---|
| 1 | Did you properly use Github to track your changes (with appropriate commit messages)? | (see below) ↓ | |
| 2 | Did you properly deploy your project to a Heroku/AFS server? | 3 | |
| 3 | Did you correctly add AngularJS to your site? (include in html file head) | 2 | |
| 4 | Are your AngularJS attributes (e.g., ng-app, ng-controller) added correctly to HTML tags? | 3 | |
| 5 | Are the title, owner, and github variables correctly initialized and contain the correct values in HomeController.js? | 6 | |
| 6 | Is HomeController.js's title variable correctly used and shown in index.html? (e.g., "IMDB + Michael's Top 8 Movies") | 2 | |
| 7 | Is the Github button (in index.html) shown correctly with the Github icon, your name, and links to your Github P3 repository when clicked (using the values from HomeController.js)? | 4 | |
| 8 | Does a poster image show for all movies in HomeController.js (using a loop with Angular code)? (Including the default 5 movies and your 3 favorite movies, a total of 8 movies should be displayed). | 8 | |
| 9 | For a movie with multiple posters, does the poster change to the next one when clicked? Does the displayed poster loop back (after clicking) to the first poster after showing the last one? | 10 | |
| 10 | Does the "Poster X of Y" text under the poster change correctly when you click on the poster? (e.g., "Poster 1 of 3") | 4 | |
| 11 | Does a star icon and IMDB score show in a green *circle* overlaid on each movie's poster? | 4 | |
| 12 | Is the title and country displayed correctly for each movie? (e.g., "Zootopia (USA)") | 2 | |
| 13 | Is the runtime displayed correctly using the timeText() function? (e.g., "Runtime: 1h 48m") | 6 | |
| 14 | Is the release date formatted correctly (sent through Angular's date filter) along with the MPAA rating? (e.g., "Released: Mar 4, 2016 l Rated: PG") | 5 | |
| 15 | Are the IMDB and Official links labeled correctly and clickable to their respective websites? (e.g., "Websites: IMDB l Official") | 4 | |
| 16 | Do your "like" and "dislike" buttons have a thumbs-up and thumbs-down icon (respectively), along with the current like and dislike number displayed? | 4 | |
| 17 | Is your "like" button green with white text, and then light-green with green text on hover? | 4 | |
| 18 | Is your "dislike" button red with white text, and then light-red with red text on hover? | 4 | |

| | **Project 3 - Rubric - Note, steps *must* be implemented using AngularJS where possible** | **Points** | **Your Points** |
|---|---|---|---|
| 19 | Do your "like" and "dislike" buttons fade smoothly (500ms) in and out to their final colors on mouseover and mouseout? | 4 | |
| 20 | When clicked, does your “like” button’s text increase by 1 and display it? | 6 | |
| 21 | When clicked, does your “dislike” button’s text increase by 1 and display it? | 6 | |
| 22 | Did you add three more of your favorite movies to the HomeController list? (This should include at least 2 correctly sized poster each (preferably 500x750 pixels), title, runtime in minutes, release date, rating, IMDB website, official website, IMDB score, poster index set to 0, and at least 0 likes and 0 dislikes). | 9 | |
| *DEDUCTION: -10 per day late (i.e. 10% deduction per day).* | | ← | |
| *DEDUCTION: Lack of or unclear commits; untested code commits (0-90% deduction).* | | ← | |
| **TOTALS:** | | **100** | |