CSC 337 Final Project: Rancid Tomatoes Website

You have the option to create a final project of your own choosing if so desired. See the expectations at the end of this document. If you want a more specific spec starting with some of the work you have already completed, you may also build a more interesting Rancid Tomatoes website specified next.

The major functionality of the Rancid Tomatoes allows anyone to find an existing movie review by selecting the one they want to view. Also, anyone can create an account at which time they will be able to add reviews to any existing movie. A logged in user may also add new Movies with no Reviews at first **OR** Edit their own reviews (your choice). *Note: Specifications are ALWAYS Subject to Change*

Home Page (name it index.html)

- Allows anyone to find a movie review (no login required)
- You must be able to search movies in the database of movies or have a list of movies, a suggestion feature, your design
- Advertise a current movie (your choice)
- Have a way for users to select the Login/Register view
- Have a way to add for logged in users to add a review to an existing movie.
- Have a way to add for logged in users to add a new Movie (need about 7 input fields)

Review Page (should appear like the review page you finished)

- Show rating, movie poster, overview (title, director, year, rating, runtime in minutes, box office revenue that could be in the millions, and all reviews

Add Review Page

- Only accessible if the user is logged in
- Must not be allowed to submit an empty review, but can cancel the effort
- Must escape html chars and avoid sql injection
- Must redirect back to the movie page

Add New Movie Page OR Allow Logged in Reviewers to Edit their own reviews

- Only accessible if the user is logged in
- Need to input title (key), image_file_name, director, rating (G,PG, R), year released, runtime in minutes, boxOffice (number of tickets sold) OR allow editing of a review
- Must escape html chars and avoid sql injection
- Must redirect back to the movie page

Login and Registration view

- Must be able to login with a user name and password
- Notify user if they could not be validated, could be just one message such as Invalid Login
- Must be able to register a new account with a user name, password, first name, last name, and publication (put this to the left or right of the login inputs)
- Must validate input on the client: both account names and passwords can be any letter or digit
- Password must have a length of 8 or more.

Security and Design

- Must be able to validate input on the server (prepared statements, html_specialchars, valid user, valid password).
- Uses Model View Controller MVC (design).
- You have a database with at least these three tables: Reviewers, Movies, Reviews

Optional Final Project of your own design

- Must convince us it is enough for a three week project
- Must have a data base with at least 2 tables, be able to log in or create a new account, use HTML, use a style sheet, use PHP, use JavaScript, employ AJAX
- Possibilities: Multi-User Chat System, A Reservation System (view things to reserve, simulate reservations, Small purchasing web site (shopping cart, very limited inventory)
- A single player games like: Boggle, Sudoku, or Hunt the Wumpus
- Board game from Bill Mitchell: "Chess, Checkers, Othello, Backgammon, and Scrabble come to mind immediately. This is not an AI class so I picture a two-player mode where the game just keeps track of state.
    - A game in progress should be able to be saved, to be resumed at a later time.
        - New since a Piazza question: Would it make sense to simply save all games? Or perhaps not save games, just win and lost records perhaps. Talk to your project manager
    - There should be a way to view the game move-by-move.
    - Any number of games should be able to be in progress at any point in time.

Added 20-Nov: Grading Criteria for all Projects *subject to change*

Note: the first 35 points are available only for projects that make a serious attempt at the solution. If you turn in 6 files with unrelated or incomplete code, you will receive 0/100

___/ 10 used all 5 languages: HTML, CSS, JavaScript, PHP, MySQL
___/ 5 used a separate style sheet
___/ 10 used AJAX somewhere (we can search for XMLHttpRequest onreadystatechange and/or verify it is doing some client side processing to change part of a page without reloading the page
___/ 10 data base must have at least two tables (joins not required in all projects, use a table even if an array is better)
___/ 5 requires login to use your system and it works (need an accounts table)
___/ 5 can register new accounts and it works (need an accounts table)
___/~~45~~ **55** must do what is supposed to do (subjective, but matching above spec or your clarified proposal)
~~___/ 10 WOW factor: A normal working project may only get 90% max (subjective)~~