|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| V0 | 4/06/2020 | Sean Boaden | First Version |
| V1 | 10/6/2020 | Gholamreza Aminy | Sprint 2 requirements applied |
| V2 | 17/06/2020 | Sean Boaden | Sprint 3 requirements |

MAster document

JMC Project

***Team G***

Gholamreza Aminy

Sean Boaden

RAD Master Document

Contents

[Analysis and Setup 4](#_Toc44333938)

[SOURCE CONTROL 4](#_Toc44333939)

[Task Matrix 4](#_Toc44333940)

[Sprint One 5](#_Toc44333941)

[SOURCE CONTROL 5](#_Toc44333942)

[PROJECT MANAGEMENT PLAN 6](#_Toc44333943)

[TEST PLAN 7](#_Toc44333944)

[Introduction 9](#_Toc44333945)

[Test Methodology 10](#_Toc44333946)

[Resource & Environment Needs 12](#_Toc44333947)

[Terms/Acronyms 12](#_Toc44333948)

[ANALYSIS REPORT 13](#_Toc44333949)

[CITE business rules for software development 13](#_Toc44333950)

[CITE Managed Services Quality Assurance 14](#_Toc44333951)

[Acme Entertainment Pty Ltd development requirements 14](#_Toc44333952)

[MULTI-PLATFORM REPORT 16](#_Toc44333953)

[Multi-Platform Tests 17](#_Toc44333954)

[Sprint Two 22](#_Toc44333955)

[SOURCE CONTROL SNAPSHOT 22](#_Toc44333956)

[UPDATED PROJECT MANAGEMENT PLAN 25](#_Toc44333957)

[UPDATED SOFTWARE REVIEW PLAN 26](#_Toc44333958)

[Purpose 26](#_Toc44333959)

[Verification 26](#_Toc44333960)

[Sprint Requirements 26](#_Toc44333961)

[System Design 27](#_Toc44333962)

[PERFORMANCE REPORT 29](#_Toc44333963)

[Abstract 30](#_Toc44333964)

[Definition: 30](#_Toc44333965)

[Code Optimization Importance 30](#_Toc44333966)

[PHP Performance Optimization: 31](#_Toc44333967)

[Where to apply 32](#_Toc44333968)

[Phases of Optimization 32](#_Toc44333969)

[Performance Monitoring Tools 32](#_Toc44333970)

[Conclusion 34](#_Toc44333971)

[References 34](#_Toc44333972)

[TEST PLAN 35](#_Toc44333973)

[Introduction 37](#_Toc44333974)

[Test Methodology 38](#_Toc44333975)

[Resource & Environment Needs 40](#_Toc44333976)

[TEST DOCUMENT 41](#_Toc44333977)

[Sprint Three 44](#_Toc44333978)

[Marking Criteria 44](#_Toc44333979)

[SOURCE CONTROL SNAPSHOT 44](#_Toc44333980)

[SPRINT 3 PROJECT MANAGEMENT PLAN 45](#_Toc44333981)

[SPRINT 3 TEST PLAN 46](#_Toc44333982)

[Introduction 48](#_Toc44333983)

[Test Methodology 49](#_Toc44333984)

[Resource & Environment Needs 52](#_Toc44333985)

[Terms/Acronyms 52](#_Toc44333986)

[OPTIMISATION REPORT 53](#_Toc44333987)

[Definition 53](#_Toc44333988)

[Objectives 53](#_Toc44333989)

[Types of Code Optimization 53](#_Toc44333990)

[Code Optimization Methods 53](#_Toc44333991)

[Where to apply 54](#_Toc44333992)

[Phases of Optimization 54](#_Toc44333993)

[Optimisation of code 55](#_Toc44333994)

[Closing DB Connection 55](#_Toc44333995)

[Checking PHP Version 55](#_Toc44333996)

[Static Analysis Tools 55](#_Toc44333997)

[Recommendations 56](#_Toc44333998)

[TEST DOCUMENTATION 57](#_Toc44333999)

[Analytics 59](#_Toc44334000)

[Auto-refresh analytics on database update 60](#_Toc44334001)

[Password Authentication 61](#_Toc44334002)

[Handover 62](#_Toc44334003)

[Software Review Report 62](#_Toc44334004)

[Requirements Checklist 62](#_Toc44334005)

[Code Testing Process 63](#_Toc44334006)

[Quality Assurance 63](#_Toc44334007)

[Quality Control Processes 63](#_Toc44334008)

[Room for Future Modifications 63](#_Toc44334009)

[Project Management Plan 64](#_Toc44334010)

[Test plan 66](#_Toc44334011)

[Introduction 69](#_Toc44334012)

[Test Methodology 70](#_Toc44334013)

[Test Deliverables 72](#_Toc44334014)

[Resource & Environment Needs 73](#_Toc44334015)

[Terms/Acronyms 73](#_Toc44334016)

[WCAG Test Documentation 74](#_Toc44334017)

[References 77](#_Toc44334018)

# Analysis and Setup

Project Team members:

Sean Boaden

Troy Lucero

Gholamreza Aminy

SOURCE CONTROL

[GitHub](https://github.com/sean-b765/RAD-Project) repository

## Task Matrix

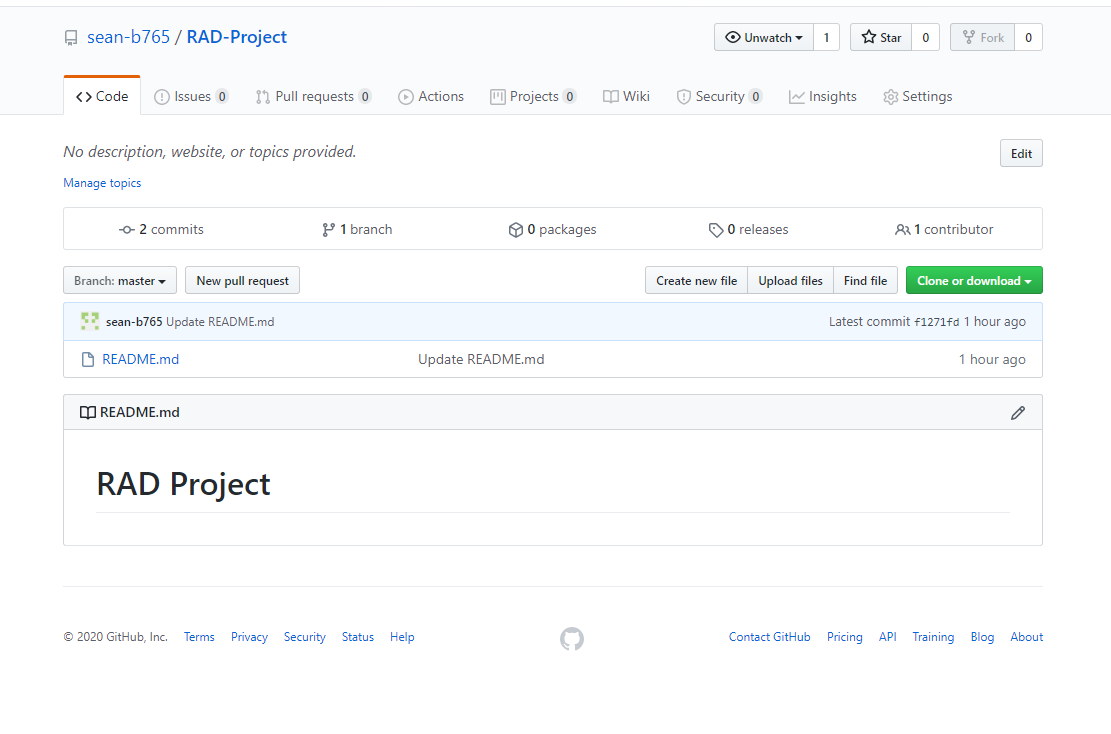
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint One** | **GANTT Chart** | **Test Plan** | **Analysis Report** | **Multi-Platform Report** | **Source Control** | **Demonstration** |
| *Sean* |  |  |  | 29/05 | 2/06 |  |
| Aveneil | Completed 4/06 |  |  |  |  |  |
| Gholamreza |  | 29/05 |  |  |  |  |
| Troy |  |  | 1/06 |  |  |  |
|  |  |  |  |  |  |  |
| **Sprint Two** | **Update GANTT Chart** | **Update Test Plan** | **Software Review Plan** | **Performance Report** | **Source Control** | **Demonstration** |
| Sean |  | x | x |  |  |  |
| Gholamreza | x |  |  | x | x | x |
| Troy |  |  |  |  |  |  |
| **Sprint Three** | **Update GANTT Chart** | **Update Test Plan** | **Optimisation Report** |  | **Source Control** | **Demonstration** |
| Sean | x | x |  |  | x |  |
| Gholamreza |  |  | x |  |  |  |
| **Assessment Point Four** |  |  |  |  |  | **Demonstration** |
| Sean |  |  |  |  |  |  |
| Gholamreza |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

# Sprint One

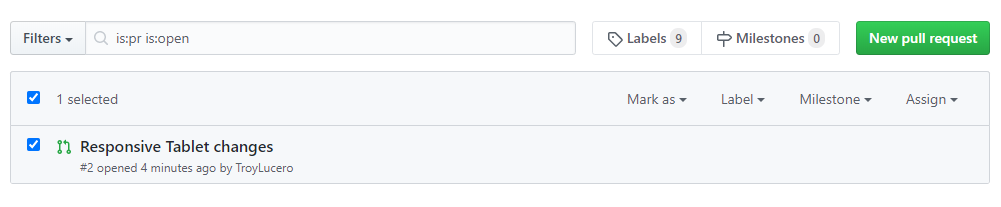
Scrum Master: Sean B

## SOURCE CONTROL

Project from Web Programming will be uploaded to GitHub. At the end of Sprint one, changes will have been pushed to master branch.

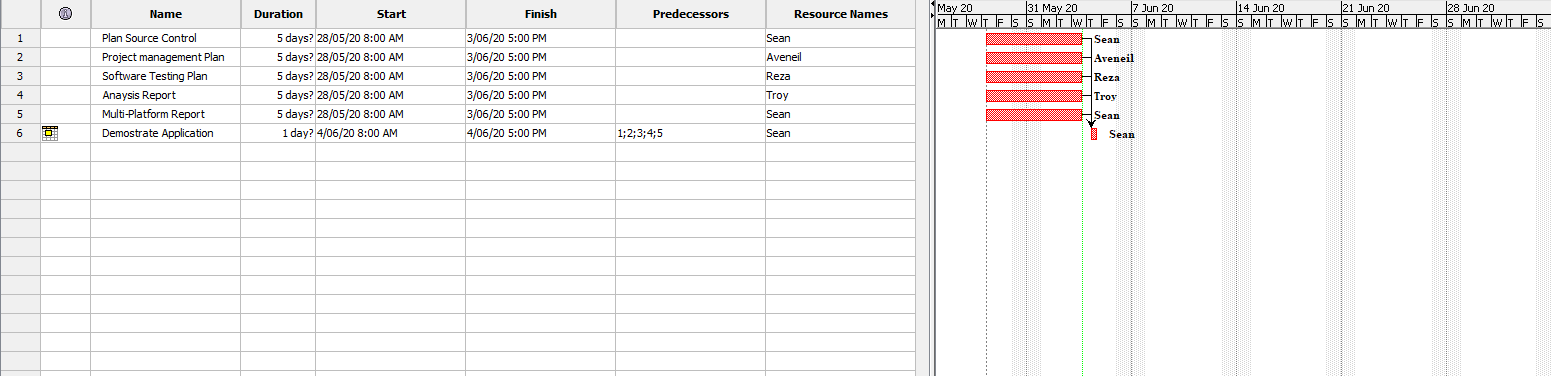


Pull Requests



## PROJECT MANAGEMENT PLAN

By: Aveneil



## TEST PLAN

By: Gholamreza

**Project: Movie Database**

**Client: Acme Entertainment Pty Ltd**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| V0 | 28/5/2020 | Group G | First Version |
|  |  |  |  |
|  |  |  |  |

Test plan 1

1. Introduction 3

Scope 3

In Scope 3

Out of Scope 3

Quality Objective 3

Roles and Responsibilities 3

2. Test Methodology 4

Overview 4

Test Levels 4

Bug Triage 4

Suspension Criteria and Resumption Requirements 5

Test Completeness 5

3. Test Deliverables 5

4. Resource & Environment Needs 6

Testing Tools 6

Test Environment 6

5. Terms/Acronyms 6

### Introduction

#### Scope

##### In Scope

The whole package including GUI, functional and non-functional requirements of the software are in the scope of this document and will be tested in different stages and sprints.

##### Out of Scope

None

#### Quality Objective

The ultimate objective of this plan is to ensure the application under test meets the functional and non-functional requirements as agreed on the contract. The AUT will also be a quality and user friendly product with no bugs.

#### Roles and Responsibilities

To ensure a clear understanding of the roles and responsibilities to achieve the above-mentioned quality objective the following roles and responsibilities are defined for this plan:

* Test Manager: has the overall responsibility of executing this plan. Test manager will schedule and implement the tests and set action plans to rectify the triaged bugs in cooperation with other team members
* QA Analyst: QA Analyst has the responsibility to monitor the proper implementation of this plan, perform regular audits and report to the Test Manager for corrections and Project Manager for further decisions.
* Configuration Manager
* Developers: are responsible to understand and analyze the assigned tasks and do the task on the assigned time ready for test. They will fix defined bugs during the tests reported by the Test Manager.
* Installation Team: They are responsible to ensure the software can be installed with minimum efforts and report any bugs to the Test Manager

### Test Methodology

#### Overview

Choosing the RAD software development requires a suitable test method like Agile in which all the functional and non-functional tests can be completed in each sprint. So the Agile method will be used for this testing plan as in Agile:

* software is developed in incremental, rapid cycles
* Interactions amongst customers, developers and client are emphasized
* focuses on responding to change rather than extensive planning
* every release of the project is tested thoroughly
* any bugs in the system are fixed before the next release

#### Test Levels

Test Levels define the Types of Testing to be executed on the Application Under Test (AUT).

The AUT will be tested in the below levels

* System Testing.
* Acceptance Testing.

The following tests will be performed to ensure clients requirements:

* Functional and Regression Testing;
* GUI and Usability Testing;
* Accessibility Testing;
* Performance Testing;
* System / Integration Testing;
* User Acceptance Testing (UAT).

#### Bug Triage

To ensure fixing the bugs in a timely manner it is absolutely important to prioritize them so that the urgent ones, which are mainly functional reuquirements, get priority in scheduling the tasks to fix the bugs.

The triage would be based on the following requirements:

* GUI and Usability
* Functional
* Accessibility
* Performance

#### Suspension Criteria and Resumption Requirements

Due to size of the project, there is no suspension and resumption criteria.

#### Test Completeness

Test process will be consicered complete if the following is met:

* 100% test coverage
* All Manual & Automated Test cases executed
* All open bugs are fixed or will be fixed in next release

#### Test Deliverables

During different phases of the testing lifecycle the following deliverables should be delivered to ensure the testing process is completed and validated:

* Test Plan
* Test Cases
* Bug Reports
* Test Strategy
* Test Metrics
* Customer Sign Off

### Resource & Environment Needs

#### Testing Tools

To run the testing plan the following tools are required and will be used:

* Test Management Tool
* Configuration management tool
* Static Analysis Tools

#### Test Environment

To test the application a test environment including hardware and software environement is equired in addition to the client specific ones.

Required hardware:

* Computer desktop
* Modem
* Flash memory

Required software:

1. Windows 10
2. Office 360
3. MS Exchange

### Terms/Acronyms

Make a mention of any terms or acronyms used in the project

| TERM/ACRONYM | DEFINITION |
| --- | --- |
| API | Application Program Interface |
| AUT | Application Under Test |

## ANALYSIS REPORT

By Troy

### CITE business rules for software development

CITE Managed Services Business Analysts define easier ways to attain what you need, while Project Managers identify the most suitable development methodology and cooperation model, as well as assemble the most effective project team, and the tech experts work on technical feasibility and select the most efficient technologies.

We are strongly committed to securing business processes from end to end and respecting our customers’ intellectual property rights and data. With this in mind, we have established a set of corporate policies and procedures that every employee must comply with such as account, data, and physical security, along with more specialized policies covering internal applications and systems that employees are required to follow.

#### Clients

Our clients’ interests always come first. If we deliver exceptional product and service to our clients, our own success will follow.

#### Accountability

We are performance oriented and unafraid to make decisions and be held accountable for those decisions.

#### People and reputation

These are our greatest assets. Without compromise, we will operate in an ethical manner and in compliance with regulations, wherever we work, and whoever we work with.

#### Professionalism and quality

We take great pride in our work and are driven to achieve excellence in every project we undertake. We aim to deliver the best products and service in the market.

#### Innovation, creativity

We never discount the past, but we will constantly strive to find a better solution to a client’s problems. Our clients’ best interests are our best interests.

#### Teamwork

Individual creativity is always encouraged, but, more often than not, team effort produces the best results. There is no room for those who put their personal interests ahead of the interests of the firm and its clients.

#### Size

We are a small firm. We want to be big enough to undertake the largest project that any of our clients could conceive, yet small enough to maintain the loyalty and camaraderie that contributes to our success.

#### Anticipation

We constantly strive to anticipate changes in markets and technologies and we will deliver the latest services, tools and technology stacks to our clients.

#### Growth

We operate in a highly competitive environment and we will grow our business aggressively. However, we will always be fair competitors and will never denigrate other firms.

Reference: <http://www.citems.com.au/?page_id=74>

### CITE Managed Services Quality Assurance

CITE Managed Services performs quality assurance throughout the entire software development lifecycle with QA team members being involved at all stages.

A Lead QA specialist is assigned at the commencement of each project and is involved into initial business analysis and requirements specification. Such a simultaneous interaction of our development and QA teams provides for a better understanding of the project scope and the client’s business objectives.

Comprehensive Approach to Quality:

Quality Planning

CITE Managed Services puts together quality plans that govern the applicable set of standards, regulations, procedures, guidelines and tools during the development lifecycle in each project.

Quality Assurance

We have established processes that evaluate project performance and aim to assure that quality standards are being followed and that the deliverables comply with customer requirements.

Quality Control

We measure performance trends to identify defective pieces of code, verify that deliverables are of high quality and that they are complete and correct.

### Acme Entertainment Pty Ltd development requirements

Acme Entertainment have commissioned a prototype movie database, however they want to review and update this application so it can be used across all the major digital platforms. They require a Multi-Platform Report on the merits of the two design options currently used; adaptive and responsive.

Each requirement must be verified and validated to ensure that these are the correct requirements. If the requirements have been accepted and a baseline is established by the stakeholders. Any changes to the requirements are controlled using a Configuration Management process.

The development or migration of the movie database can be hosted on the cloud or suitable local server. Conduct and record suitable testing of the completed development, include this information in the Testing Plan.

Functional Requirements

• Website needs to be adaptive or responsive.

• Accessible from multiple devices

• Can search movies by individual categories or altogether.

• Readability.

Non-functional Requirements

• No bugs or glitches

• Easily accessible

## MULTI-PLATFORM REPORT

***By: Sean***

When choosing between responsive and adaptive design, you should take into account the target audience of your website/business, and plan according to the most used devices within this demographic.

The design to be implemented in **our project** is a **responsive design**. It may be quicker to implement a responsive design over the adaptive web design as adaptive requires multiple stylesheets. Adaptive will often grant *quicker* site load times (especially on mobile), but there are mixed results as to which design can produce the best load time. If no optimization is made to a responsive website, the adaptive solution often loads faster, despite requiring multiple stylesheets.

Basically, responsive web design can produce quicker load times on desktop, while mobile users can get much quicker load times using the adaptive design. For this reason, target audience is vital.

Improper use of media queries can result in mobile devices loading full desktop styles, rather than only the needed mobile styles. A media query is an expression which will check the screen size, and resolve to true or false depending on if the condition is met. E.g.

@media only screen and (min-device-width : 320px) and (max-device-width : 767px) {

    /\* Styles \*/

}

Will check if the device browser width is between 320 and 767 pixels. If the condition is not met, the inner styles will be ignored.

Studies done by Google show that the average user will expect a website to load in under 3 seconds before they consider leaving.

The responsive design will respond quickly to the size of the screen due to CSS media queries. This means only one CSS stylesheet is required, although it will be a larger file. The adaptive design will adapt to the screen size depending on a set of width constraints. There will be no fluid UI changes, and instead the layout will snap to the given screen size constrains when resizing the browser. This can result in some irregular-sized devices having a crowded or lackluster design in some cases.

**Common size constraints:**

The most important factor in screen size is width.

The typical **Mobile** screen width will be **below 768** pixels, when taking landscape mode into account.

A **Tablet** screen will have a width **between 1024 and 768** pixels.

The **Desktop** screens will have a width **above 1024** pixels.

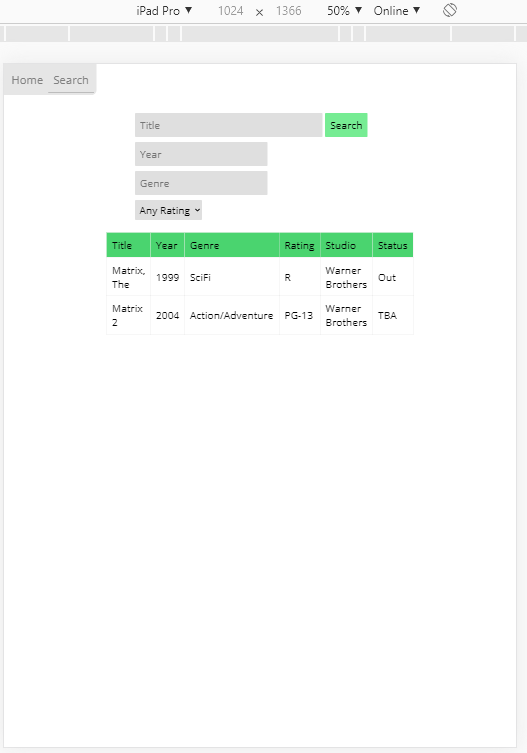
**Responsive VS Adaptive design**

|  |  |
| --- | --- |
| **Responsive** | **Adaptive** |
| Requires one larger spreadsheet. | Labour-intensive, requires multiple stylesheets. |
| Browser loads one large stylesheet. | Browser loads multiple stylesheets. |
| Easier to implement. | Time-consuming. |
| More flexible as browser size changes will produce fluid UI changes. | Changing browser width will not reposition elements immediately. The stylesheet is only adapted at specific width constraints.  This can cause problems on devices which use an irregular screen size. |
| SEO (Search Engine Optimisation) favours Responsive Design, despite load times. | Challenging to SEO – A responsive site will most likely be shown first in Google search results. |

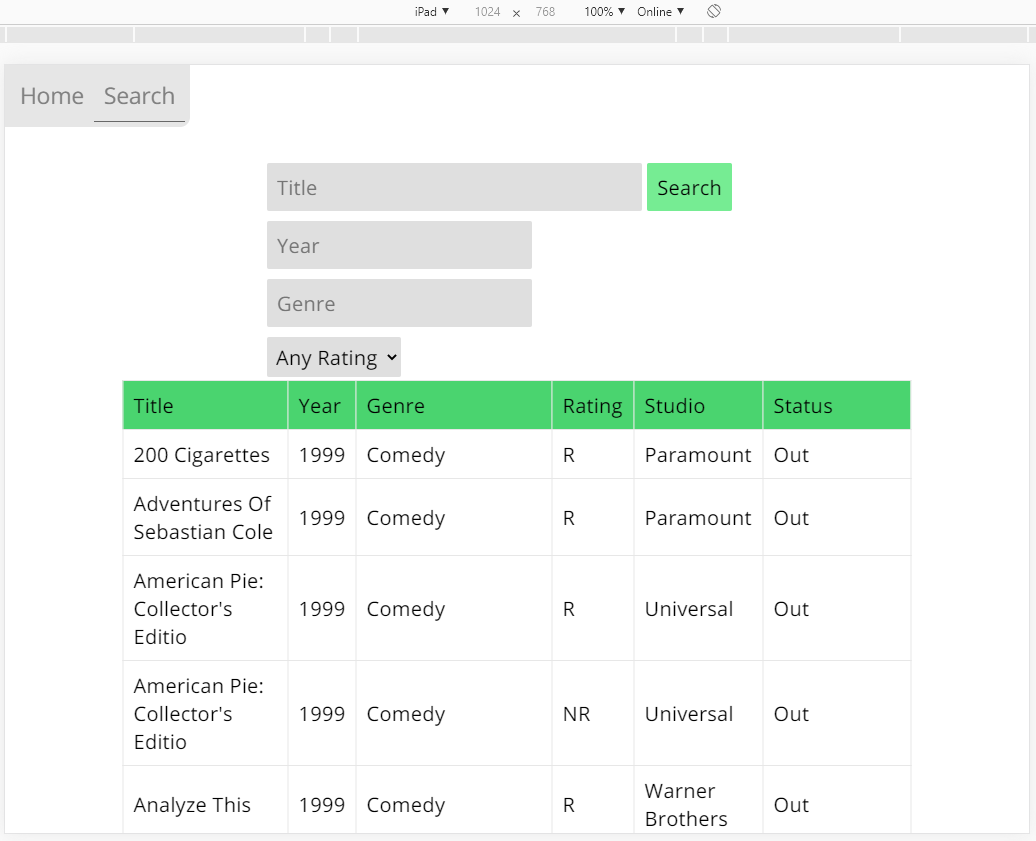
### Multi-Platform Tests

*by Sean*

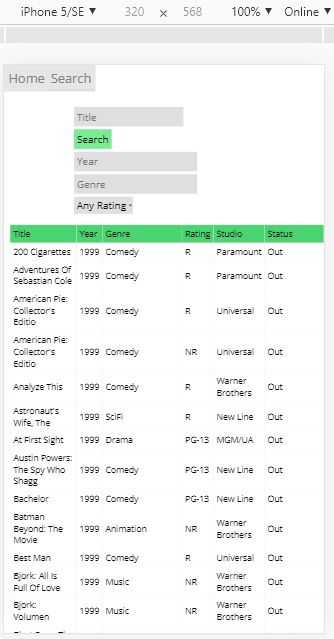
**Tablet** layout (portrait):



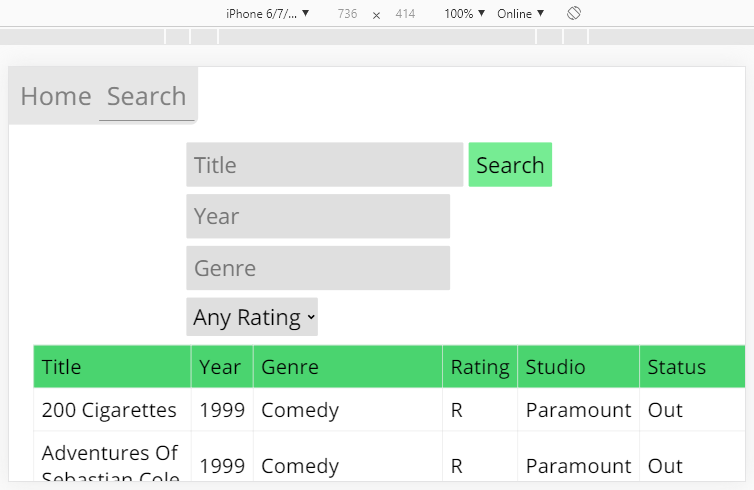
**Tablet** (landscape):



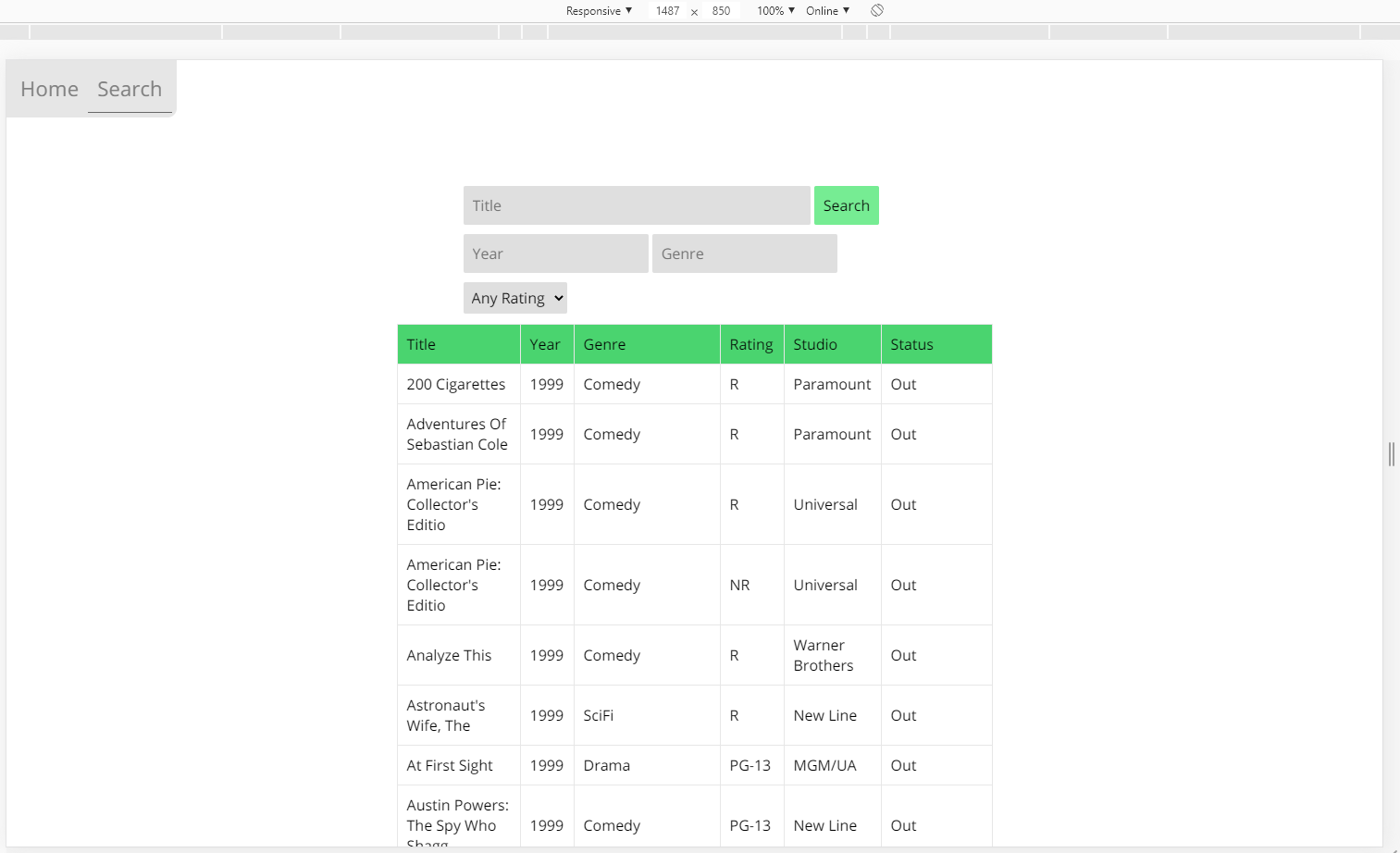
**Mobile** (portrait):



**Mobile** (landscape):



**Desktop**:



**Home Page** in different sizes

|  |  |
| --- | --- |
|  |  |
|  | |

|  |  |
| --- | --- |
| *Testing the bounds of the Search function…*  **No search filters/Title:** | Only 1 filter applied: **Year = 1** |
| **Genre = a** | **Title = ‘ OR 1 == 1**    The user could possibly find a vulnerability here, so I can insert an IF statement into the code to check that *$result is TRUE* (an error has occurred if mysqli\_query() returns false) |
|  | |

SPRINT TWO

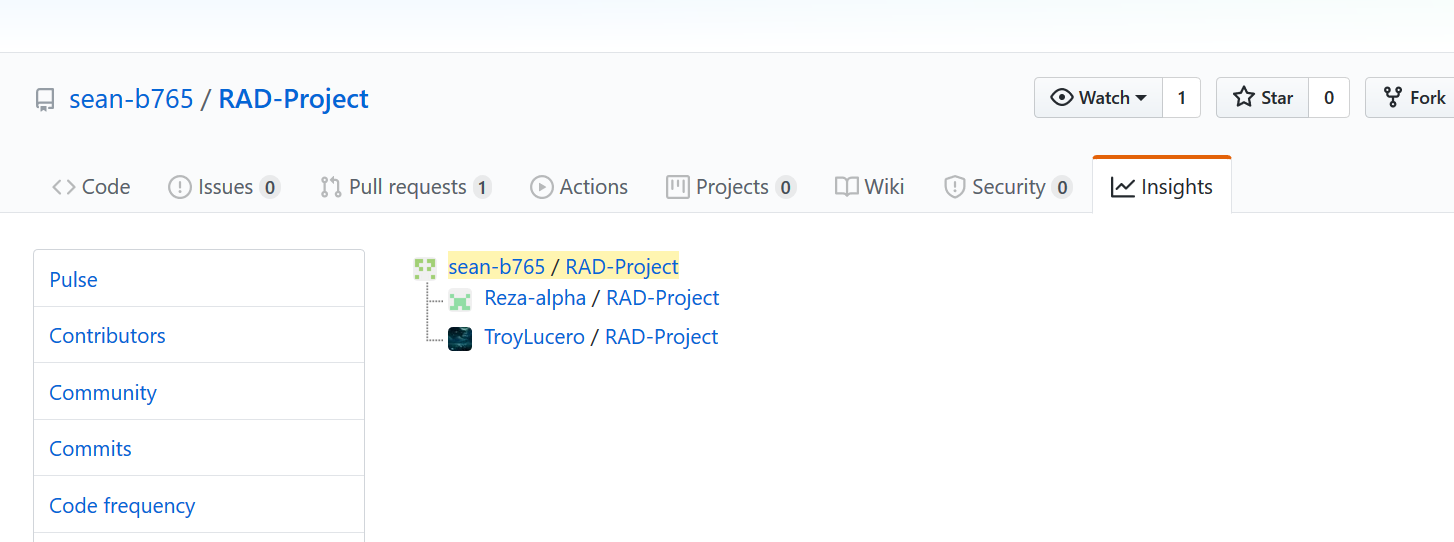
# Sprint Two

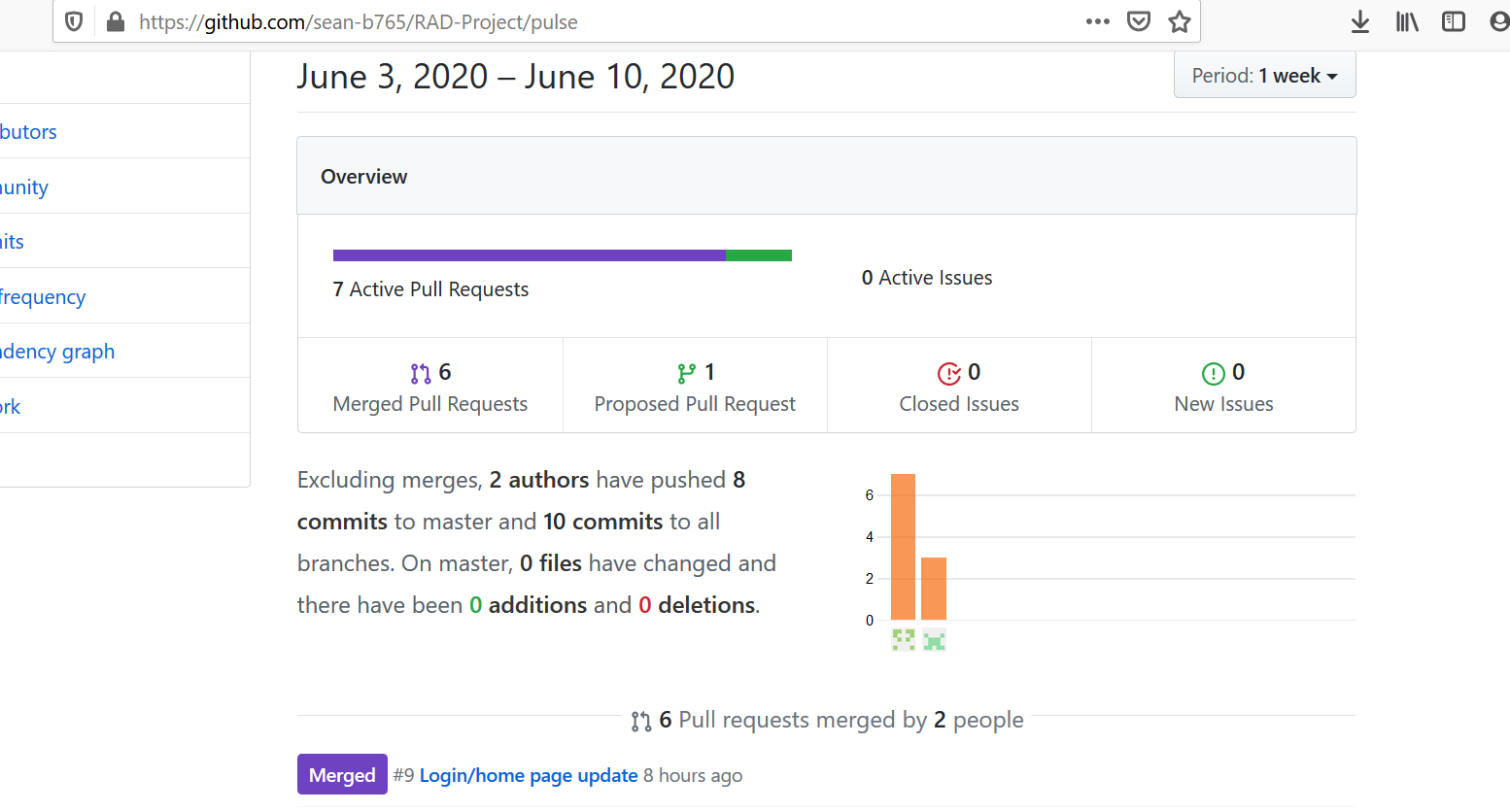
Scrum Master: Gholamreza Aminy

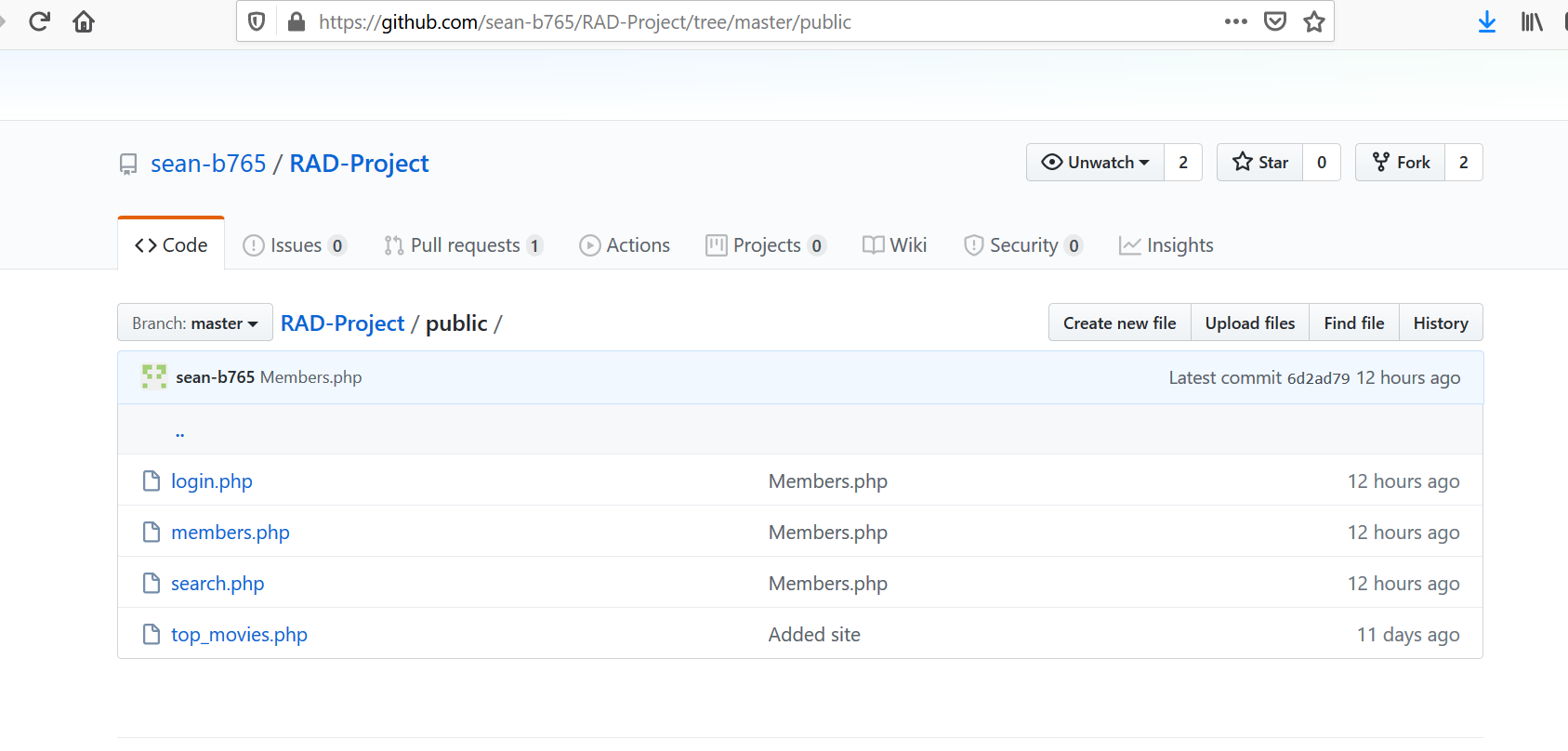
## SOURCE CONTROL SNAPSHOT

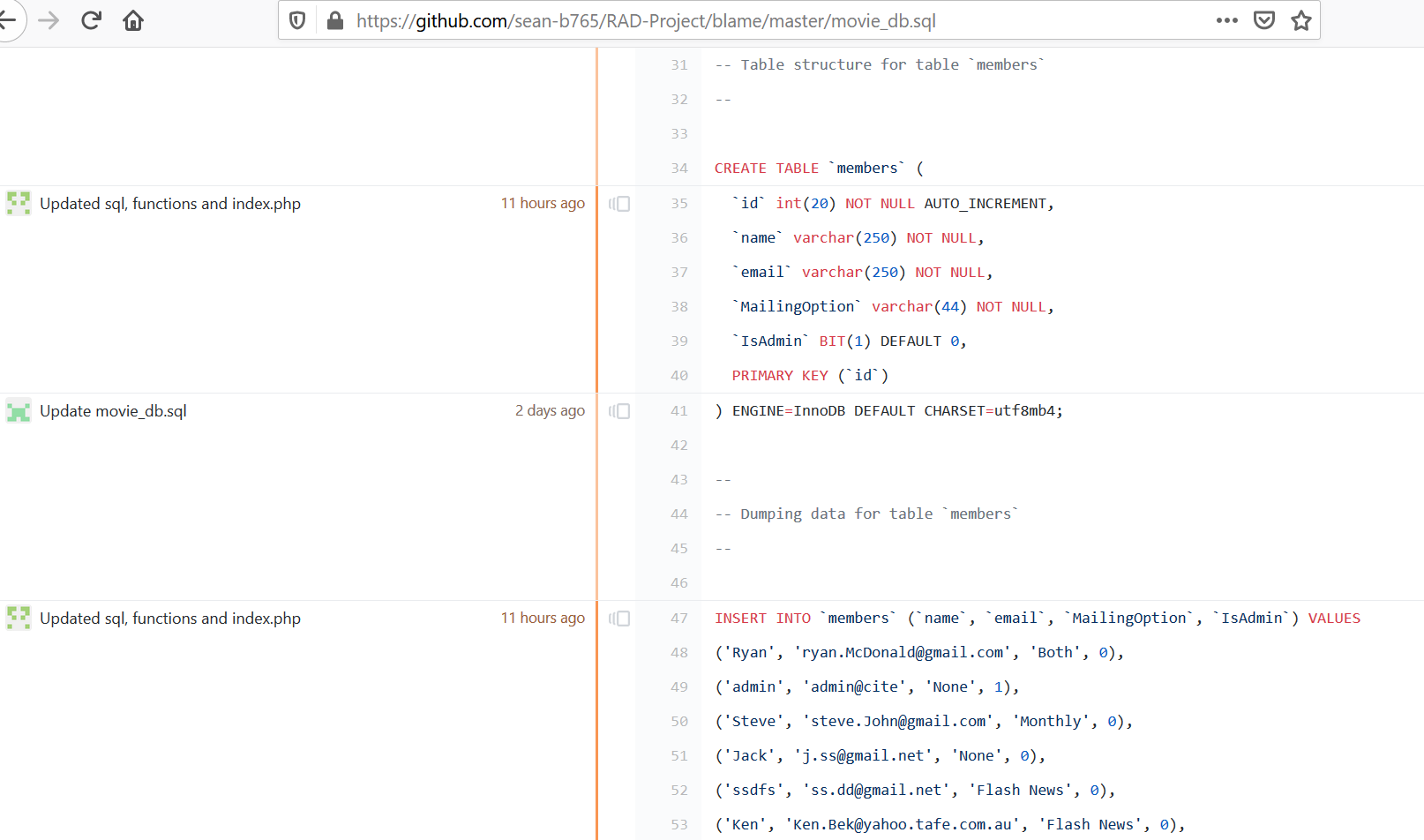
During the project modification as per JMC management requirements the following changes were applied into the source control:

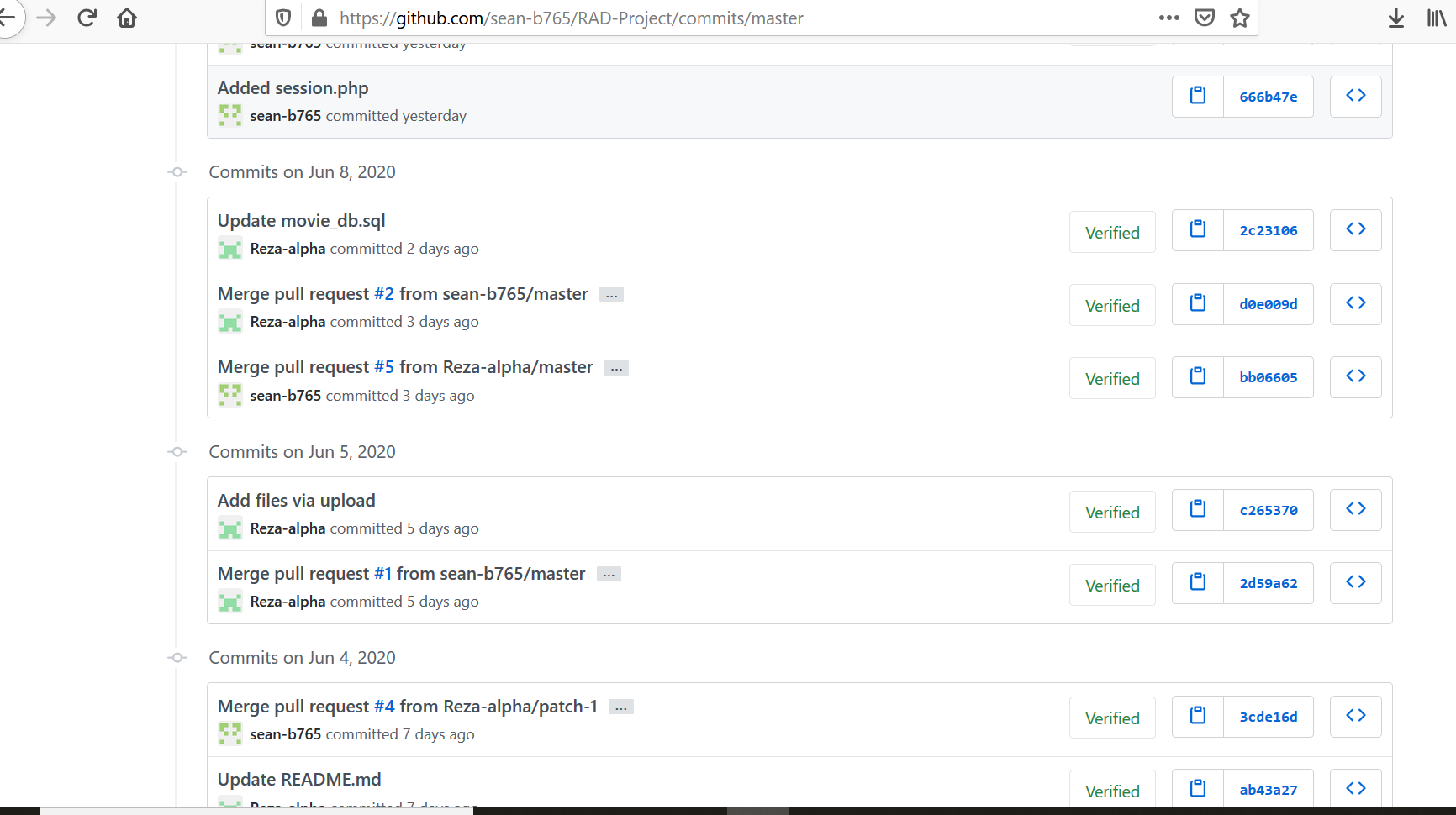
* A new “members” table has been added to the data base to store the members information
* Login fields have been added to the home page to allow users to login to the system
* Signup page has been added to enable new members to register





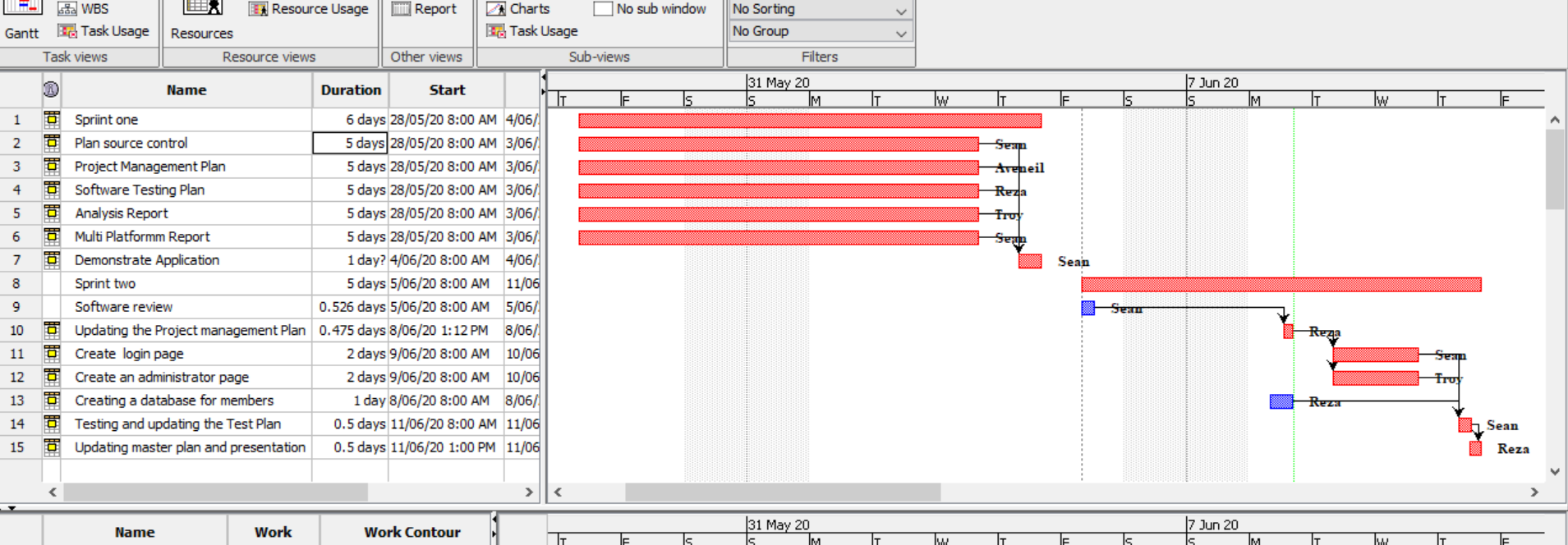






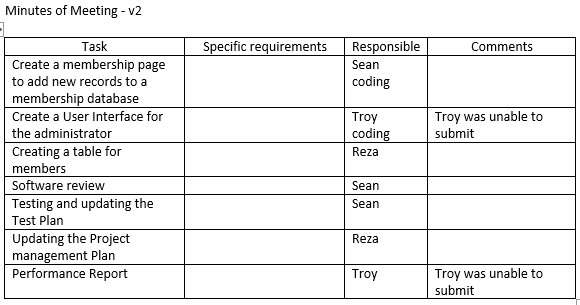
## UPDATED PROJECT MANAGEMENT PLAN

As one of the team members left the company, current team is smaller and tasks are a bit condensed. The team decided to



Troy one of team members was not attending the project to meet milestones. The work load was shared between other team members. The issue was raised to client who advised the team to have a meeting to rectify the issue. To fulfil client requirements, a meeting will be held to clarify the project team work load and applying a catch-up plan.

So the final achievement was as below



## UPDATED SOFTWARE REVIEW PLAN

By Sean

### Purpose

The purpose of this Software Review Plan is to verify the requirements, test plan and outline testing documentation for the project at hand. This report will allow team members to gain a better understanding of this sprint.

### Verification

During development, team members are expected to be familiar with requirements and relevant coding standards, such as naming conventions. Verification can be performed regularly by the Scrum Master, and can be done by meeting with developers for code review.

Verification is required to ensure development is being carried out properly and in an efficient manner, avoiding unnecessary features (gold-plating), and to make sure standards and regulations are being met.

* Functional and Non-Functional requirements from previous Sprint have been fulfilled…

Functional

* Website is a responsive design
* Accessible from desktop, tablet, and mobile devices
* Searching a movie by individual filters, search terms or a mix
* Code has been commented and is readable

Non-Functional

* No bugs, glitches
* Interface is user friendly and accessible
* User Interface test has been documented
* Functionality test has been performed and documented (for search page)

Standards and Business Rules

* CITE Business Rules have been incorporated in this project.
* ISO 12207:2017 Software Standards and coding conventions have been implemented.
* Modularity: classes are separated between files, and project file structure is consistent and logical.
* Naming conventions follow camel case, as should be the standard in web development.
* Sufficient commenting
* Sufficient reports and documentation
* Code formatting and indentation is consistent

### Sprint Requirements

Functional:

* Membership sign up form via Full name and Email address.
  + Option to join the mailing list at sign up
    - Option of monthly newsletter, or
    - Breaking newsflash emails (or both)
* Membership portal page
  + Enter email address to be removed from mailing list which gets sent to administrator
* Admin Portal
  + View all members
  + Enter an email address to be removed from the mailing list
* Reports
  + Testing Report (Software Testing Plan update)
  + Testing Plan (test documentation of User Interface, functionality…)
  + Performance Report
  + Project Management Plan

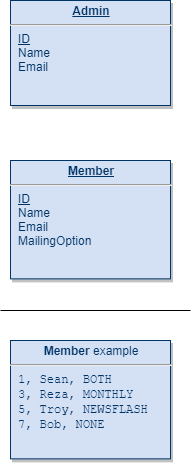
Non-Functional:

* The newly added pages should follow a responsive design
* Validate and sanitize user inputs

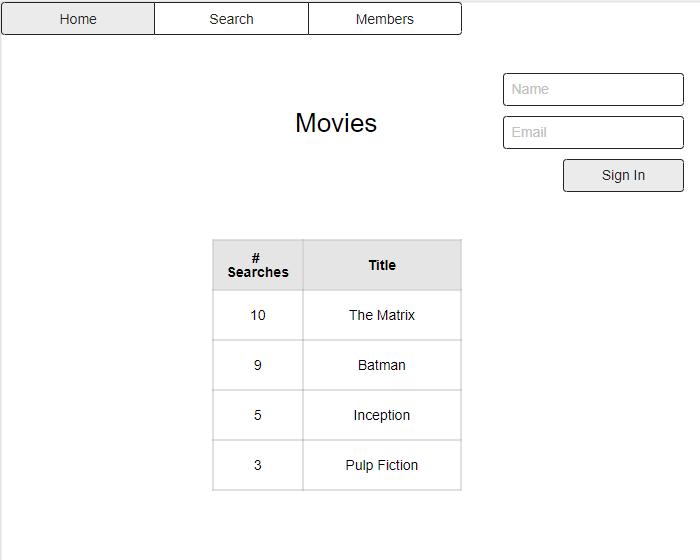
### System Design

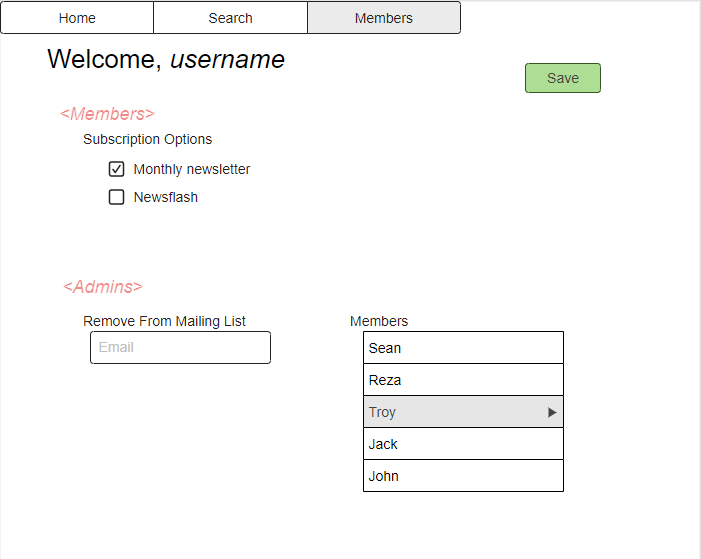
#### Database

The original database already consists of movies, top\_searches tables. The following Member table will need to be created, and default admin records insertted.



#### User Interface





## PERFORMANCE REPORT

By Gholamreza

Contents

[Abstract 2](file:///E:\Documents\xampp\htdocs\RAD-Project\Handover\Performance_Report.docx#_Toc44152103)

[Definition: 2](file:///E:\Documents\xampp\htdocs\RAD-Project\Handover\Performance_Report.docx#_Toc44152104)

[Code Optimization Importance 2](file:///E:\Documents\xampp\htdocs\RAD-Project\Handover\Performance_Report.docx#_Toc44152105)

[PHP Performance Optimization: 2](file:///E:\Documents\xampp\htdocs\RAD-Project\Handover\Performance_Report.docx#_Toc44152106)

[Where to apply 4](file:///E:\Documents\xampp\htdocs\RAD-Project\Handover\Performance_Report.docx#_Toc44152107)

[Phases of Optimization 4](file:///E:\Documents\xampp\htdocs\RAD-Project\Handover\Performance_Report.docx#_Toc44152108)

[Performance Monitoring Tools 4](file:///E:\Documents\xampp\htdocs\RAD-Project\Handover\Performance_Report.docx#_Toc44152109)

[Conclusion 6](file:///E:\Documents\xampp\htdocs\RAD-Project\Handover\Performance_Report.docx#_Toc44152110)

[References 6](file:///E:\Documents\xampp\htdocs\RAD-Project\Handover\Performance_Report.docx#_Toc44152111)

### Abstract

Application proper performance is an important part of application development where will affect the user’s satisfaction which can make a business successful or take it down. Application performance can be improved by implementing code optimisers and performance tools.

### Definition:

To improve the [code quality](http://www.viva64.com/en/t/0077/) and efficiency it is necessary to optimize the code as much as possible to ensure the best speed and experience for users.

Code optimization is a vital concept in application performance and can make a big difference in application performance when compiling.

Code optimization can be defined as any method of code modification to improve [code quality](http://www.viva64.com/en/t/0077/) and efficiency. A program may be optimized so that it becomes a smaller size, consumes less memory, executes more rapidly, or performs fewer input/output operations.

The basic requirements optimization methods should comply with, is that an optimized program must have the same output and side effects as its non-optimized version. This requirement, however, may be ignored in the case that the benefit from optimization, is estimated to be more important than probable consequences of a change in the program behaviour. (Viva64, 2020)

### Code Optimization Importance

There are many reasons for performance importance and the main ones can be:

* User experience

Loading and running time of the application are the important issues which can easily affect the number of users in using your app or your competitors.

* Conversions

Conversions are the number of users who actually buy or download your product or generally the users who give money for your services. A faster website means more conversions and more profit

* Scalability

The more requests your application can handle per second, the more traffic you can handle. If your application can handle a single process in 100ms, it means your application can serve 10 requests per second so if you reduce the application processing time to the half, your application capacity will double. (Aladdin, 2020)

* Memory usage

Reduces the space consumed and increases the speed of compilation.

### PHP Performance Optimization:

There are two types of optimization:

* Machine Independent Optimization which attempts to improve the intermediate code to get a better target code as the output. The part of the intermediate code which is transformed here does not involve any CPU registers or absolute memory locations.
* Machine Dependent Optimization which is done after the target code has been generated and when the code is transformed according to the target machine architecture. It involves CPU registers and may have absolute memory references rather than relative references. Machine-dependent optimizers put efforts to take maximum advantage of the memory hierarchy. (Singh, 2020)

There are some good practices in optimizing php performance including but not limited to:

* Choose the Right Version

Well, this one is obvious especially after PHP7 which have the best performance among the old PHP versions. I will not make a comparison here between PHP versions since there are so many articles discussed that, but you still can take a look at the following chart showing the performance of PHP versions run by different CMSs.

* PHP Micro-Optimization

Micro-Optimization is the minor changes in your code that improve your application performance. For example, if you are going to use a for-loop, it is always better to calculate the length in advance. The next image shows the result of for-loop with 1000 keys with 1-byte values are given. You can notice the improvement in the execution time.

* XHProf**:** Profiling PHP Code

After we have seen the limited value of PHP Micro-Optimization, it is time to determine precisely which part of our code is slow without the need to guess using a profiling tool called XHProf. After installing and configuring XHProf on your server, XHProf will append header and footer to all your PHP scripts and generate a report where you can find all the executed functions, the execution time, and the number of calls of each function. (Aladdin, 2020)

Code Optimization is done in the following different ways:

* Compile Time Evaluation
* Variable Propagation
* Dead code elimination:

Variable propagation often leads to making assignment statement into dead code

* Code Motion:  
  • Reduce the evaluation frequency of expression.  
  • Bring loop invariant statements out of the loop.
* Induction Variable and Strength Reduction:  
  • An induction variable is used in loop for the following kind of assignment i = i + constant.  
  • Strength reduction means replacing the high strength operator by the low strength.

### Where to apply

Optimization can be applied in the following stages of coding:

* Source program  
  Optimizing the source program involves making changes to the algorithm or changing the loop structures. User is the actor here.
* IntermediateCodeOptimizing the intermediate code involves changing the address calculations and transforming the procedure calls involved. Here compiler is the actor.
* TargetCodeOptimizing the target code is done by the compiler. Usage of registers, select and move instructions is part of optimization involved in the target code. (Singh, 2020)

### Phases of Optimization

There are generally two phases of optimization:

* GlobalOptimization**:**Transformations are applied to large program segments that includes functions, procedures and loops.
* LocalOptimization**:**  
  Transformations are applied to small blocks of statements. The local optimization is done prior to global optimization. (wikipedia, 2020)

### Performance Monitoring Tools

Here are some performance monitoring tools you may want to consider in building your PHP application.

* PHP code profilers:

There are two types of code profilers: standard and tracing profilers which are widely used by the web developers to improve their performance or to determine the weak points.

Standard profilers periodically record stack traces of your application, while tracing profilers are lighter-weight than standard profiles and can be used while you code.

Standard code profilers give you a snapshot of important metrics like CPU, memory usage, time spent per line of code, and frequency of method calls.

Tracing profilers have quite an advantage from standard as you can use it every day, and it won’t slow you down. It is designed to catch errors in real time while you’re developing, instead of catching errors after the fact. Tracing profilers are considered as your first line of defence in dealing with bugs.

* PHP application performance management (APM) tools

Tracking web application behaviours in certain scenarios is so important so that developers can optimize parts of the application that might be underperforming or which part of a web application slows down the entire application. In most cases, performance bottlenecks happen during database queries and API calls because the application has to wait for these processes to complete before moving on to the next task. Retrace is one of the tools that can be used in performance monitoring. Here are some screen captures on how Retrace works.

Based on the data gathered by Retrace, this web application has multiple inserts and select queries that put a lot of load on the server. By analysing these insert and select queries, the developers were able to pinpoint what’s causing the multiple inserts and select database queries. A possible solution like applying left joins in select queries also improved the performance of the web application.

* Real user monitoring (RUM) tools

Real user monitoring (RUM) such as [Google Analytics](http://www.google.com/analytics/), relies heavily on services that constantly measure the application in the background. This is done by a small JavaScript in a webpage that will collect some data as the user browses the web application. One of the main uses of RUM is to record which pages are being viewed by the users and how long they stay on each one. This kind of data is useful to decide where to put ads or where to put information for maximum viewability. The data gathered from RUM can also be used to improve the user experience of a website or application.

* Web server access logs

For Retrace to be able to monitor a web application, the time, request path, status, and log event date must be present in the web server’s access logs. These parameters, if not present in the logs, can be enabled in the configuration of the web server. Using the data collected from the access logs, Retrace is able to formulate the requests per minute, average load time, and HTTP error request data of the web application.

* Exception tracking

PHP, like other programming languages, handles exceptions pretty well. Exceptions are very important to track because they usually stop the execution of a system. But if the system is throwing exceptions in the back-end, e.g. web services, then it’s hard to keep track of these exceptions. (MENDEZ, 2018)

### Conclusion

In conclusion, application performance is an important aspect of coding and need to be optimized to ensure customer satisfaction and a wider market. Users need a faster and more reliable site which can provide them with the best services. Meanwhile we should consider optimizing the code using specialized utilities whenever possible.

### References

Aladdin, M. (2020). *PHP Performance Optimization*. Retrieved from https://codeburst.io: https://codeburst.io/php-performance-optimization-992acaa78817

MENDEZ, J. (2018). *PHP Performance: Five Types of Tools You Need To Know*. Retrieved from https://stackify.com: https://stackify.com/php-performance-tool-types/#:~:text=%20PHP%20Performance%3A%20Five%20Types%20of%20Tools%20You,web%20application%2C%20the%20time%2C%20request%20path%2C...%20More%20

Singh, P. (2020). *Code Optimization in Compiler Design*. Retrieved from https://www.geeksforgeeks.org: https://www.geeksforgeeks.org/code-optimization-in-compiler-design/

Viva64. (2020). *Code Optimization*. Retrieved from https://www.viva64.com: https://www.viva64.com/en/t/0084/

wikipedia. (2020). *Optimizing\_compiler* . Retrieved from https://en.wikipedia.org: https://en.wikipedia.org/wiki/Optimizing\_compiler

## TEST PLAN

**Project: Movie Database**

**Client: Acme Entertainment Pty Ltd**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| V0 | 28/5/2020 | *Gholamreza Aminy* | First Version |
| V1 | 7/06/2020 | *Sean Boaden* | updated to meet requirements of Sprint Two |
|  |  |  |  |

Test plan 1

1. Introduction 3

Scope 3

In Scope 3

Out of Scope 3

Quality Objective 3

Roles and Responsibilities 3

2. Test Methodology 4

Overview 4

Test Levels 4

Bug Triage 4

Suspension Criteria and Resumption Requirements 5

Test Completeness 5

3. Test Deliverables 5

4. Resource & Environment Needs 6

Testing Tools 6

Test Environment 6

5. Terms/Acronyms 6

### Introduction

#### Scope

#### In Scope

The whole package including GUI, functional and non-functional requirements of the software are in the scope of this document and will be tested in different stages and sprints.

#### Out of Scope

None

#### Quality Objective

The ultimate objective of this plan is to ensure the application under test meets the functional and non-functional requirements as agreed on the contract. The Application Under Test will also be a quality and user friendly product with no defects.

#### Roles and Responsibilities

To ensure a clear understanding of the roles and responsibilities to achieve the above-mentioned quality objective the following roles and responsibilities are defined for this plan:

* Test Manager: has the overall responsibility of executing this plan. Test manager will schedule and implement the tests and set action plans to rectify the triaged bugs in cooperation with other team members
* QA Analyst: QA Analyst has the responsibility to monitor the proper implementation of this plan, perform regular audits and report to the Test Manager for corrections and Project Manager for further decisions.
* Configuration Manager
* Developers: are responsible to understand and analyze the assigned tasks and do the task on the assigned time ready for test. They will fix defined bugs during the tests reported by the Test Manager.
* Installation Team: They are responsible to ensure the software can be installed with minimum efforts and report any bugs to the Test Manager

### Test Methodology

#### Overview

Choosing the RAD software development requires a suitable test method like Agile in which all the functional and non-functional tests can be completed in each sprint. So the Agile method will be used for this testing plan as in Agile:

* software is developed in incremental, rapid cycles
* Interactions amongst customers, developers and client are emphasized
* focuses on responding to change rather than extensive planning
* every release of the project is tested thoroughly
* any bugs in the system are fixed before the next release

#### Test Levels

Test Levels define the Types of Testing to be executed on the Application Under Test (AUT).

The AUT will be tested in the below levels

* White Box Tests
  + Branch Coverage tests – can be performed on certain conditional operations. It is important that conditional bounds are tested on certain points, to ensure the application works well.
* Black Box Tests
  + **Functional tests** – where the new membership portal inputs will be tested. Sign up / sign in functions, administrative functions should be tested and documented.
  + **User interface** tests – will be used to test that the new members page follows the responsive design.

The following tests will be performed to ensure clients requirements:

* Functional and Regression Testing;
* GUI and Usability Testing;
* Accessibility Testing;
* Performance Testing;
* System / Integration Testing;
* User Acceptance Testing (UAT).

#### Bug Triage

To ensure fixing the bugs in a timely manner it is absolutely important to prioritize them so that the urgent ones, which are mainly functional reuquirements, get priority in scheduling the tasks to fix the bugs.

The triage would be based on the following requirements:

* GUI and Usability
* Functional
* Accessibility
* Performance

#### Suspension Criteria and Resumption Requirements

Due to size of the project, there is no suspension and resumption criteria.

#### Test Completeness

Test process will be considered complete if the following is met:

* 100% test coverage.
* All Manual & Automated Test cases executed
* All open bugs are fixed or will be fixed in next release

#### Test Deliverables

During different phases of the testing lifecycle the following deliverables should be delivered to ensure the testing process is completed and validated:

* Test Documentation Report
  + Test Plan
  + Test Cases
  + Bug Reports
  + Test Strategy
  + Test Metrics
* Customer Sign Off

### Resource & Environment Needs

#### Testing Tools

To run the testing plan the following tools are required and will be used:

* Test Management Tool
* Configuration management tool
* Static Analysis Tools

#### Test Environment

To test the application a test environment including hardware and software environement is equired in addition to the client specific ones.

Required hardware:

* Computer desktop
* Modem
* Flash memory

Required software:

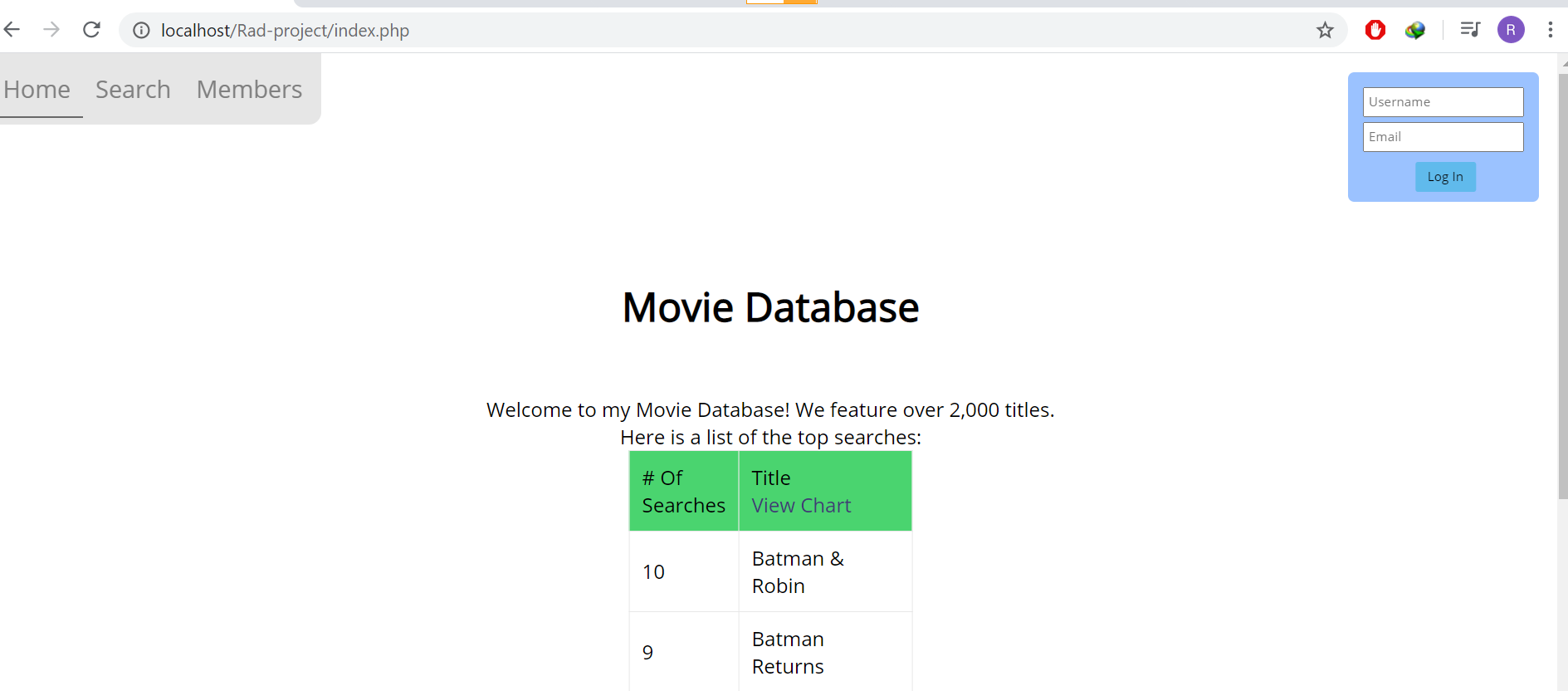
1. Windows 10
2. Office 360
3. MS Exchange
4. XAMPP or similar web server software packages including MySQL and Apache

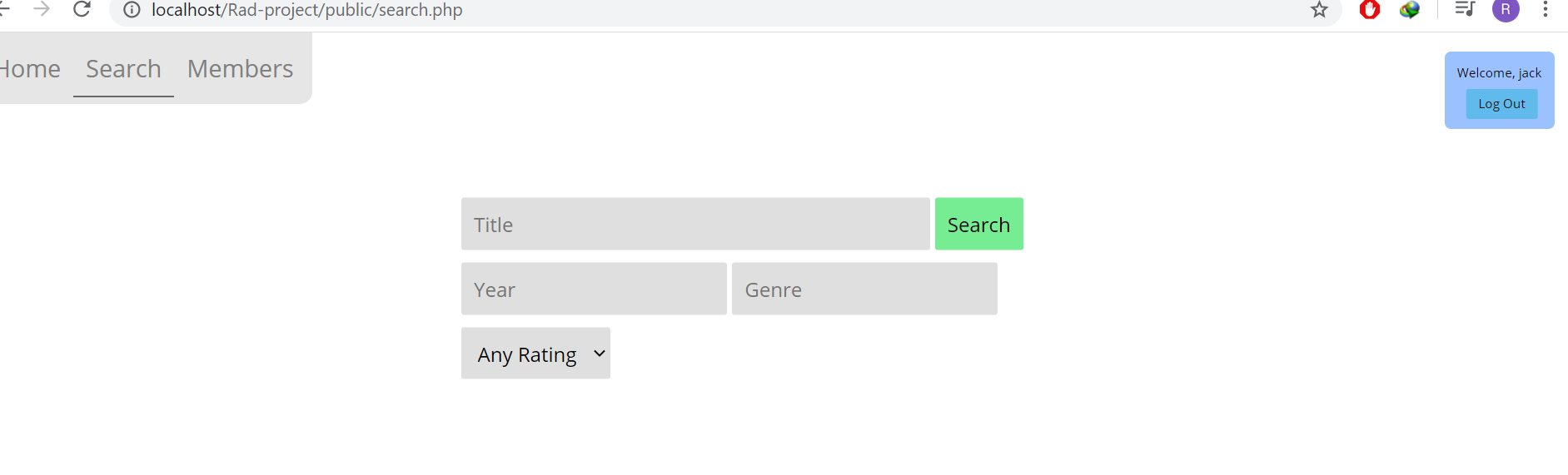
#### Terms/Acronyms

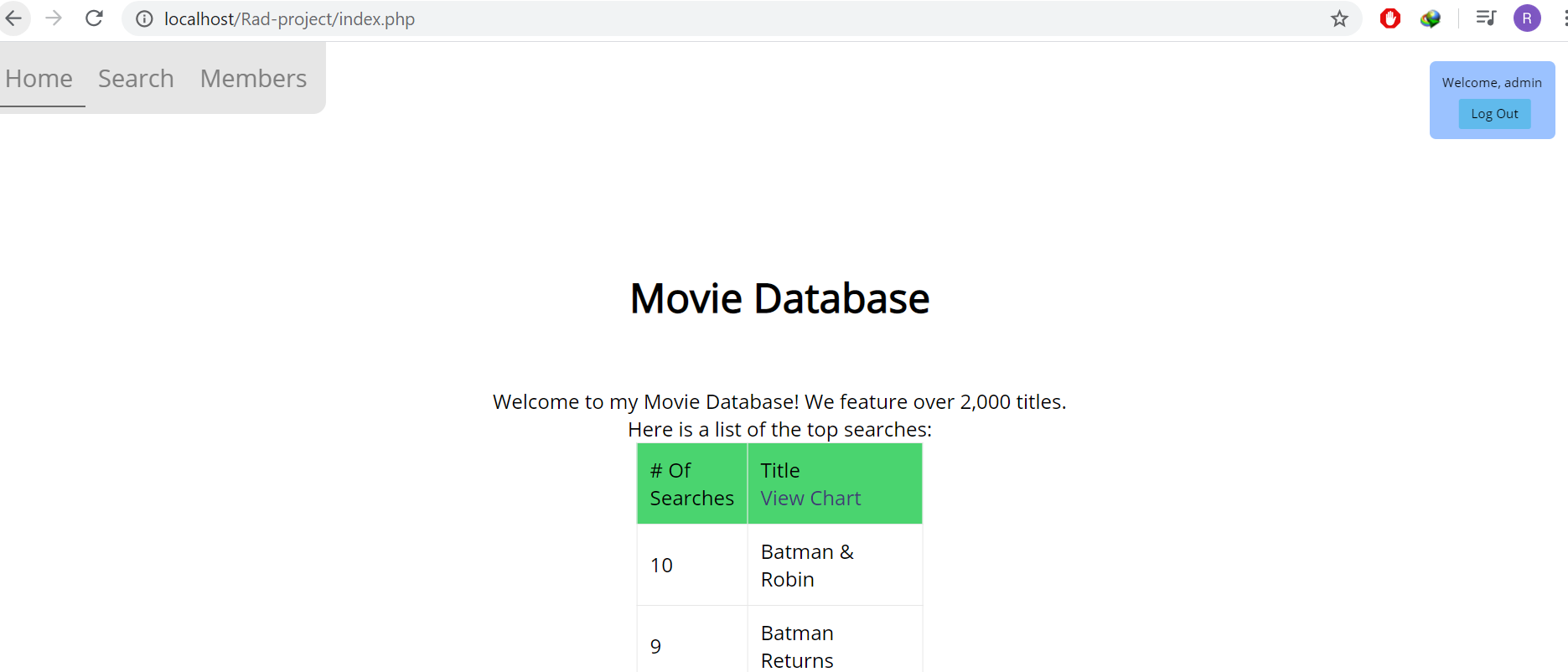
Make a mention of any terms or acronyms used in the project

| TERM/ACRONYM | DEFINITION |
| --- | --- |
| API | Application Program Interface |
| AUT | Application Under Test |

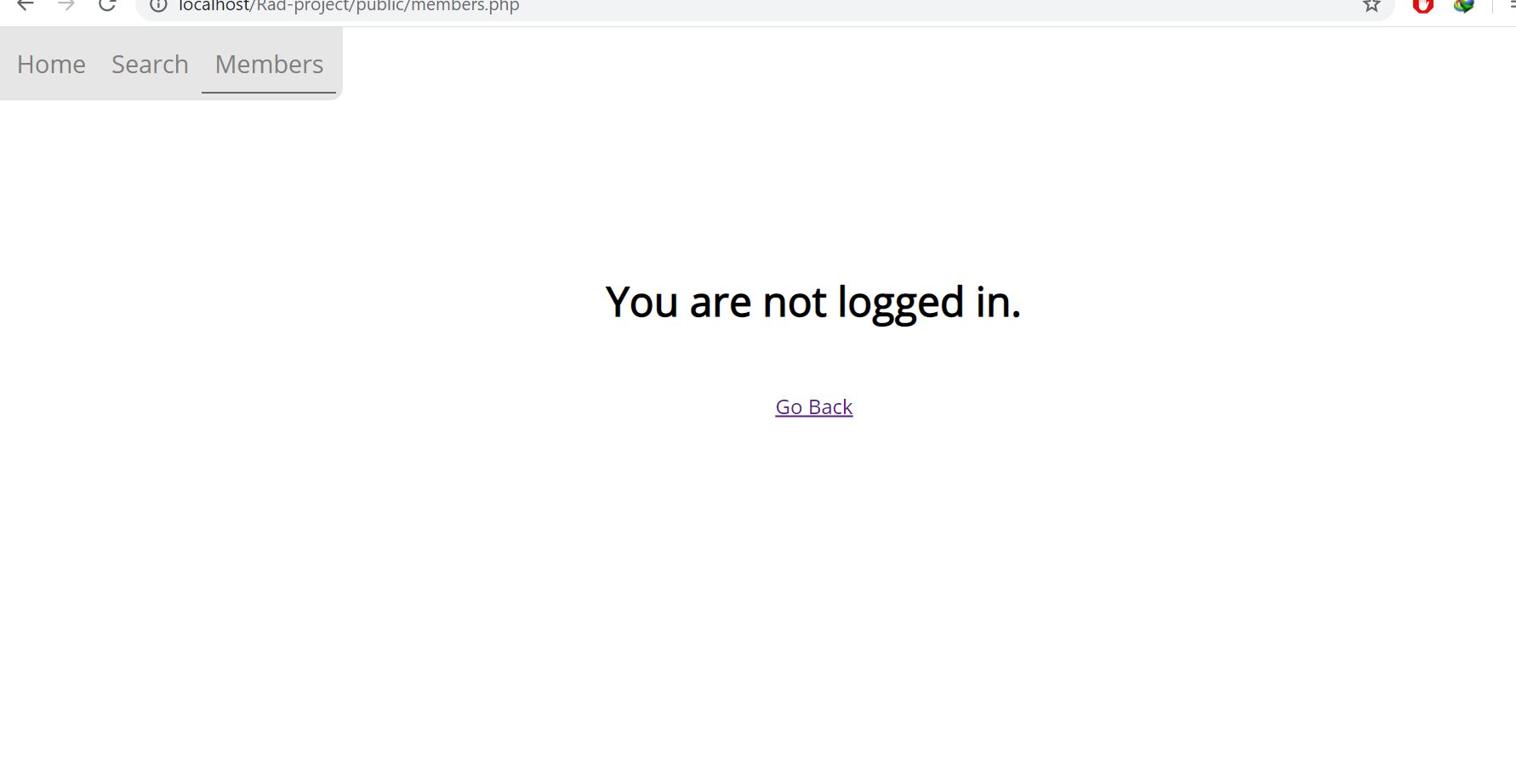
## TEST DOCUMENT

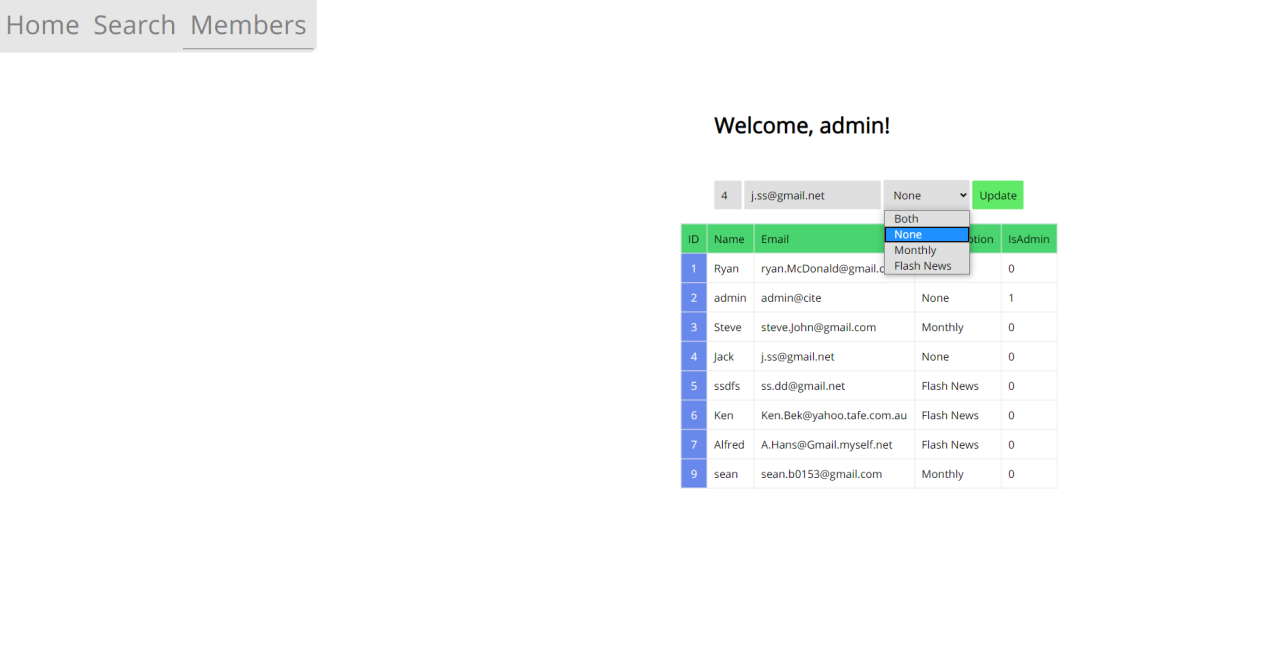


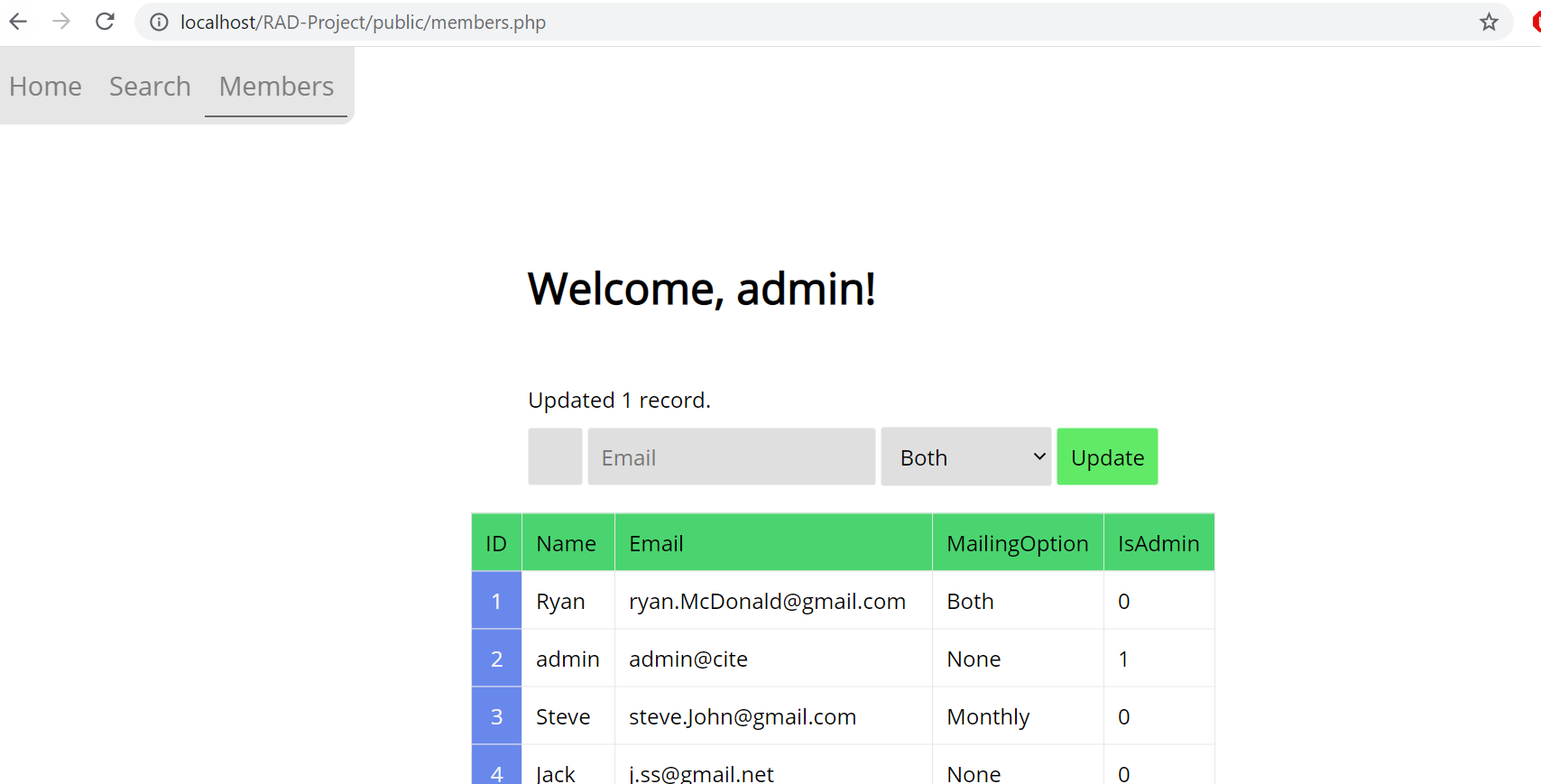


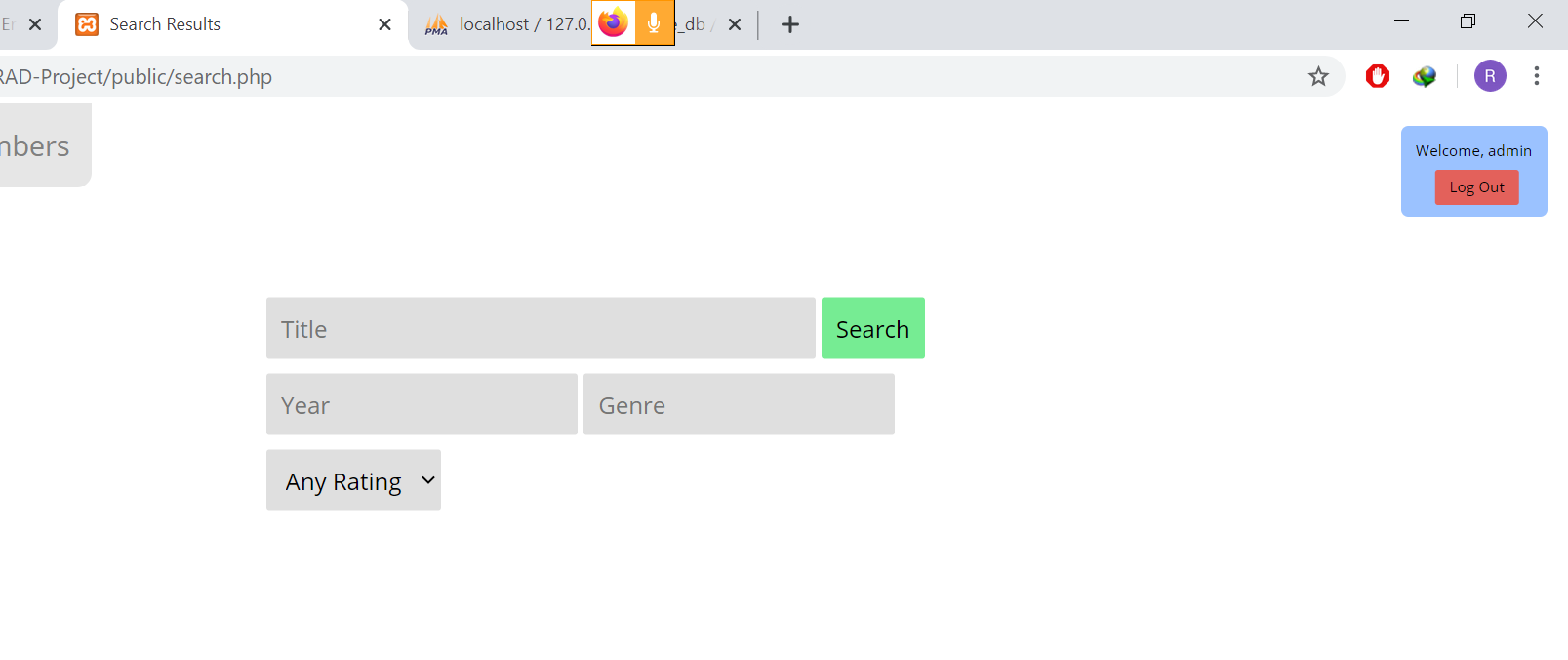












# Sprint Three

Scrum Master: Sean Boaden

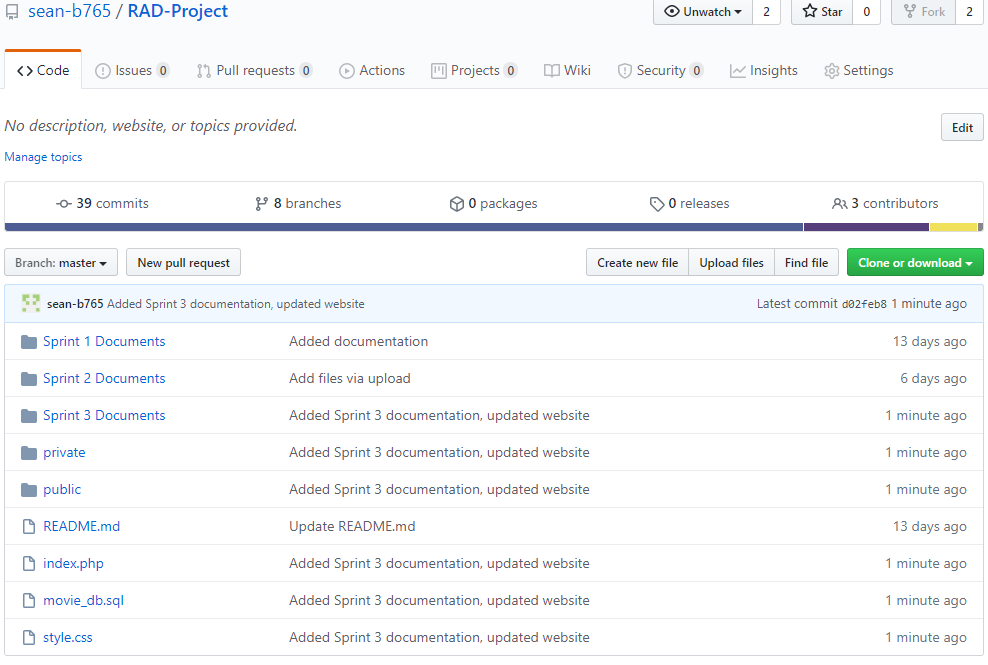
## Marking Criteria

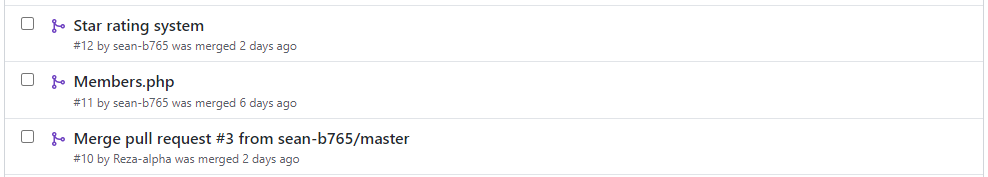
|  |
| --- |
| The team has updated the source control system with the sprint three code and associated documents. |
| The team has a [Project Management Plan](#_SPRINT_3_PROJECT) completed up to the sprint three assessment milestone. |
| Update [Software Testing Plan](#_SPRINT_3_TEST) document that reflects the spring three development phase |
| The Movie Database Application functions correctly using administrator password. |
| The Movie Database Application display top 10 information. |
| The presenter can explain [how the administrator sign-in works](#_Password_Authentication), with different users and groups. |
| The [analytics](#_Analytics) web page display streamed data |
| Documentation for [Optimisation Report](#_OPTIMISATION_REPORT) |

## SOURCE CONTROL SNAPSHOT

The following changes were applied into the source control:

* New DB tables include Groups, Group\_Members, and Member\_Ratings and Streaming
* New columns include AvgRating, TotalIntegerRating, and NumberOfRatings in table Movies. Members table now has Password column.
* If an Admin or ACME user has no password, a ‘Change Password’ message appears to get them to set their password. The Password must be 8 characters, alphanumeric and the Password will only be set if it reaches this criteria.

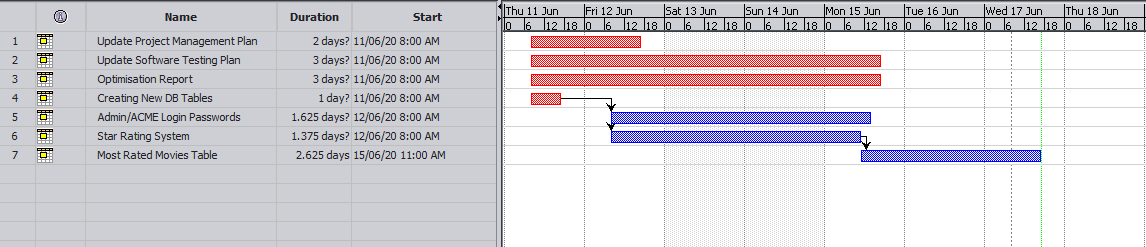




Some Pull Requests

## SPRINT 3 PROJECT MANAGEMENT PLAN

Updated by Sean *for Sprint 3*



## SPRINT 3 TEST PLAN

Updated by Sean *for Sprint 3*

**Project: Movie Database**

**Client: Acme Entertainment Pty Ltd**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| V0 | 28/5/2020 | *Gholamreza Aminy* | First Version |
| V1 | 7/06/2020 | *Sean Boaden* | Second Version – updated to meet requirements of Sprint Two |
| V2 | 10/06/2020 | Sean Boaden | Updated to meet requirements of Sprint Three |

Test plan 1

1. Introduction 3

Scope 3

In Scope 3

Out of Scope 3

Quality Objective 3

Roles and Responsibilities 3

2. Test Methodology 4

Overview 4

Test Levels 4

Bug Triage 5

Suspension Criteria and Resumption Requirements 6

Test Completeness 6

3. Test Deliverables 6

4. Resource & Environment Needs 7

Testing Tools 7

Test Environment 7

5. Terms/Acronyms 7

### Introduction

#### Scope

#### In Scope

The whole package including GUI, functional and non-functional requirements of the software are in the scope of this document and will be tested in different stages and sprints.

#### Out of Scope

None

#### Quality Objective

The ultimate objective of this plan is to ensure the application under test meets the functional and non-functional requirements as agreed on the contract. The Application Under Test will also be a quality and user friendly product with no defects.

#### Roles and Responsibilities

To ensure a clear understanding of the roles and responsibilities to achieve the above-mentioned quality objective the following roles and responsibilities are defined for this plan:

* Test Manager: has the overall responsibility of executing this plan. Test manager will schedule and implement the tests and set action plans to rectify the triaged bugs in cooperation with other team members
* QA Analyst: QA Analyst has the responsibility to monitor the proper implementation of this plan, perform regular audits and report to the Test Manager for corrections and Project Manager for further decisions.
* Configuration Manager
* Developers: are responsible to understand and analyze the assigned tasks and do the task on the assigned time ready for test. They will fix defined bugs during the tests reported by the Test Manager.
* Installation Team: They are responsible to ensure the software can be installed with minimum efforts and report any bugs to the Test Manager

### Test Methodology

#### Overview

Choosing the RAD software development requires a suitable test method like Agile in which all the functional and non-functional tests can be completed in each sprint. So the Agile method will be used for this testing plan as in Agile:

* software is developed in incremental, rapid cycles
* Interactions amongst customers, developers and client are emphasized
* focuses on responding to change rather than extensive planning
* every release of the project is tested thoroughly
* any bugs in the system are fixed before the next release

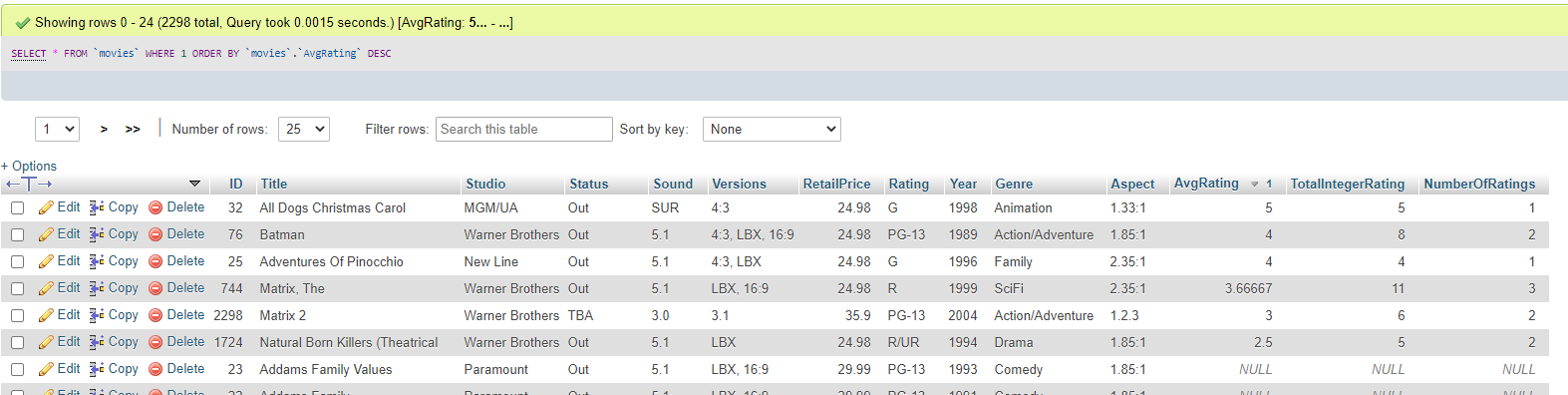
#### Test Levels

Test Levels define the Types of Testing to be executed on the Application Under Test (AUT).

The AUT will be tested in the below levels

* White Box Tests
  + Branch Coverage tests – can be performed on certain conditional operations. It is important that conditional bounds are tested on certain points, to ensure the application works well.
  + **Query Tests** – can be done with PHP or in a SQL test environment, depending on the type of query.

E.g. testing a query to get the most rated movies in the phpMyAdmin test environment:



**Testing a query in PHP** is as simple as using *echo $sql;* when performing a query on a page. The query will be printed so you can see if there are any errors. Unfortunately it can be hard to control how this gets displayed.



* Black Box Tests
  + **Functional tests** – where the new membership portal inputs will be tested. Sign up / sign in functions, administrative functions should be tested and documented.
  + **User interface** tests – will be used to test that the new members page follows the responsive design.

The following tests will be performed to ensure clients requirements:

* Functional and Regression Testing;
* GUI and Usability Testing;
* Accessibility Testing;
* Performance Testing;
* System / Integration Testing;
* User Acceptance Testing (UAT).

#### Bug Triage

To ensure fixing the bugs in a timely manner it is absolutely important to prioritize them so that the urgent ones, which are mainly functional reuquirements, get priority in scheduling the tasks to fix the bugs.

The triage would be based on the following requirements:

* Functional *(is now top priority as back-end coding plays a more important part for this sprint than user interface)*
* GUI and Usability
* Accessibility
* Performance

#### Suspension Criteria and Resumption Requirements

Due to size of the project, there is no suspension and resumption criteria.

#### Test Completeness

Test process will be considered complete if the following is met:

* 100% test coverage.
* All Manual & Automated Test cases executed
* All open bugs are fixed or will be fixed in next release

#### Test Deliverables

During different phases of the testing lifecycle the following deliverables should be delivered to ensure the testing process is completed and validated:

* Test Documentation Report
  + Test Plan
  + Test Cases
  + Bug Reports
  + Test Strategy
  + Test Metrics
* Customer Sign Off

### Resource & Environment Needs

#### Testing Tools

To run the testing plan the following tools are required and will be used:

* Test Management Tool
* Configuration management tool
* Static Analysis Tools [PHP Code Sniffer](https://github.com/squizlabs/PHP_CodeSniffer) with your desired coding standards

#### Test Environment

To test the application a test environment including hardware and software environement is equired in addition to the client specific ones.

Required hardware:

* PC or laptop
* Modem for internet connection
* Sufficient Disk Drive space

Required software:

1. Windows 10 / Mac OS / Linux
2. Email provider / individual preference
3. Git – to clone the source repository
4. XAMPP or similar web server software packages including MySQL and Apache
5. Text Editor such as Notepad++ or Visual Studio Code

### Terms/Acronyms

Make a mention of any terms or acronyms used in the project

| TERM/ACRONYM | DEFINITION |
| --- | --- |
| API | Application Program Interface |
| AUT | Application Under Test |

## OPTIMISATION REPORT

By Gholamreza

### Definition

In general, trying to achieve the best design and performance related to a set of prioritized criteria or contraints which is achievable by maxamizing the productivity, strength, reliability, longevity, efficiency, and utilization factors. This process of decision making is known as optimization process.

Code Optimization in Compiler Design. The code optimization in the synthesis phase is a program transformation technique, which tries to improve the intermediate code by making it consume fewer resources (i.e. CPU, Memory) so that faster-running machine code will result. (geeksforgeeks, 2020)

### Objectives

The main objectives of the code optimization can be:

* Increases the compilation’s speed by reduce the space usage.
* An optimized code often promotes re-usability.

### Types of Code Optimization

In general there are two types of optimization proces:

* Machine Independent Optimization

This code optimization phase attempts to improve the intermediate code to get a better target code as the output. The part of the intermediate code which is transformed here does not involve any CPU registers or absolute memory locations.

* Machine Dependent Optimization

Is normally done after the target code has been generated and when the code is transformed according to the target machine architecture. It involves CPU registers and may have absolute memory references rather than relative references. Machine-dependent optimizers put efforts to take maximum advantage of the memory hierarchy.

### Code Optimization Methods

Code Optimization is done in the following different ways :

* Compile Time Evaluation
* Variable Propagation
* Dead code elimination : Variable propagation often leads to making assignment statement into dead code
* Code Motion :  
  • Reduce the evaluation frequency of expression.  
  • Bring loop invariant statements out of the loop.
* Induction Variable and Strength Reduction :  
  • An induction variable is used in loop for the following kind of assignment i = i + constant.  
  • Strength reduction means replacing the high strength operator by the low strength.

### Where to apply

The best place for applying the optimization could be:

* Sourceprogram  
  Optimizing the source program involves making changes to the algorithm or changing the loop structures where the user is the actor.
* Intermediate Code  
  Optimizing the intermediate code involves changing the address calculations and transforming the procedure calls involved. Here compiler is the actor.
* Target Code  
  Optimizing the target code is done by the compiler. Usage of registers, select and move instructions is part of optimization involved in the target code.

### Phases of Optimization

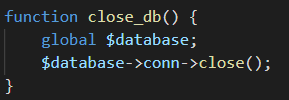
There are generally two phases of optimization:

* Global Optimization:  
  Transformations are applied to large program segments that includes functions, procedures and loops.
* Local Optimization:  
  Transformations are applied to small blocks of statements. The local optimization is done prior to global optimization.

## Optimisation of code

### Closing DB Connection

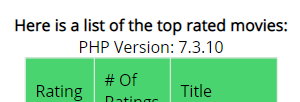
Prior to code optimisation, the database was kept open in all web pages. This is bad practise, as it can use more memory and could potentially be insecure. I implemented a close\_db() function which will close the mysqli\_connection object. This is called at the end of each web page.



### Checking PHP Version

Echo out the *PHP\_VERSION* constant in a convenient point in the page

echo “PHP Version: “ . PHP\_VERSION;

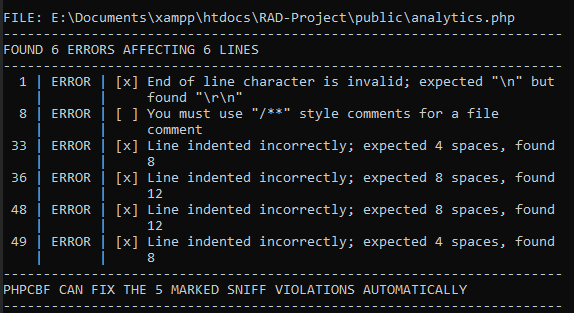


This is a very recent version of PHP. Anything lower than 7.0 may be slower.

### Static Analysis Tools

[PHP Code Sniffer](https://github.com/squizlabs/PHP_CodeSniffer) is a static analysis tool for PHP. Used via cmd:

php phpcs <files / directory>



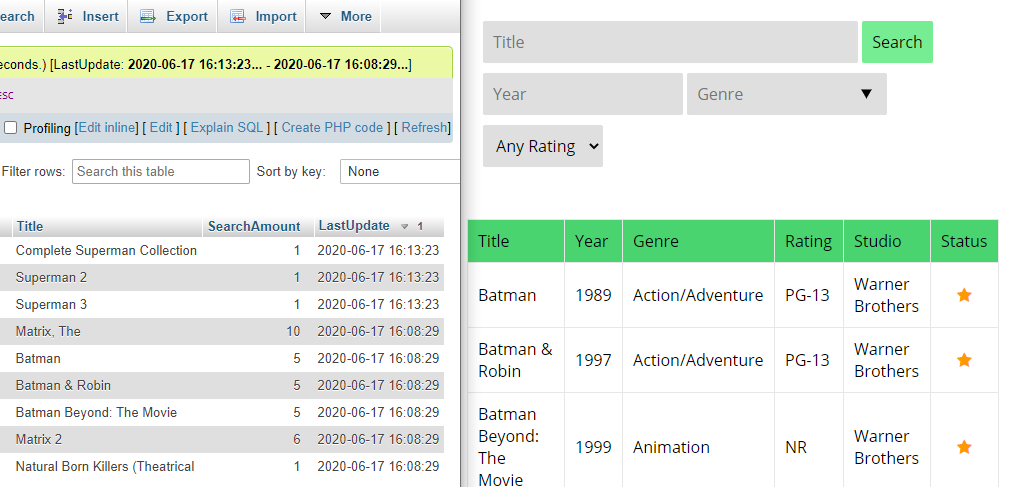
### Recommendations

Using a CDN, or **Content Delivery Network**, when hosting the website can improve users’ load times significantly. A CDN is a group of servers which are geographically distributed. They are spread out to provide end users with a closer and faster connection.

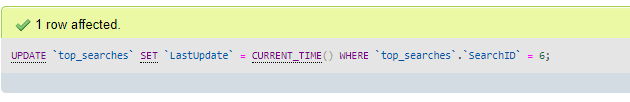
When hosting your website on a CDN, it is best to move your repository closer to your target audience. If most of your website users are from USA, it does not make sense to locate your repository in Australia.

## TEST DOCUMENTATION

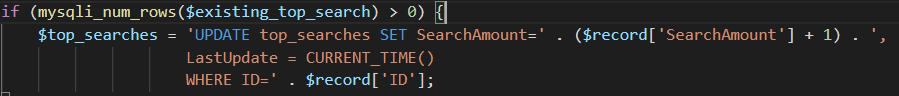
Fixing a error with analytics, LastUpdate field only changes when insertting new record, not when updating a search result.



To fix this, I updated a row in phpMyAdmin to see what SQL function was used to select the current time.

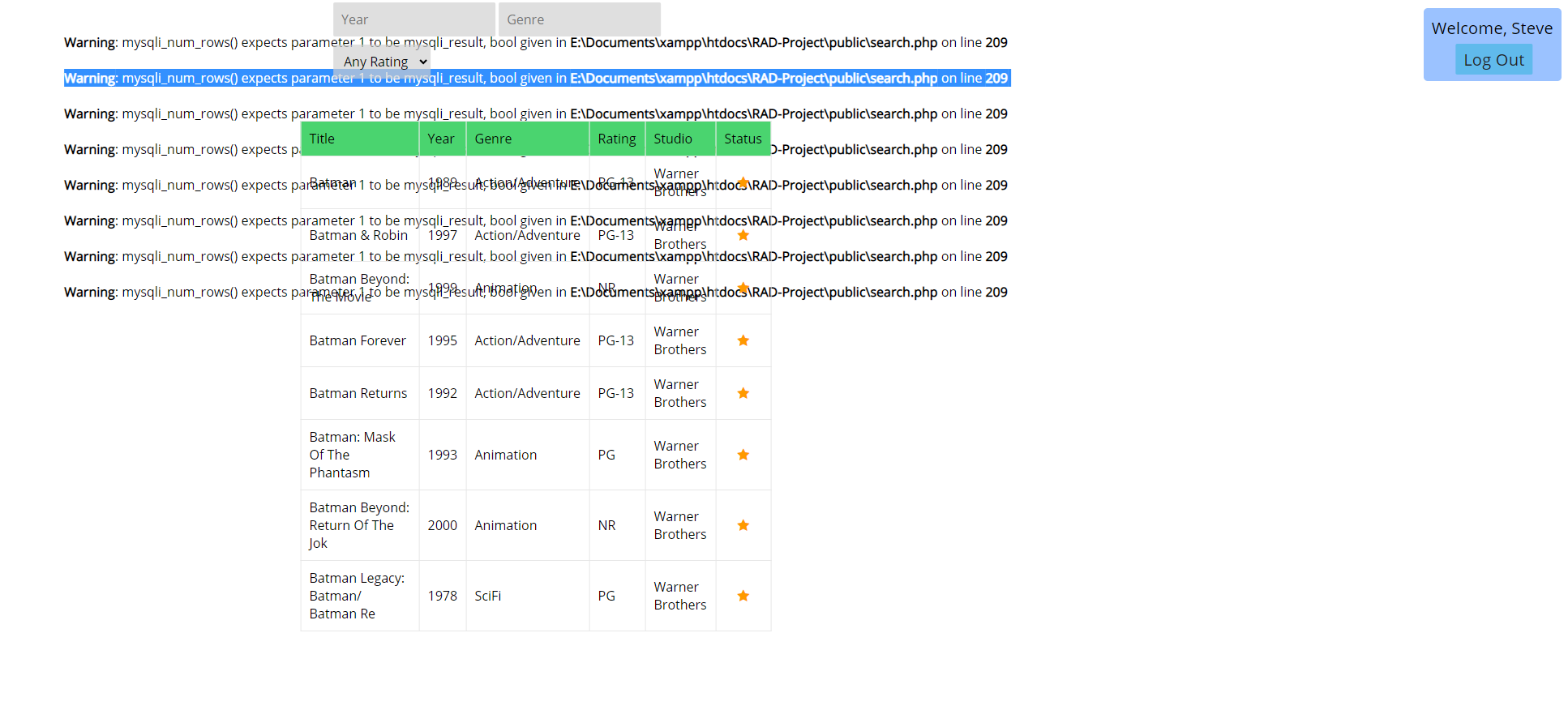


The UPDATE query now looks like this:

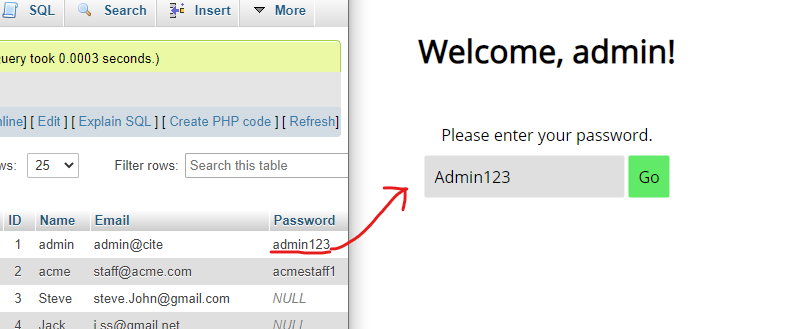


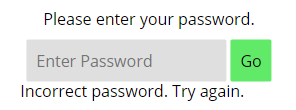
Errors are displayed if a member does not belong to a group, in the group\_members table.

When a new user signs up, they are now automatically placed in the Users group.

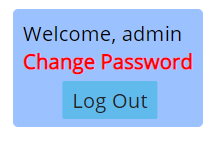


**Admin login**

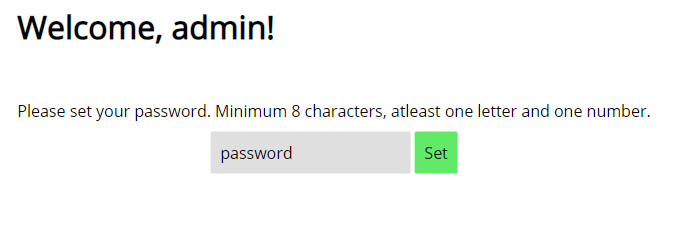


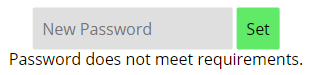
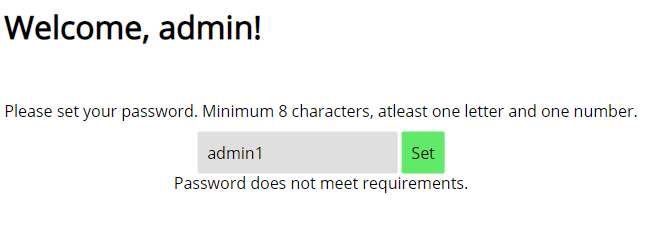
The capitalised password did not work.

**Changing / Setting password**

Pressing the ‘Change Password’ notification will redirect to members.php,

where Admins/ACME staff are prompted to set a password.





The password MUST be atleast 8 characters, and contain a letter and number.

## Analytics

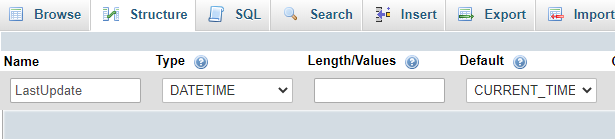
Top 10 movies can be queried from movies table, taking the AvgRating field in descending order. Each time a user rates a movie, their rating is saved to their member ID, and the movies table is also updated.

TotalIntegerRating, NumberOfRatings fields are used to calculate the AvgRating field.

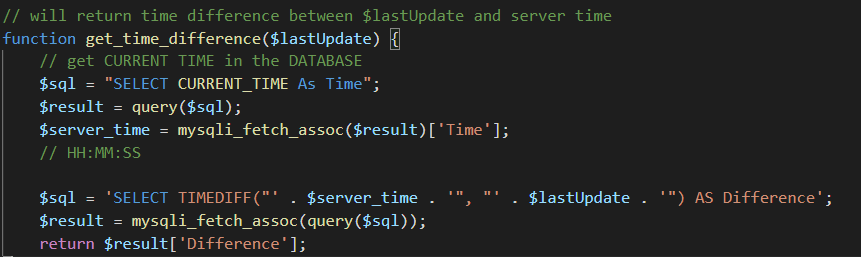
Currently streaming movies can be the most popular, or most searched for movies at the current time.

The current top\_searches table already stores the movies and the amount of times it was searched for. The SearchAmount field is incremented when the movie appears in somebodies top 3 search results.

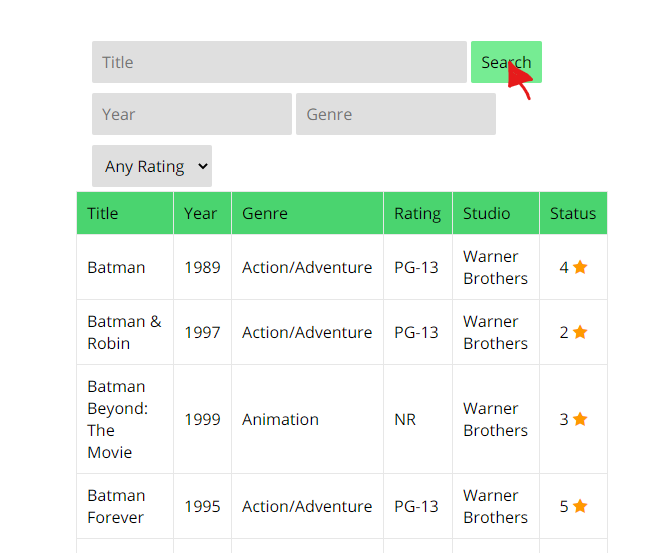
To display historical streaming data, it will have to contain a LastUpdate field, to compare with other recent searches.



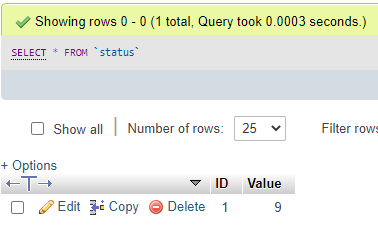
Using SQL TIMEDIFF(time1, time2) as a PHP query function, we could find the time since the last search.

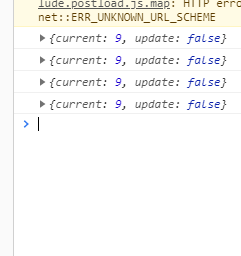


### Auto-refresh analytics on database update



In **search.php**, when a user clicks search, this Value is incremented in the status table.



In **analytics.php**, we send a POST request to a PHP script every 5 seconds.

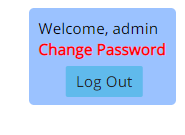
In **checker\_scr.php**, it checks if the client is up-to-date or behind the database.

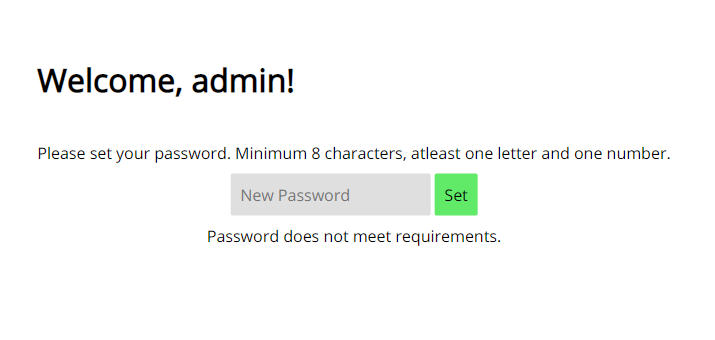
If the client needs to be updated, the POST request will return *update: true* and the analytics page will refresh.

## Password Authentication

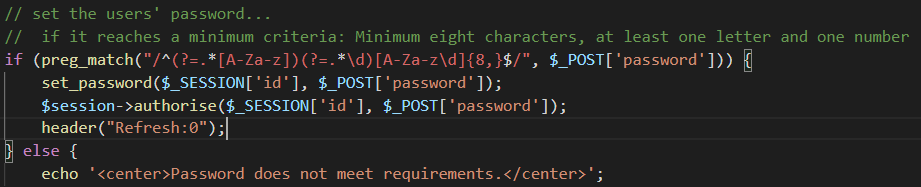
Passwords are required to be set for ACME and Admin accounts. This is done by testing if the Passwords field is null or empty for those users which belong to ACME and Admin group.

If their password is not set, they are told to change password immediately:



Clicking Change Password redirects to members.php, which can only be accessed when Admins or ACME users have a password set.

The password is validated against a Regular Expression using the PHP function preg\_match()



This will set the $\_SESSION[password], and will also set the password in the database.

When logging in, it is a similar process, except the set\_password() function and preg\_match(RegEx, str) functions are not used.

# Handover

[Performance Report](#_PERFORMANCE_REPORT) was completed by Gholamreza as Troy was unable to finish Sprint Two

## Software Review Report

Gholamreza Aminy and Sean Boaden were the sole members after Sprint One. Tasks were divided equally between the two remaining members.

### Requirements Checklist

**Reports**

|  |  |
| --- | --- |
| Project Management Plan, Software Testing Plan, Source Control documentation. Updated for each Sprint | ✓ |
| (S1) Analysis Report | ✓ |
| (S1) Multi-Platform Report | ✓ |
| (S1) Test Documentation for multi-platform accessibility | ✓ |
| (S2) Software Review Plan | ✓ |
| (S2) Performance Report | Completed during Handover |
| (S3) Optimisation Report | ✓ |

**Software**

|  |  |
| --- | --- |
| Movie Database Application – searching and filtering movies is quick and easy | ✓ |
| (S1) The application demonstrates either Adaptive or Responsive web design techniques | ✓ |
| (S2) **Membership Portal and Administrator Functions**  New members can sign up and choose a mailing option  Admins can view and edit all members | ✓ |
| (S3) **Passwords** are required for Admin and ACME accounts | ✓ |
| (S3) All users are assigned a Group. When a member joins, they are placed in the default group User | ✓ |
| (S3) **If the user is signed in, they are able to rate a movie 1 - 5 stars**  User ratings are saved in a DB table  Average Rating is calculated when a user rates a movie | ✓ |
| (S3) **When a movie is searched, analytics page is automatically updated**  The LastUpdate field in top\_searches table is used to find the most recently searched movies (historical data) | ✓ |
| (Handover) **Web Application incorporates Web Content Accessibility Guidelines**  Text is readable when zooming in to 200%  All non-text elements have alternative-text attributes  Colours are suitable for colour blind | ✓ |
| (Handover) Application functions correctly on a chosen platform | ✓ |
| (Handover) **Administrators can access DB**  -Viewing all members  -Editing mailing options | ✓ |

### Code Testing Process

When testing the code, it is important to know what features you need to test, and the methodology you need to use. The code testing process is set out in the Software Testing Plan. It defines the types of tests to perform for each sprint, and the issues to prioritise when coding and bugfixing.

For example, Sprint 1 saw a focus on User Interface / UX testing as this sprint was where the responsive design was implemented. Sprint 2 was focused on Functionality tests and System testing as the membership and PHP Sessions were implemented.

After each team member had familiarised themselves with the Test plan for the current sprint, they are required to take screenshots of the tests they performed, and provide an explanation of the test/output. These were compiled into the Test Documentation sections in the Master Document.

### Quality Assurance

Before coding, everybody knew the functional and non-functional requirements of the sprint. This way, members knew what features to develop and test before adding anything else. Team members finished reports prior to moving onto web development.

To ensure that all members knew what to do, we held meetings where the Scrum Master would assign tasks and give a run-down of the sprint. Discussions were encouraged. It is important to take notes of meetings for documentation purposes.

After development, the team was expected to perform testing according to the Test Plan, so that in the next sprint there would be little or no errors.

### Quality Control Processes

Testing was performed – including User Interface, System, Integration, Acceptance Testing.

Reviews were performed

* Code reviews, where the developer reads their code line-by-line
  + GitHub alows for changes to be reviewed by anyone. Pull-requests and commit history will highlight deleted lines of code as red, while new lines will be highlighted green.
* Requirements reviews
* Test Plan Reviews – as required, the Software Test Plan was altered each sprint to suit the needs of the tasks at hand.

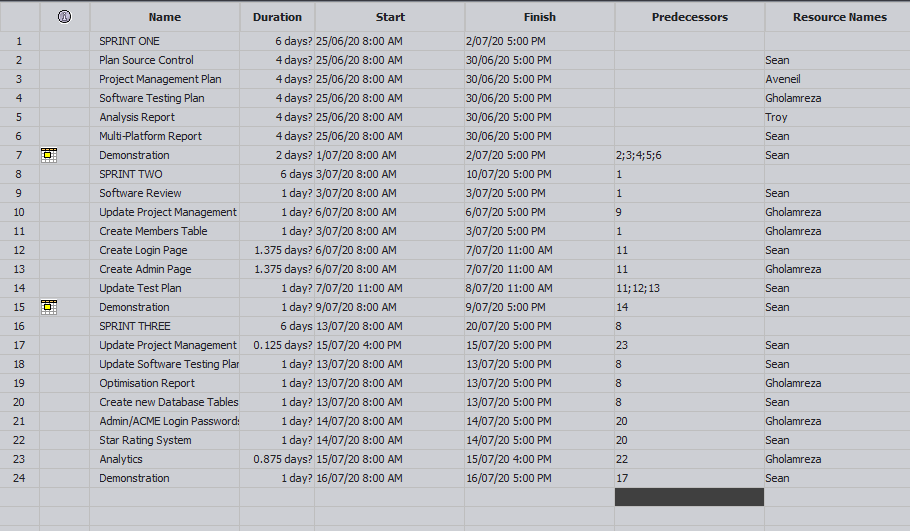
### Room for Future Modifications

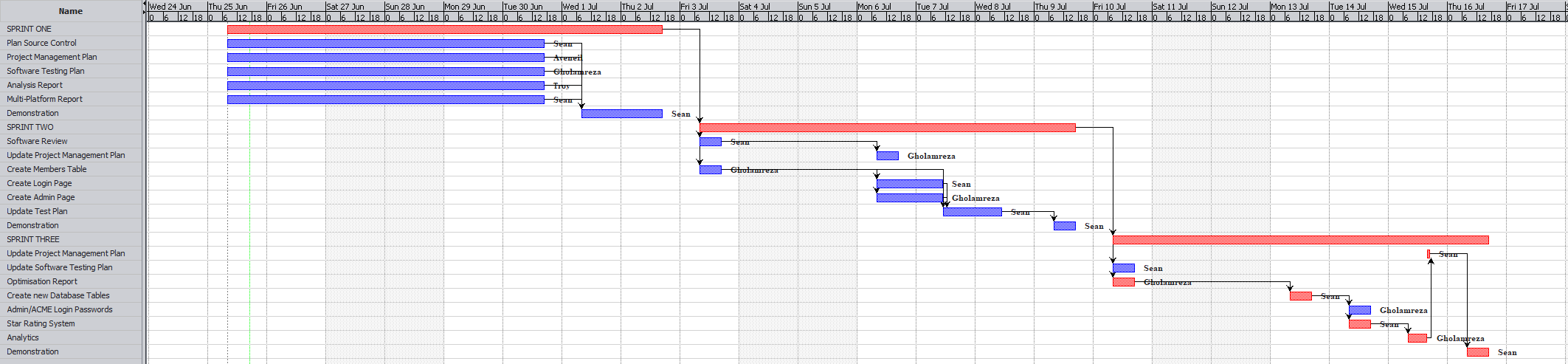
Each team member recognised that they must provide sufficient commenting for the future developers who may work on this application.

It is highly recommended that the source control is kept by ACME as it contains useful reports and documentation for any future developers.

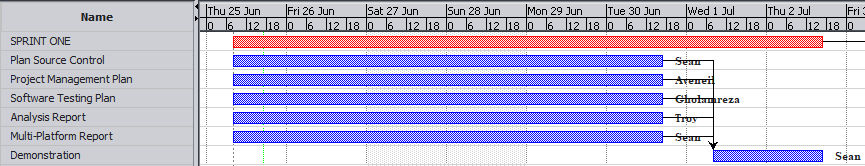
## Project Management Plan

**Sprints 1 - 3**

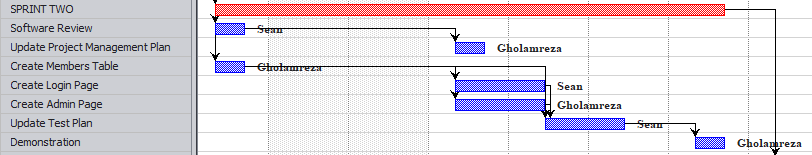




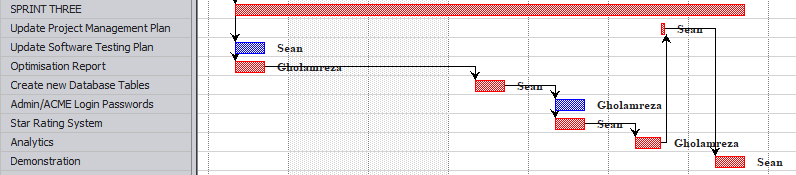
**Sprint 1**



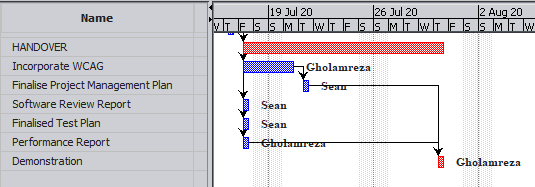
**Sprint 2**



**Sprint 3**



**Handover**



## Test plan

**Project: Movie Database**

**Client: Acme Entertainment Pty Ltd**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| V0 | 28/5/2020 | *Gholamreza Aminy* | First Version |
| V1 | 7/06/2020 | *Sean Boaden* | Second Version – updated to meet requirements of Sprint Two |
| V2 | 10/06/2020 | Sean Boaden | Updated to meet requirements of Sprint Three |
| V3 | 29/06/2020 | Sean Boaden | Updated and finalised for project handover |

Test plan 1

1. Introduction 3

Scope 3

In Scope 3

Out of Scope 3

Quality Objective 3

Roles and Responsibilities 3

2. Test Methodology 4

Overview 4

Test Levels 4

Bug Triage 5

Suspension Criteria and Resumption Requirements 6

Test Completeness 6

3. Test Deliverables 6

4. Resource & Environment Needs 7

Testing Tools 7

Test Environment 7

5. Terms/Acronyms 7

### Introduction

#### Scope

#### In Scope

The whole website software package, including user interface and client-side *(JS)* and server-side scripts *(PHP)*, functional and non-functional requirements of the software are in the scope of this document and will be tested in different stages and sprints.

#### Out of Scope

None

#### Quality Objective

The ultimate objective of this plan is to ensure the web-application under test meets the functional and non-functional requirements as agreed on the contract assigned to *CITE Managed Services* by *ACME Entertainment Pty. Ltd*. The Application Under Test *(AUT)* will be a quality and user friendly product with no defects.

#### Roles and Responsibilities

To ensure a clear understanding of the roles and responsibilities to achieve the above-mentioned quality objective the following roles and responsibilities are defined for this plan:

* **Test Manager**: has the overall responsibility of executing and updating this plan. Test manager will schedule and implement the tests and set action plans to rectify the triaged bugs in cooperation with other team members
* **QA Analyst**: QA Analyst has the responsibility to monitor the proper implementation of this plan, perform regular audits and report to the Test Manager for corrections and Project Manager for further decisions.
* **Configuration Manager**
* **Developers**: are responsible to understand and analyze the assigned tasks and do the task by a deadline ready for test. They will fix bugs during testing phases reported by the **Test Manager**.
* **Installation Team**: They are responsible to ensure the software can be installed and operated with minimum efforts and report any bugs to the Test Manager

### Test Methodology

#### Overview

Choosing the RAD software development requires a suitable test method like Agile in which all the functional and non-functional tests can be completed in each sprint. So the Agile method will be used for this testing plan as in Agile:

* software is developed in incremental, rapid cycles
* Interactions amongst customers, developers and client are emphasized
* focuses on responding to change rather than extensive planning
* every release of the project is tested thoroughly
* any bugs in the system are fixed before the next release

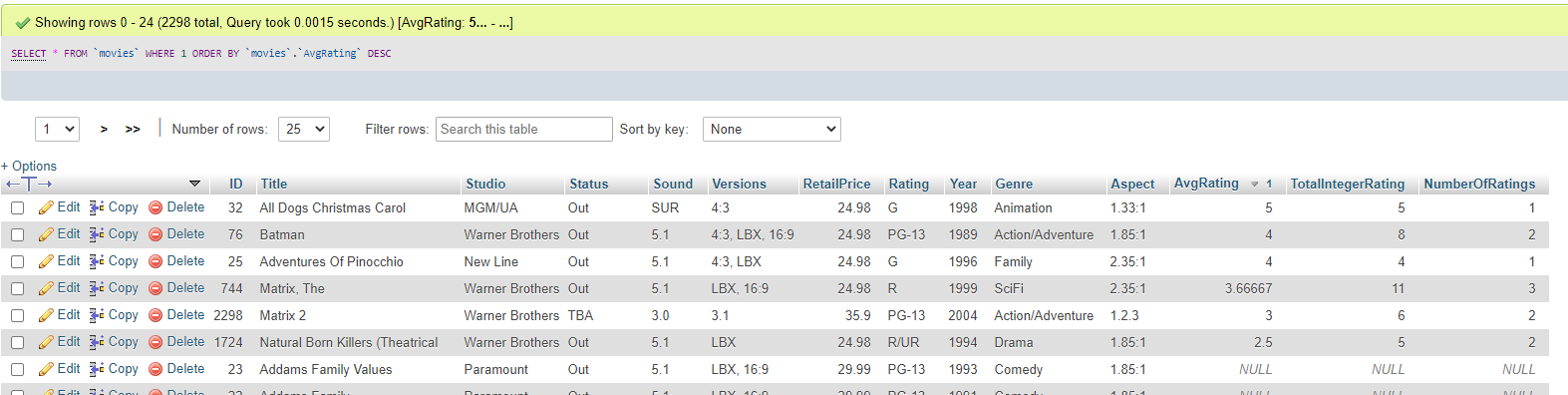
#### Test Levels

Test Levels define the Types of Testing to be executed on the Application Under Test (AUT).

The AUT will be tested in the below levels

* White Box Tests
  + **Branch Coverage tests** – can be performed on certain conditional operations. It is important that conditional bounds are tested on certain points, to ensure the application works well.
  + **Query Tests** – can be done with PHP or in a SQL test environment *(phpMyAdmin)*, depending on the type of query.

E.g. testing a query to retrieve the most rated movies in the phpMyAdmin test environment:



**Testing a query in PHP** is as simple as using *echo $sql;* when performing a query on a page. The query will be printed so you can see if there are any errors. Unfortunately it can be hard to control how this gets displayed.



* Black Box Tests
  + **Functional tests** – where the new membership portal inputs will be tested. Sign up / sign in functions, administrative functions should be tested and documented.
  + **User interface** tests – will be used to test that the new members page follows the responsive design.

The following tests will be performed to ensure clients requirements:

* Functional and Regression Testing;
* GUI and Usability Testing;
* Accessibility Testing;
* Performance Testing;
* System / Integration Testing;
* User Acceptance Testing (UAT).

#### Bug Triage

To ensure fixing the bugs in a timely manner it is absolutely important to prioritize them so that the urgent ones, which are mainly functional reuquirements, get priority in scheduling the tasks to fix the bugs.

The triage would be based on the following requirements:

* **Accessibility** – *WCAG is implemented, all non-text images have alternative text, page is readable up to 200% zoom*.
* GUI and Usability
* Functionality
* Performance

#### Suspension Criteria and Resumption Requirements

Due to size of the project, there is no suspension or resumption criteria.

#### Test Completeness

Test process will be considered complete if the following is met:

* 100% test coverage.
* Web Content Accessibility Guidelines
* All Manual & Automated Test cases executed
* All open bugs are fixed or will be fixed in next release

### Test Deliverables

During different phases of the testing lifecycle the following deliverables should be delivered to ensure the testing process is completed and validated:

* Test Documentation Report
  + Test Plan
  + Test Cases
  + Bug Reports
  + Test Strategy
  + Test Metrics
* Customer Sign Off

### Resource & Environment Needs

#### Testing Tools

To run the testing plan the following tools are required and will be used:

* Test Management Tool
* Configuration management tool
* Static Analysis Tools [PHP Code Sniffer](https://github.com/squizlabs/PHP_CodeSniffer) with your desired coding standards

#### Test Environment

To test the application a test environment including hardware and software environement is equired in addition to the client specific ones.

Required hardware:

* PC or laptop
* Modem for internet connection
* Sufficient Disk Drive space

Required software:

1. Windows 10 / Mac OS / Linux
2. Email provider / individual preference
3. Git – to clone the source repository
4. XAMPP or similar web server software packages including MySQL and Apache
5. Text Editor such as Notepad++ or Visual Studio Code

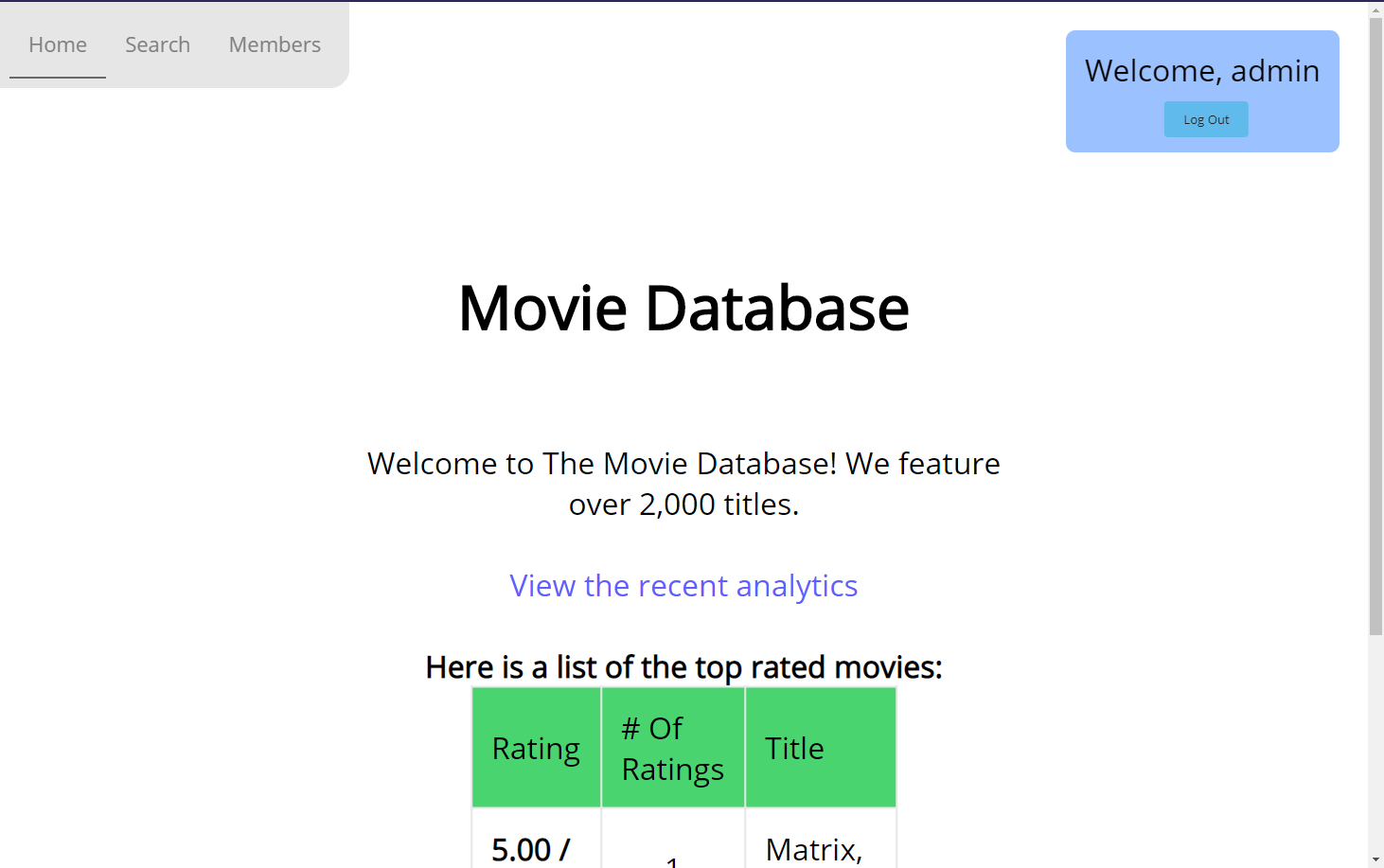
### Terms/Acronyms

Make a mention of any terms or acronyms used in the project

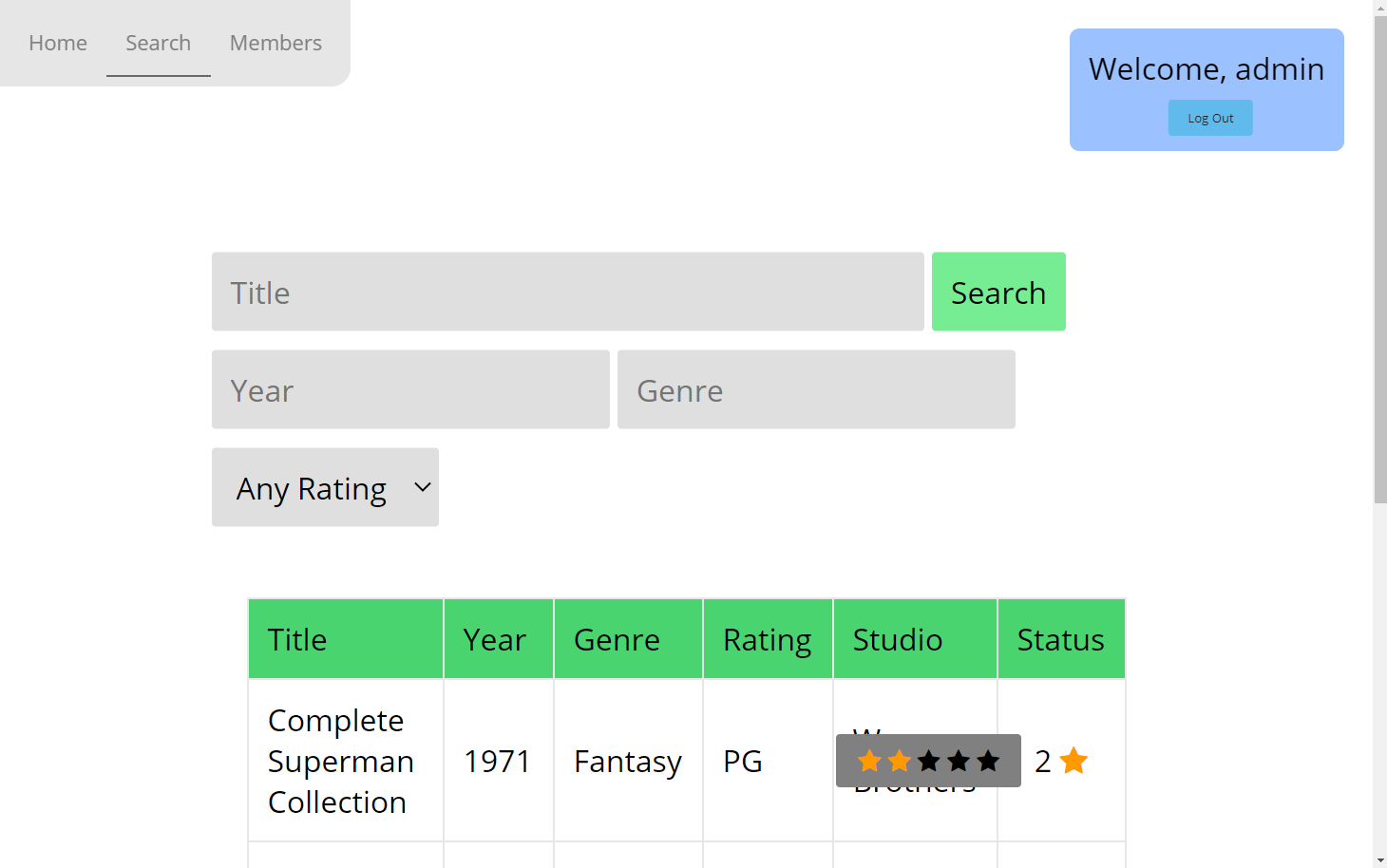
| TERM/ACRONYM | DEFINITION |
| --- | --- |
| API | Application Program Interface |
| AUT | Application Under Test |

## WCAG Test Documentation

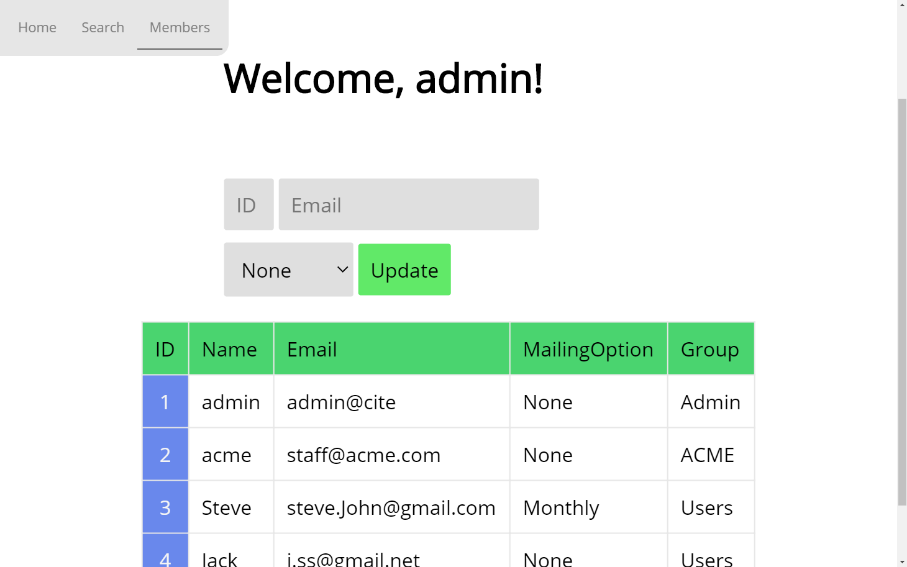
**Home Page – 200% Zoom**



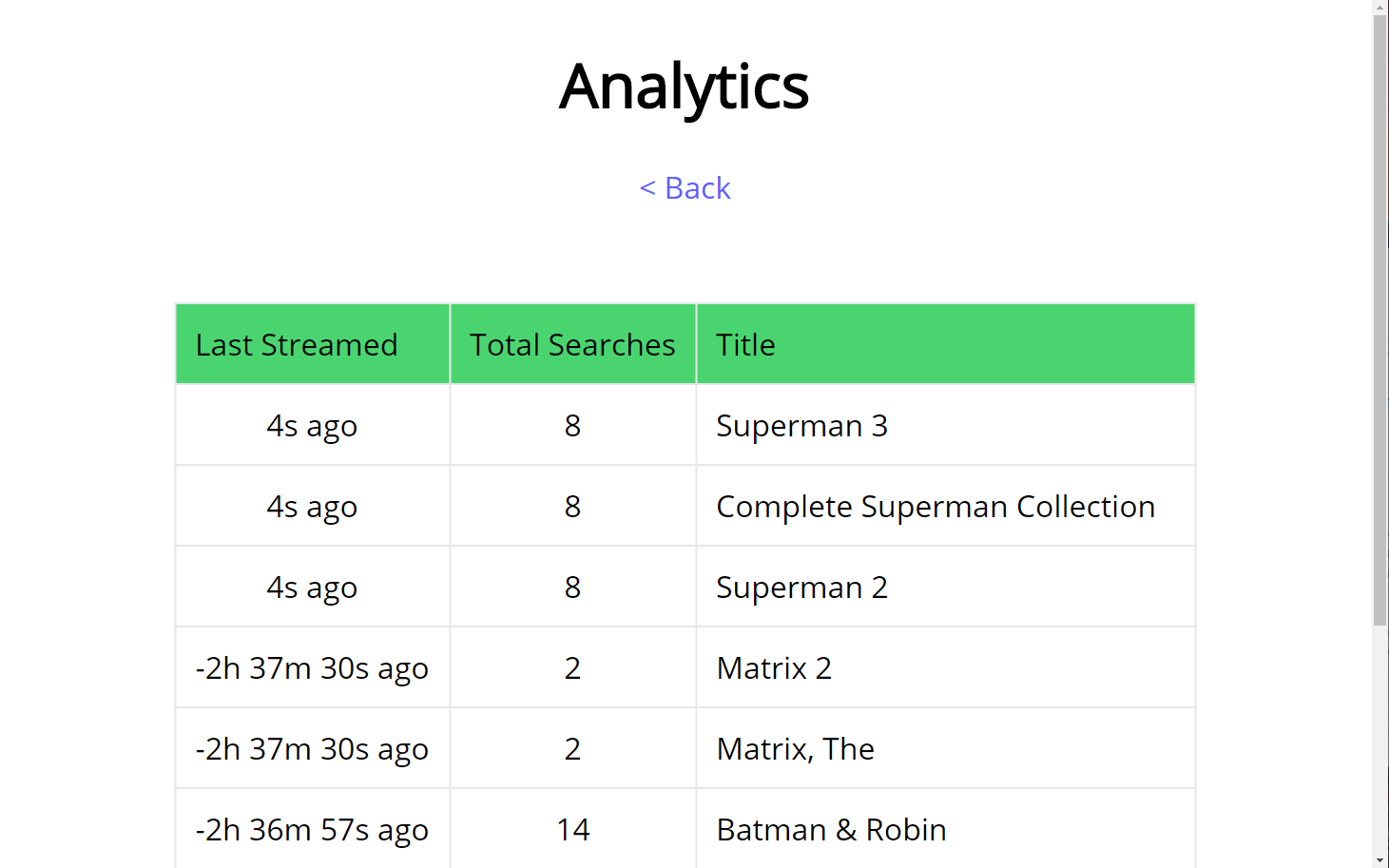
**Search Page – 200% Zoom**



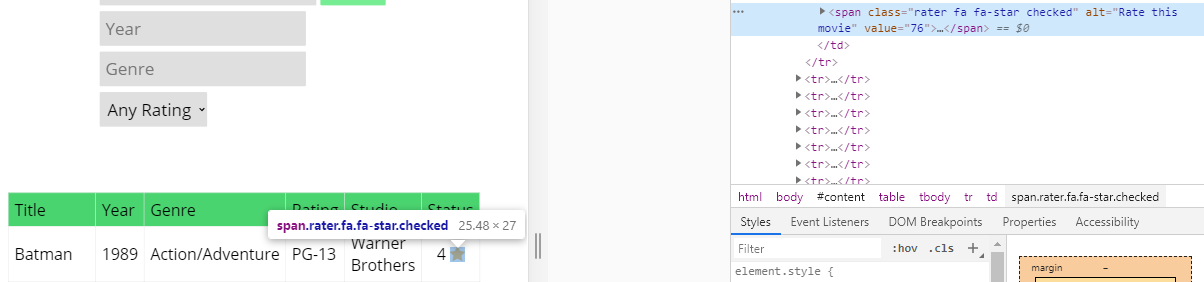
**Members Page – 200% Zoom**



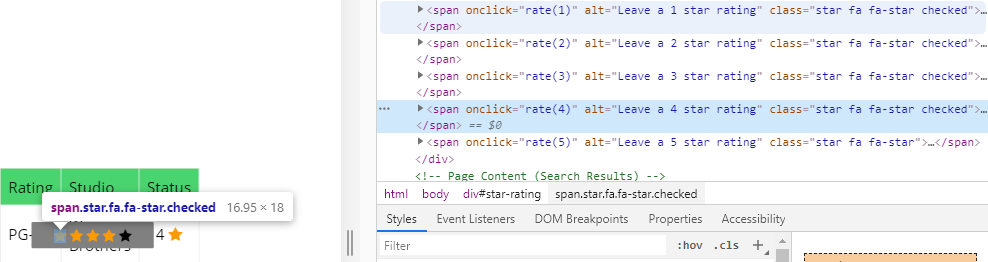
**Analytics – 200%**



**Alternative Text – Star Rating**



**Alt-Text – Stars 1 to 5**



# References

CITE business rules for software development: <http://www.citems.com.au/?page_id=74>

Functional vs. Nonfunctional Req. - <https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/>

Software Verification - <https://blackboard.southmetrotafe.wa.edu.au/bbcswebdav/pid-1719638-dt-content-rid-15646676_1/courses/16_C_EIO_BUSAPPCLU_1/verification.pdf>

[Software Quality Control](http://softwaretestingfundamentals.com/software-quality-control/#:~:text=Software%20Testing%20Jokes-,Software%20Quality%20Control,the%20products%20meet%20specifications%2Frequirements.) – SoftwareTestingFundamentals.com