**585347**

11

**Movie Review Website**

**An analysis, design, implementation and evaluation of a movie review website**

**Kieran Foxley-Jones**

Contents

[Proposal 3](#_Toc287806664)

[Title 3](#_Toc287806665)

[Introduction 3](#_Toc287806666)

[Key Phrases 3](#_Toc287806667)

[Objectives 3](#_Toc287806668)

[Required Resources 3](#_Toc287806669)

[Analysis 4](#_Toc287806670)

[Rottentomatoes.com 4](#_Toc287806671)

[Review System 4](#_Toc287806672)

[Registration System 5](#_Toc287806673)

[Login Page 5](#_Toc287806674)

[User Profile 6](#_Toc287806675)

[Design – Prototype 1 8](#_Toc287806676)

[General Layout 8](#_Toc287806677)

[Registration 9](#_Toc287806678)

[Login 9](#_Toc287806679)

[Cookies and Session Control 9](#_Toc287806680)

[Database design 9](#_Toc287806681)

[What Is A Hash? 9](file:///I:\Dropbox\EasyPHP-5.3.5\www\MovieReview\Word\Proposal.docx#_Toc287806682)

[Psudo Code 11](#_Toc287806683)

[Registration 11](#_Toc287806684)

[Bibliography 12](#_Toc287806685)

[Work Log 13](#_Toc287806686)

# Proposal

## Title

A website that takes user submitted movie reviews and stores them for other people to read.

## Introduction

The project will concern itself with the development of a movie review website, it will include elements of database design, client side and server side scripting, and general web design. There are many site dedicated to movie reviews, but many of these sites emphasise the professional critics point of view, this review site focuses on the average persons experience with a particular film. The finished project would be concerned with its own information and stored data; it should also allow the possible expansion to include off site data. This project will also require me to learn how to create many aspects of the site, for example the querying of a database the use of information queried to generate a new webpage for viewing by a user.

## Key Phrases

Databases, Normalisation, PHP, Javascript, Scripting Security Issues, Website readability.

## Objectives

* To evaluate competitive review websites, and establish what features are critical to similar sites, and what my particular site can do differently to compete.
* Ensure the storage of information is secured, especially personal details and security information (username/passwords)
* Usage of database queries to store, search and categorise user submitted reviews.
* Use of server-side scripting to generate pages from user selection.
* Output of generated pages in a readable format

## Required Resources

PHP, MySQL, Multiple web browsers (Chrome, Firefox, IE), Web server hardware and software (WAMP, XAMPP).

# Analysis

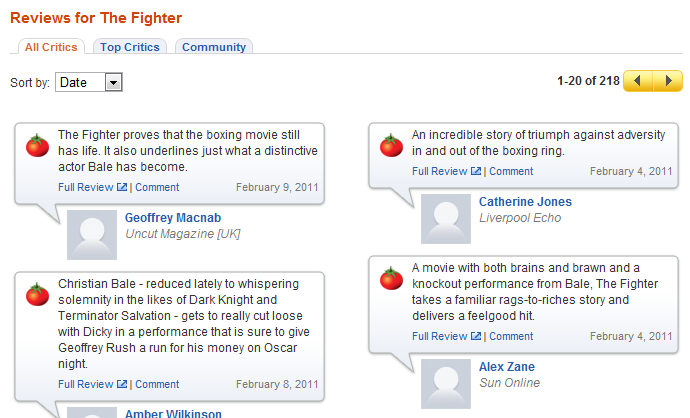
There are a number of other movie review sites that already take and output reviews created by the general public.

## Rottentomatoes.com

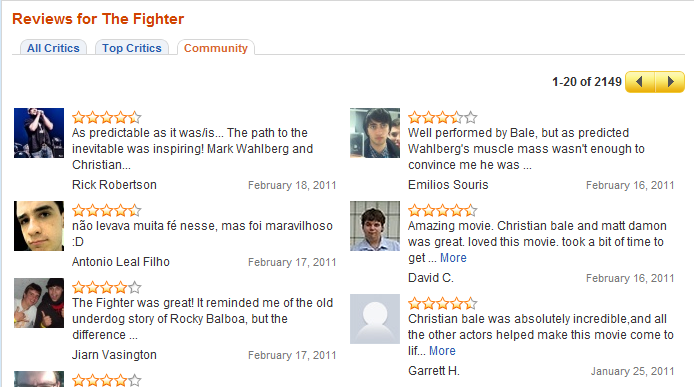
Rotten Tomatoes is one of the biggest review sites on the internet, people tend to refer to this site when they are talking about new or recent movies. A lot of message boards and forums I have researched, such as Something Awful and Reddit, use rotten tomatoes’ fresh and rotten marker as a general quality marker in discussions of movies.

### Review System

Rotten Tomatoes takes an aggregate of review scores from multiple sources, and displays it along with the subject movie, as either fresh or rotten with their percentage mark. These sources include professional reviews from published critics, along with user submitted ones.



When you choose a particular movie, you get a multitude of information, with links to the full reviews of the professional critics and a short summary on the page itself. This allows you to quickly ascertain what professional critics say about the movie, and also head straight to their published review. This gives a nice idea on what you may like about it

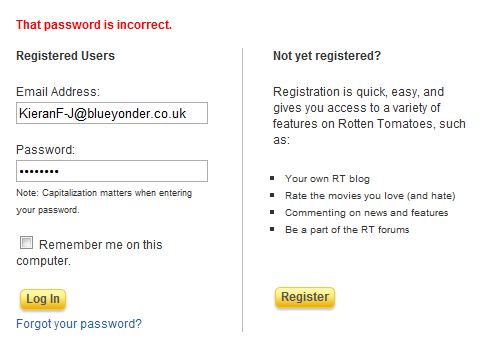


You can also get links and summaries of community posted reviews in a separate tab and what they thought in multiple forms (star rating, words). You can read these community reviews just like the critic reviews. Clicking through into these reviews takes you to the selected persons profile, where you can see what other films they have commented on, which is an idea to consider for my site.

### Registration System

Rotten Tomatoes registration page is simple, and typical for many websites. You enter your email address twice for verification, again with passwords and your first name and last name. There is no username field as your email is used for logins. Gender, date of birth and Country are also included possibly for data collection reasons or some kind of script dedicated to sorting data retrieved when viewing a movie page (reviews by country, DOB etc).

Also present is a checkbox for the understanding of a user agreement and privacy policy. This is important when storing personal details and necessary should any legal or technical problems arise.

I can safely assume these details are stored within a database, and the password itself has some sort of security possibly using some form of hashing and salting.

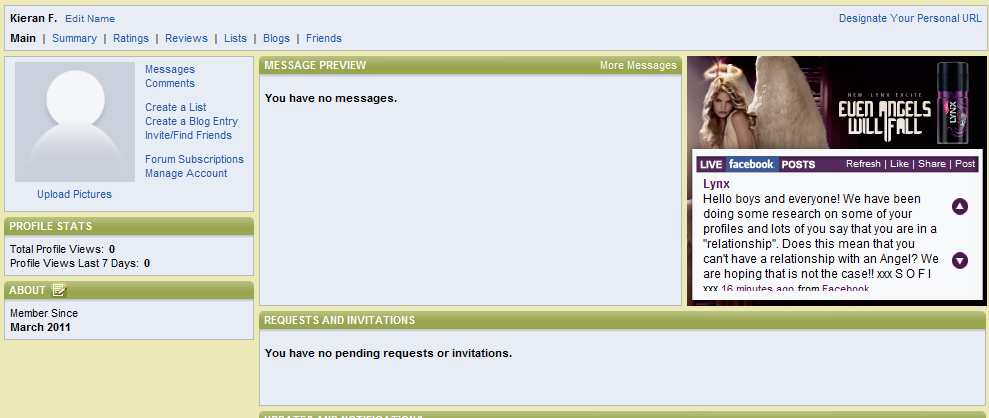
### Login Page

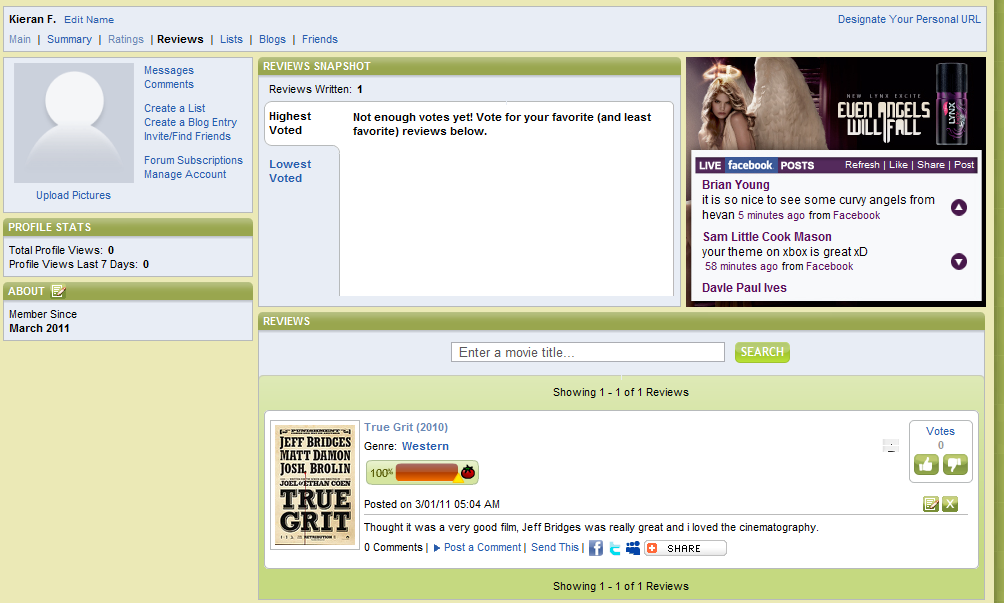
The login page is pretty simple, a simple well laid out login form, with “remember me” check box and a forgotten password link. Also on the page is a link to register an account, this allows the main front page to have less confusing links to a new user. When a new user arrives on the site and wants to make an account, they do not have to choose between a link straight to registration and a link to login.

I believe I can safely assume that this login page sends the user inputs to a PHP script, which possibly reverses the process used to provide security for the user passwords (hash, salt), and this hash will be stored into a secured database.

### User Profile

The user profile is pretty standard for sites like RottenTomatoes, it provides you with access to your public available information, reviews you have posted, statistics on ratings you have given, your own mini blog and access to friends list.



Any reviews you have posted show up on the reviews part of your profile, allowing you to edit or delete them at will. This system is also used for the friends reviews page, except you cannot edit or delete those reviews, but you can comment and vote up or down the review. This system is more than adequate for a website of this type, and it presents the information in a clean, accessible way.

**Posting a Review**

Moviereviews.com

Design – Prototype 1

The first prototype will be very simple and will concentrate on creating the bare essentials, the visual design of the site will be sparse, and barely there. The intention is to finish this prototype with:

* a working registration and login system
* session control so users can move from one page to another without repeated logins
* a basic review upload system allowing users to write reviews and save them to the server
* have a movie page output any reviews stored for it

## General Layout

The general layout of the first prototype will be a simple affair; it will initially be built to incorporate my intended layout with a header bar, small navigation bar below that, a left column and main content area, and footer bar. Specific sizes of these areas will likely change throughout the prototyping process, but the basic layout will stay as such.

The navigation bar will house links to login, registration and display whether you are already logged in or not.

## Registration

### What Is A Hash?

*A hash (also called a hash code, digest, or message digest) can be thought of as the digital fingerprint of a piece of data. You can easily generate a fixed length hash for any text string using a one-way mathematical process. It is next to impossible to (efficiently) recover the original text from a hash alone. It is also vastly unlikely that any different text string will give you an identical hash - a 'hash collision'. These properties make hashes ideally suited for storing your application's passwords. Why? Because although an attacker may compromise a part of your system and reveal your list of password hashes, they can't determine from the hashes alone what the real passwords are.*

(McGlinn)

This prototypes registration page will be simple, I do not intend to implement many security and validation features at this early stage, JavaScript validation on user inputs may come in future version, and the same can be said for a image based Captcha system to avoid repeated registration from automated external sources.

The registration form itself will be left in an unformatted state whilst I concentrate on the important task of creating essential systems for the website. Registration will consist of a Username and Password field, when the user submits the required information; it will be posted to a PHP script. This script will take the values for password, and apply a sha256 hash to it; it will also generate a random string of letters and use that as a salt for the now hashed password. This salt and hash combo will again be hashed and the result will be stored as the password within the database.

This provides some security to an outside attack, if someone gains authorised access to the database, all they will have available to them is a string of 64 characters which would not give any hints to what the original password is. Combine this with the salting of the password, and even brute force attacks, which know what the end hash is, would be problematic.

## Login

Login will appear to be a simple affair, with a login form available on all standard pages (index, movie list, movie specific etc) should a user not be logged in. The act of login will send the user inputted username and password to a login script, this script will search the user table of the database for the username. It will then take the password hash and stored salt and use them to check the user input password. The script will take the password, hash it, add the salt to this hash, and then hash again, just like in the registration script. This hash is compared to the stored hash and a decision is made to whether it is a valid password.

If it is valid then the script also creates a session and keeps the user logged in during their visit.

## Movie List Page

The movie list page will have a small script which queries the database for movie names and MovieID’s, it will then output this data as hypertext links. The movie name will make up the visual link, however the link itself will contain the movie ID, using php you can append to a url a php variable as so:

<a href="title.php?Mov\_ID=<?php echo $movieNames[$i]['Mov\_ID'];?>">

This will create a url of titple.php?Mov\_ID=1 if the Mov\_ID variable is 1, this allows the title page to pick up this variable using $\_GET[‘Mov\_ID’]; allowing the page to output dynamically. The movie list itself will have a loop outputting every single movie name and id stored within the movie database.

## Title.php page

This page contains the most potential content

## Review Submission

The form for submitting reviews will be located within the individual movie page, within the title.php page. The user will input their review into the available textarea box, and also choose a rating

Due to potential errors caused by punctuation marks combined with the SQL statements I will need to implement a function to parse the input string and escape any problematic characters, in this case apostrophes. The PHP function, mysql\_escape\_string(), parses the string a adds a \ to any apostrophes that need it, this stops the string contents from interfering with the SQL statement when it is passed to the SQL server.

## Cookies and Session Control

SQLHandler

The SQLHandler script was given to me by Kevin Woodard, it contains functions that automatically call a connection to the SQL database, with information stored within the script itself, it also contains functions to handle all types of queries that would crop up, also handling any errors that may appear. This allows you me to use one script file to connect to and query throughout the site, rather than having to retype database connection code and any error handling. This will make the code through the site easier to read for everyone.

## Database design

When a user posts a new review, an insert query inserts the body of content from the review into the review table, along with the user rating. This auto generates a Review ID number, stored in the Rev\_ID field. This is then retrieved through a select query and input into the link table, along with Movie ID (Mov\_ID) and User ID (User\_ID). This provides the necessary link between users, reviews and movies.

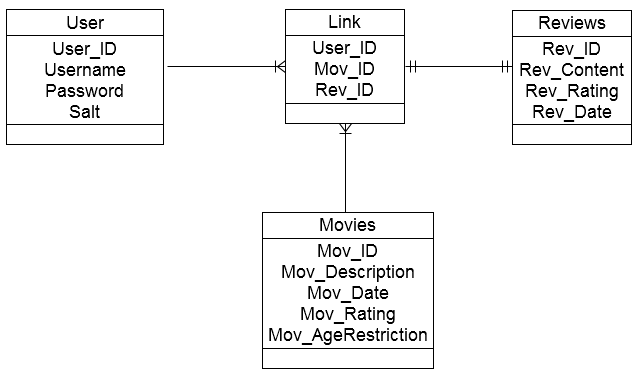


Figure ERD diagram

Software

PHP

MySQL

# Psudo Code

### Registration

Input Username and Password

Connect to Database

Create query (select password, salt where username = input username)

Send query to MySQL server

Receieve results

If number of results < 1 then

Error, no such user

Else

Hash user input password

Add salt to hash

Hash salthash

Check subsequent hash against hash received from database

If scripthash != databasehash

Send page to registration

Else

Create session,

Send username, password and user ID to session control

Send page back to index.php

End script

# Bibliography

McGlinn, J. (n.d.). *phpsec.org*. Retrieved from http://phpsec.org/articles/2005/password-hashing.html

## Work Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Item to do | Date Started | Date Completed | Work done | Sig. |
| Proposal | 15/02/11 | 15/02/11 | Created the proposal, identified the areas needed to focus the design on. |  |
| Analysis | 15/02/11 |  | Looked at a number of potential rival sites, outlined roughly how they work and made notes on what features work well and could be implemented into my site.  Layout Research – 15/3/11 |  |
| Design | 15/02/11 |  | Login, Registration, Movie list, Title.php design  Layout - |  |
| Development  Prototype 1 | 20/02/11 |  | Created Database 21/2/11  Created basic layout framework (css) 21/2/11  Created login and registration script 21/2/11  Created movie listing page 28/2/11  Created title.php page 7/2/11  Implemented mysql\_escape\_string() – 15/3/11  Layout – started 15/3/11  Implemented sqlHandler.php throughout 13/3/11 |  |
| Evaluation |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |