# Software Review(Handover)

## Current Architecture

### Search

The application in its current state is a fully functional movie search website with database connection. Movie information is accessed through a single PDO connection script using various SQL queries. Users can search for movies and view details such as classifications, year of release, etc.

### Top Ten Most Searched

The application contains a page which finds the ten most searched movies and displays them in a bar graph format alongside their list entries.

### Top Ten Rated Movies

As above, but with user scores.

### Rating System

Users can leave ratings on movies in the form of a score out of 5. Average scores are calculated and updated in the database. This function is an exception to the usual avoidance of letting users directly impact database contents; however, the scores have no effect on other components.

### Subscription

The application contains a function by which users can subscribe to either a monthly newsletter, newsflash notifications, or both. This is offered on a dedicated signup page that accepts the users’ name and email, and enters these details into a database table.

An unsubscribe function is also included. The form accepts the users’ name and email, and sets their subscription status to false. The change is approved by an administrator account.

### Login

The application contains a login form that checks for admin status of the user. Accounts with admin privileges can action user requests to unsubscribe from mailing lists and potentially make other changes to the database.

## Possible Enhancements

The following are a few suggested improvements for future expansion. Not currently planned or requested.

### Review-Bombing Protection

It should be noted that the application contains no protection against “review-bombing”, a practice in which large groups of people, or even one person with multiple accounts, leave numerous negative user reviews to alter review scores for various reasons. Note for future consideration.

### Additional User Functions

* Users logged in could make lists of their own favorite movies
* Polls to determine best movie of a certain genre (best action, etc.)
* Choice of UI colour scheme

### Additional Admin Functions

* Edit fields of movie entries (title, etc.)
* Ban users
* Check user review history

## Summary

The application is functioning as intended with all requested features implemented. Testing has not revealed any significant bugs or issues. A more-than-reasonable degree of WACG compliance has been achieved, considering such compliance has not been mandated for the project. Visual and functional design flow has been executed in a consistent and predictable manner. Following any final checks, the application in its current form is ready for deployment. Handover is set to proceed on schedule.

## WACG Compliance Report

### Introduction

The Web Content Accessibility Guidelines (WCAG) are established through a global collaborative process involving a range of organizations and individuals. WCAG represents a universal standard for accessibility of web content to those living with disabilities. WCAG documentation describes the methods of making content accessible to such individuals.

The following report details the areas of requirement for web accessibility to meet WCAG standards. Areas of possible improvement in the application will be identified and discussed where applicable.

### Perceivable

#### Provide text alternatives for non-text content.

The only pieces of non-text content in the project are the tables for the top-ten most searched and highest rated movies and both have tables attached below that shows the data in a text format

#### Provide captions and other alternatives for multimedia.

Two pages in the application (top ten most-searched and top ten rated) contain visual graphs to represent data. The graph also provides text detailing the contents of the graph including x and y axis, and each value and labels on each bar showing the values.

#### Create content that can be presented in different ways, including by assistive technologies, without losing meaning.

The content in the program can be read by most narrator programs with not much lost as a result.

#### Make it easier for users to see and hear content.

The text sizes are rather large and built to scale to the width of the screen so that they are still visible on any device. Also all text is displayed as text rather than an image containing text allowing for the webpage to be read by most narrator programs.

### Operable

#### Make all functionality available from a keyboard.

All page elements can be reached using the tab key and any data entry can have submit be triggered by the enter key allowing for full functionality of the website using only a keyboard

#### Give users enough time to read and use content.

All readable content is kept on screen until the user themselves chooses to click away to a different page.

#### Do not use content that causes seizures or physical reactions.

There is no content in the page that could carry a seizure warning as the only animated content is the generation of the 10 most searched graph which is done at a slow enough pace to not risk any form of seizure or reaction

#### Help users navigate and find content.

Every single page includes a navigation bar along the left side of the screen that allows for navigation to all relevant pages. This navigation bar does not include the page users are currently on at the time for the sake of avoiding constant looping.

#### Make it easier to use inputs other than keyboard.

All website elements can be selected via clicking and any data entry section that needs text can be done with an on-screen keyboard. (these text entry sections are often completely required eg: the name field for movie searching

### Understandable

#### Make text readable and understandable.

Text is clear and large to try and accommodate for being easily readable at any screen size

#### Make content appear and operate in predictable ways.

The program is designed to be visually standardized across the different parts of the program allowing for easier understanding and more predictable operation.

#### Help users avoid and correct mistakes.

Any mistakes that the website detects will be mentioned to the user in a way that explains to the user what the mistake was and a way to correct it.

#### Robust

After multiple testing runs both as development was done and at the end of every sprint multiple code paths have been designed so that the website should never break during normal use (the main way to break it would be to go to URLs only reached via POST commands however this wouldn’t affect the database and would only impact a user who isn’t using the website properly anyway. As such I would say that the program is robust enough to meet standards.

#### Maximize compatibility with current and future user tools.

Not many user tools are needed for this project so this section doesn’t fit as well despite that the project should be just as compatible as most basic websites available.