CSCI 342, Fall 2016, Homework # 3

Due date: Wednesday, November 2, midnight. Please ZIP all your files together into one archive! There will be lots of files for this assignment.

Instructions: Create two files to recreate the web page seen in Figure 1: `tomayto.php` and `tomayto.css`. This webpage will have the addition of a single input form at the top, allowing the user to pick the movie from a drop-down list of all available movies, and having that movie displayed when the user clicks the "Select Movie" button.

Your page does not have to match mine perfectly, in fact, it should match your solution to homework #2. You should take that file and modify it so that all of the dynamic information in the web page comes from files and computations on those files performed in php on the server. You should not have to change your CSS file at all, except to add some styling to the movie selector.

Again, format your html, php, and css to be as readable as possible.

Place a comment header in each file with your name, a brief description of the assignment, and the file's contents.

Movie files: I have provided a zip archive of several movie files (actually, these were created by the textbook authors, but I modified them slightly). Your program should handle any number of movies, each in its own subfolder of `moviefiles`. I will add one or two more movies when testing your page, so do not hard-code the movie names, use a `glob` pattern.

Each subfolder will consist of the following files:

info.txt A file with three lines of information about the film. We will only use the first two, the title and the year. For example:

```
The Princess Bride
1987
95
```

overview.txt A file with information about the movie to be placed in the General Overview section. Each line starts with a title for that line, followed by a colon, and then the information for that line. There are no line breaks other than between items.

These items are to be displayed as a definition list with a definition list term `(dt)` and its description `(dt)`. The number of lines in the file varies from movie to movie. Example:

```
STARRING:Cary Elwes, Robin Wright, Andre the Giant, Mandy Patinkin
DIRECTOR:Rob Reiner
PRODUCER:Arnold Scheinman, Rob Reiner
SCREENWRITER:William Goldman
RATING:PG
RELEASE DATE:September 25, 1987 (USA)
RUNTIME:98 min
SYNOPSIS:Director Rob Reiner breathes vividly colored cinematic life into William Goldman's THE PR
RELEASE COMPANY:20th Century Fox
```

overview.png The image to display at the top of the General Overview section. This image will be of size 250×412px.

review1.txt, review2.txt, ... Files containing information for each review of the film. Each review contains exactly four lines: the review, the number of tomatoes (1-4), the reviewer's name, and their affiliation. For example:

```
One of Reiner's most entertaining films, effective as a swashbuckling epic, romantic fable, and sa
4
```

```
Emanuel Levy
emanuellevy.com
```

Different movies will have different numbers of reviews. Show half of the reviews in the left column, and the other half in the right (an extra review goes in the left column). Do not worry about the possibility that the columns may be very different in height. You may assume that every film has at least one review, but you do not know the maximum. Don't hard-code review names, use a `glob` pattern.

All the movie files should be in a folder called `moviefiles` in the same folder as your `tomayto.php`. So, for example, the complete path to the Princess Bride image should be (starting from the same folder as your php file):

`moviefiles/princessbride/overview.png`

Your Own Movie: As part of your turnin, create your own set of movie input data for this page. Write `info.txt`, `overview.txt`, and at least four review text files for your movie. Also find a suitable image to use as `overview.png`, of size 250×412px.

Server side computations: You will, of course, have to read data from the files to prepare each page. You will also have to do some processing to compute some of the content. For example, you will have to find the *average tomato ranking* to put as the giant red number at the top of the page. You will do this by running through all the reviews, finding the number of tomatoes and the number of reviews, and calculating the average.

Select movie dropdown: You will also have to find each of the movie folders in order to populate the dropdown selection next to the average tomato rating. This should be a form with a `select` input in it, with each `value` being the `folder name` for the movie, and each text in the selection being the actual movie name (found in the `info.txt` file).

Further, initially the page should load the "Princess Bride" movie, and this should be selected in the drop-down menu. Thereafter, the *selected* movie in the drop-down should always be the movie selected.

You will also need a `submit` button in the form containing the drop-down menu. Its action should be the same page, `tomayto.php`. We could make this a little nicer by submitting on selection, but that would require javascript and that's forbidden for this assignment. Your php, therefore, needs to consult the `$_GET` variable to determine which film to display (with the default of "Princess Bride" if there is none selected).

No javascript: There should be *no* javascript used in this program. The entire appearance of the page should be created on the server in php, css, and html.

Validation: For full credit the page's output (not the php file) should pass the W3C html validator, and the css page should pass the CSS validator. To validate the output, view your page, then choose "View Source", then copy and paste that code into the validator.

Style: Your PHP code should not produce any warnings or errors. Programming style will be considered in your grade. Do not use any `global` variables. Produce as much of your output as possible in HTML mode as possible, without `print` or `echo` statements. Use functions when they seem appropriate.

Put descriptive comments in your PHP code where appropriate.

Figure 1: Screenshot of final webpage.