RAD Project

South Metro TAFE | Burslem Drive

Kyle Cleofe; JASON King; CHRISTINA Tatang

2019

Table of Contents

[Introduction 6](#_Toc26393454)

[Source Control 6](#_Toc26393455)

[Project Management Plan 6](#_Toc26393456)

[Analysis Documentation 8](#_Toc26393457)

[CITE business rules for software development 8](#_Toc26393458)

[CITE Managed Services Quality Assurance 9](#_Toc26393459)

[Acme Entertainment Pty Ltd development requirements 9](#_Toc26393460)

[**Sprint One** 10](#_Toc26393461)

[Multi-platform report (adaptive vs responsive) 11](#_Toc26393462)

[Responsive Design 11](#_Toc26393463)

[Adaptive design 11](#_Toc26393464)

[Test Plan 13](#_Toc26393465)

[Introduction 13](#_Toc26393466)

[Scope 13](#_Toc26393467)

[Quality Objective 13](#_Toc26393468)

[Roles and Responsibilities 13](#_Toc26393469)

[Test Methodology 14](#_Toc26393470)

[Overview 14](#_Toc26393471)

[Test Levels 14](#_Toc26393472)

[Bug Triage 15](#_Toc26393473)

[Suspension Criteria and Resumption Requirements 15](#_Toc26393474)

[Test Completeness 15](#_Toc26393475)

[Test Deliverables 15](#_Toc26393476)

[Resource & Environment Needs 16](#_Toc26393477)

[Testing Tools 16](#_Toc26393478)

[Test Environment 16](#_Toc26393479)

[Testing Document 17](#_Toc26393480)

[Scope 17](#_Toc26393481)

[Testing Case 17](#_Toc26393482)

[Responsive Layout 18](#_Toc26393483)

[Home Page 18](#_Toc26393484)

[Desktop 18](#_Toc26393485)

[Tablet 19](#_Toc26393486)

[Mobile 20](#_Toc26393487)

[Frequent Search Chart 21](#_Toc26393488)

[Desktop 21](#_Toc26393489)

[**Sprint Two** 22](#_Toc26393490)

[Software Review Plan 23](#_Toc26393491)

[Purpose and objective 23](#_Toc26393492)

[Software Review Criteria 23](#_Toc26393493)

[Performance Report 25](#_Toc26393494)

[Code Optimisers 25](#_Toc26393495)

[Types and Levels of Optimization 25](#_Toc26393496)

[What to Optimize 25](#_Toc26393497)

[Good and Bad Outcomes of Optimization 25](#_Toc26393498)

[Performance Tools 26](#_Toc26393499)

[WebLOAD 26](#_Toc26393500)

[SmartMeter.io 26](#_Toc26393501)

[Apache JMeter 26](#_Toc26393502)

[Test Plan 27](#_Toc26393503)

[Introduction 27](#_Toc26393504)

[Scope 27](#_Toc26393505)

[Quality Objective 27](#_Toc26393506)

[Roles and Responsibilities 27](#_Toc26393507)

[Test Methodology 28](#_Toc26393508)

[Overview 28](#_Toc26393509)

[Test Levels 28](#_Toc26393510)

[Bug Triage 29](#_Toc26393511)

[Suspension Criteria and Resumption Requirements 29](#_Toc26393512)

[Test Completeness 29](#_Toc26393513)

[Test Deliverables 29](#_Toc26393514)

[Resource & Environment Needs 30](#_Toc26393515)

[Testing Tools 30](#_Toc26393516)

[Test Environment 30](#_Toc26393517)

[Testing Document (Sprint Two) 31](#_Toc26393518)

[Scope 31](#_Toc26393519)

[Changelog 31](#_Toc26393520)

[Testing Case 31](#_Toc26393521)

[**Sprint Three** 34](#_Toc26393522)

[Test Plan (Sprint Three) 35](#_Toc26393523)

[Introduction 35](#_Toc26393524)

[Scope 35](#_Toc26393525)

[Quality Objective 35](#_Toc26393526)

[Roles and Responsibilities 35](#_Toc26393527)

[Test Methodology 36](#_Toc26393528)

[Overview 36](#_Toc26393529)

[Test Levels 36](#_Toc26393530)

[Bug Triage 37](#_Toc26393531)

[Suspension Criteria and Resumption Requirements 37](#_Toc26393532)

[Test Completeness 37](#_Toc26393533)

[Test Deliverables 37](#_Toc26393534)

[Resource & Environment Needs 38](#_Toc26393535)

[Testing Tools 38](#_Toc26393536)

[Test Environment 38](#_Toc26393537)

[Software Review Plan (Sprint Three) 39](#_Toc26393538)

[Purpose and objective 39](#_Toc26393539)

[Software Review Criteria 39](#_Toc26393540)

[Optimisation report 41](#_Toc26393541)

[Introduction 41](#_Toc26393542)

[Performance tool used 41](#_Toc26393543)

[PHP\_CodeSniffer and PHP Beautifier 41](#_Toc26393544)

[WebLOAD 41](#_Toc26393545)

[WebLOAD Summary 41](#_Toc26393546)

[PHP\_CodeSniffer and PHP Code beautifier summary 43](#_Toc26393547)

[Optimisation conclusion 46](#_Toc26393548)

[Testing Document 48](#_Toc26393549)

[Scope 48](#_Toc26393550)

[Device Requirements 48](#_Toc26393551)

[Computers 48](#_Toc26393552)

[Tablets and Mobiles 48](#_Toc26393553)

[Apple 48](#_Toc26393554)

[Android 48](#_Toc26393555)

[Testing Case 48](#_Toc26393556)

[Responsive Layout 51](#_Toc26393557)

[Admin Page 52](#_Toc26393558)

[Desktop 1920x1080 52](#_Toc26393559)

[Tablet 800 x 1280 Portrait 53](#_Toc26393560)

[Tablet 1280x800 Landscape 53](#_Toc26393561)

[Mobile 375x667 Portrait e.g. Iphone 6 54](#_Toc26393562)

[Mobile 667x375 Landscape e.g. Iphone 6 54](#_Toc26393563)

[Frequent Search Chart 55](#_Toc26393564)

[Desktop 1920x1080 55](#_Toc26393565)

[Tablet 1280x800 Landscape 55](#_Toc26393566)

[Tablet 1280x800 Portrait 56](#_Toc26393567)

[Mobile 375x667 Portrait e.g. Iphone 6 57](#_Toc26393568)

[Mobile 667x375 Landscape e.g. Iphone 6 57](#_Toc26393569)

[Top 10 most Liked 58](#_Toc26393570)

[Desktop 1920x1080 58](#_Toc26393571)

[Ipad 58](#_Toc26393572)

[Iphone 6/7/8 59](#_Toc26393573)

[Glossary 60](#_Toc26393574)

*ChangeLog*

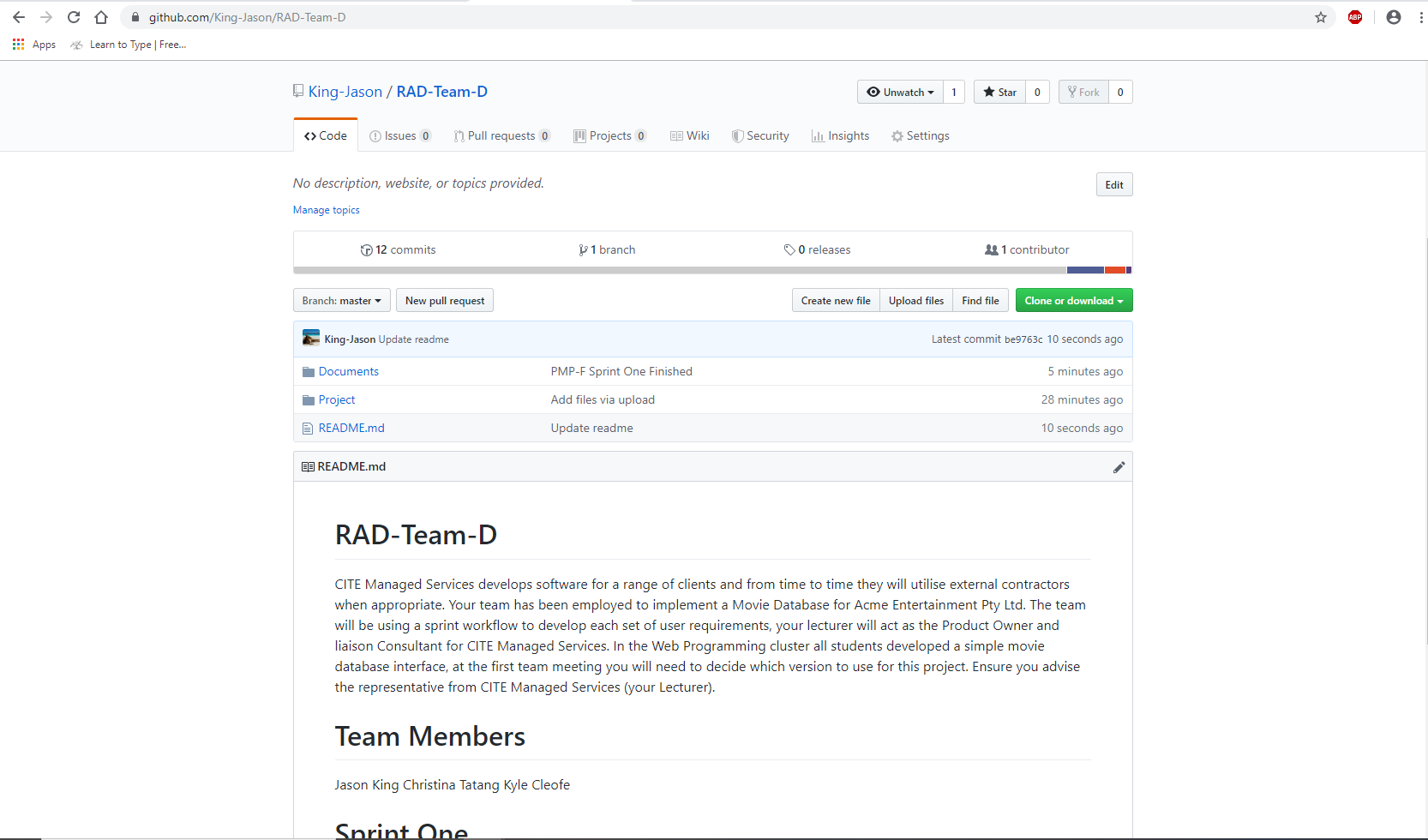
|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Change Date** | **By** | **Description** |
| version number | Date of Change | Name of person who made changes | Description of the changes made |
| V1 | 7/11 | Kyle, Jason | Made master documentation and added analysis.doc |
| V2 | 14/11/2019 | Jason, Kyle, Christina | Edited Analysis documentation, Added Mutli-platform report, Added Test plan, Added Intro, Added Testing Document |
| V2.1 | 20/11 | Kyle | Added and updated testing plan, edited document to be more clear. |
| V3 | 21/11 | Kyle, Christina, Jason | Added Sprint two documents: Performance report, Software review plan and updated software testing plan. |
| V4 | 28/11 | Kyle, Christina, Jason | Added Sprint Three documents: Optimisation report, testing document, updated software testing plan updated software review plan. |
| V4.1 | 28/11 | Kyle | Updated optimisation report |
| V4.2 | 04/12 | Kyle | Added tests on clientpage.php (like and dislike feature) and mostliked.php (analytics on most liked movie) |

# Introduction

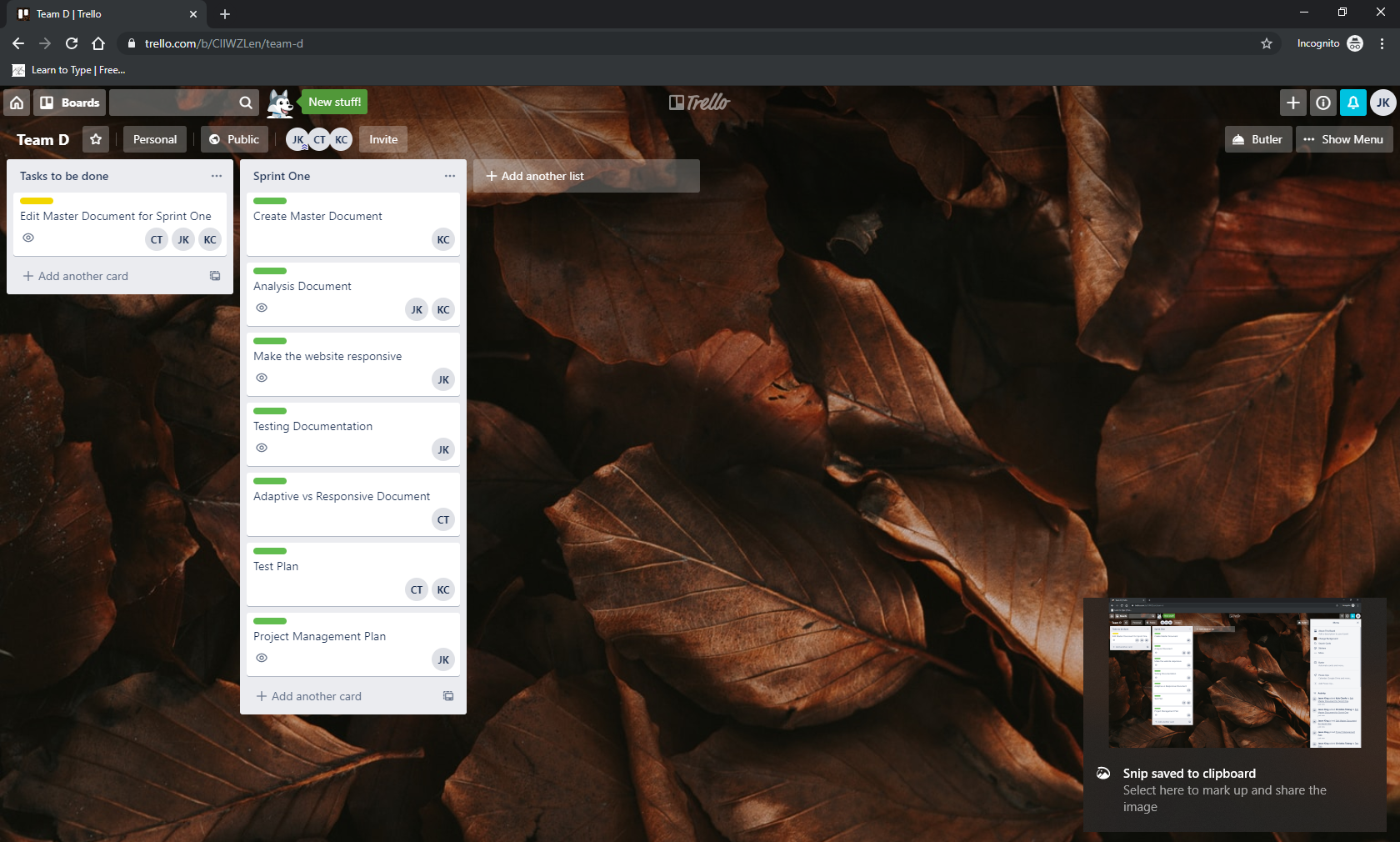
This document is a collection of other subdocuments that are required for the revision of the ACME movie database application.

## Source Control

We are using GitHub as a source control. Here is a link to our source control <https://github.com/King-Jason/RAD-Team-D>



## Project Management Plan



We are using Trello as a project management and collaboration the link is provided below.

<https://trello.com/b/CIlWZLen/team-d>

# Analysis Documentation

## CITE business rules for software development

Our mission is to help enterprises accelerate adoption of new technologies, untangle complex issues that always emerge during digital evolution, and orchestrate ongoing innovation. Whether it is a consumer-oriented app or a trans-formative enterprise class solution, the company leads the process from idealization and concept to delivery, and provides ongoing support through ISO standards and frameworks.

**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 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.

**Integrity**

We expect our people to maintain high ethical standards in everything they do.

Ref= <http://www.citems.com.au/?page_id=62>

## CITE Managed Services 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.

Ref= <http://www.citems.com.au/?page_id=84>

## Acme Entertainment Pty Ltd development requirements

* Review and Update the move database application. Rework the prototype.
* Make a multiplatform report document – adaptive or responsive.
* Move Database local server or cloud hosted.
* Record testing and make a testing document.

CITE Managed Services develops software for a range of clients and from time to time they will utilise external contractors when appropriate. Team D (us) have been employed to implement a Movie Database for Acme Entertainment Pty Ltd.

# **Sprint One**

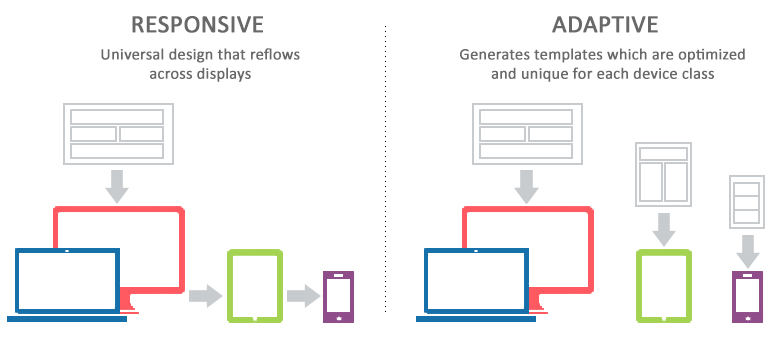
# Multi-platform report (adaptive vs responsive)

Responsive sites and adaptive sites are the same in that they both change appearance, based on the browser environment they are being viewed on. Both responsive and adaptive design have a very similar objective. They seek to provide the best possible experience on a single website when it’s viewed on a multi-platform (desktop, mobile, tablet devices). In this report, we will explaining the difference between responsive and adaptive design.

## Responsive Design

Responsive Design is the use media queries to serve different CSS allowing a site to change its appearance when viewed on different size screens. So in short it can be accomplished with just plain HTML and CSS. Responsive websites respond to the size of the browser at any given point. No matter what the browser width may be, the site adjusts its layout (any perhaps functionality) in a way that is optimized to the screen.

Responsive design is client-side which means the page is sent to the device browser (the client), and the browser then modifies the appearance of the page in relation to the size of the browser window.



The pros of responsive web design:

1. User experience friendly.
2. Save cost on responsive web design development
3. SEO friendly

The cons of responsive web design:

1. Bad compatibility for the old version of IE browser
2. Slow down the loading
3. Time-consuming on the development
4. Influence on the layout

## Adaptive design

Adaptive design uses predefined layouts that have been carefully constructed for a variety of screen sizes. A particular layout is activated when the screen of the device viewing the website is detected and matched with a style sheet. In other words, the website is only concerned about the browser being a specific width, at which point it adapts the layout.

Adaptive design is predominantly server side. This means that the web server does all of the work of detecting the various devices and loading the correct style sheet depending on the attributes of the device. As well as the layout changing depending on the size of the screen they can also change depending on conditions like whether the device has a retina display or not.

The server can detect this and display high quality images for retina display devices like iPads and lower quality images for standard-definition displays.



The pros of adaptive design:

1. Faster page loads
2. Optimized user experience
3. Improved SEO
4. Serving a wider audience

The cons of adaptive website design:

1. Labour-intensive
2. Professional required

As the result, our team (team D) will use responsive design for our group project. One of the reasons why we choose responsive design because it is easier and takes less work to implement. It affords less control over the design on each screen size, but it’s by far the preferred method for creating new sites ate this moment. Responsive designers create a single design to be used on all screens and will generally start in the middle of the resolution and use media queries to determine what adjustments will be made for the lower and higher end of the resolution scale. This will tends make users happy, because that familiar web design seems guaranteed to translate across to any device’s screen.

**References**

<https://medium.muz.li/responsive-and-adaptive-design-69604ab1abd4>

<https://css-tricks.com/the-difference-between-responsive-and-adaptive-design/>

<https://www.uxpin.com/studio/blog/responsive-vs-adaptive-design-whats-best-choice-designers/>

<https://www.interaction-design.org/literature/article/adaptive-vs-responsive-design>

<https://www.mockplus.com/blog/post/pros-and-cons-of-responsive-web-design>

<http://mediumwell.com/pros-cons-adaptive-web-design-web-developers-need-know/>

# Test Plan

## Introduction

This testing plan is the documentation for Assessment Task 2 project for Rapid Application Documentation. In this project, we develop a web site that will show movies information. The website will show movies information including ID, title, studio, status, sound, versions, rec ret price, rating, year, genre, and aspect of the movie. In this website, the user also can search for the name, title, year, and rating and will show the top 10 most searched movies in the bar chart.

This testing plan documentation will explain about the testing that our group will develop in this project. Our strategies will be based on the quality assurance lifecycle at CITE Managed Services including initiation and planning, first review, iteration audits, final verification and validation. The methodology that we will use is rapid application development. The plan identify the items to be tested, the features to be tested, the types of testing to be performed, the personel reponsible for testing, the resources and schedule required to complete testing, and the risks associated with the plan.

### Scope

**In Scope**

The features that we will test are the size of the website and test it on various sizes. The website will be opened on various sizes should be able to responds to or resizes itself depending on the type of device it is being seen through. The website working correctly including the website can be searched using title, rating, year, and genre, the website showing the top 10 most searched movies in bar chart and showing all the movies information correctly.

**Out of Scope**

The features that will not be tested are the website security and performance, high quality website, high performance because they are not included in the software requirements

### Quality Objective

The test objectives are to verify the functionality of the movie database website for our group (team D), the project should focus on testing the responsive functionality that will responds to or resizes itself depending on the type of device it is being seen through and to guarantee all the requirements working correctly as it should be. The other objectives of our testing project should be able to meet all the functional and non-functional requirements, ensure the application under tets meets the quiality specifiations definied by the client, ensure all the bugs or issues are identified and fixed before go live, and ensure all the code meet all the quality assurance standard of CITE.

### Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| No. | Member | Tasks |
|  | Testers (Christina T, Kyle C, Jason K) | Test the software at different levels and record the results, develop the test specifications, conduct the test and record the results for different items of the software |
|  | Test manager (Jason K) | Prepares test strategy, define test levels and test cycles, coordinate with other teams for software releases, gather the updates on testing progress |
|  | Test designer (Christina T) | The person responsible for creating the test scripts, scenarios, test lives, and so on that make the tests to be performed. |
|  | Test approver (Kyle C) | The person that responsible for reviewing, validating, and approving that test materials created by the test designer. |
|  | Reviewer | The person responsible for reviewing reports from the testers and determining what subsequent actions will be taken. |

## 

## Test Methodology

### Overview

The test methodology for the project we will use is rapid application development. We are using the rapid application development because rapid application development can be used with minimal planning in favor of rapid prototyping. Rapid application development was conceived for this purpose – to develop prototypes rapidly for testing functions and features, without having to worry about how the end product will be affected.

The reasons why we choose rapid application development also because if there are changes of requirements then we can change it any time. This methodologies also focuses on speed, compared to other models which usually focus on bringing a working product to the customer.

### Test Levels

The test levels or test types we will be conducting in our web application under testing are UI testing, grey-box testing, rule based testing and database testing which falls all under RAD methodology testing.

The RAD testing methodology has lots of testing as it requires a test every sprint to ensure that the application is working as expected. We will test the database first if it connects, SQL statements works correctly and running database tests just like we would run our applications.

We would run the grey box testing by testing the web application with a mix of black box testing (testing the UI) and white box testing (testing the source code) to search for defects.

### Bug Triage

Our bug triage will have a process of each bug is prioritized based on its severity and frequrency in our movie application. We will evaluate the bug next and think of a resolution of the defects. We notify our team members on the bugs that occurred on the program and how we fix it. Generally we try to fix the bugs during development of the application.

### Suspension Criteria and Resumption Requirements

Our suspension criteria mainly depend on the working days that we can do the testing procedures such as holidays or weekends. A critical defect is on suspension criterion that cannot allow further testing until the defect is treated and out of the way.

Resumption requirements of the testing procedure may only happen when the holiday period ends or it is a working day, or the fix is successfully implemented and working again making it available for the testing team to continue the testing procedures.

### Test Completeness

The testing criteria will deem the testing complete if the bugs that we found are fixed and all test cases are followed and executed. After testing we work on the code again to fix all the bugs that were found this mainly determines if the test is deemed to be achieved/completed.

## Test Deliverables

The test artifacts that will be delivered during the different testing lifecycles are:

* Test Plan
* Test Documents/Test Cases
* Bug Reports

|  |  |
| --- | --- |
| Test Plan (this document) | the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements. |
| Test Document | documentation of artifacts created before or during the testing of software. It helps the testing team to estimate testing effort needed, test coverage, resource tracking, execution progress, etc. It is a complete suite of documents that allows you to describe and document test planning, test design, test execution, test results that are drawn from the testing activity. |
| Bug Report | Reports that contains information needed to reproduce and fix problems. |

## Resource & Environment Needs

### Testing Tools

Testing tools used to test the application:

* XAMPP Control Panel
* PhpMyAdmin
* MySQL
* PHP, CSS, HTML
* Office 365 mainly Microsoft Word
* Notepad++ or Adobe Dreamweaver as a source code editor

### Test Environment

Minimum hardware and software requirements for test environment

1. OS: Microsoft Windows 7
2. Processor: Intel Core i5-4590 CPU @3.30GHz Quad core
3. storage that has a free space of greater than 1MBs
4. RAM: greater than 2GBs
5. Separate GPU is not needed for this game as not much graphics processing is needed and some CPUs already integrate graphics the processor.
6. Hard Drive disk or SSD with a free space of at least 20Mbs.

# Testing Document

## Scope

CITE Managed Services develops software for a range of clients and from time to time they will utilise external contractors when appropriate. Team D (us) have been employed to implement a Movie Database for Acme Entertainment Pty Ltd.

It is required that the website is reworked and updated so it is responsive meaning the layout of the webpages depending on the screen size.

## Testing Case

Testing Case for database searching

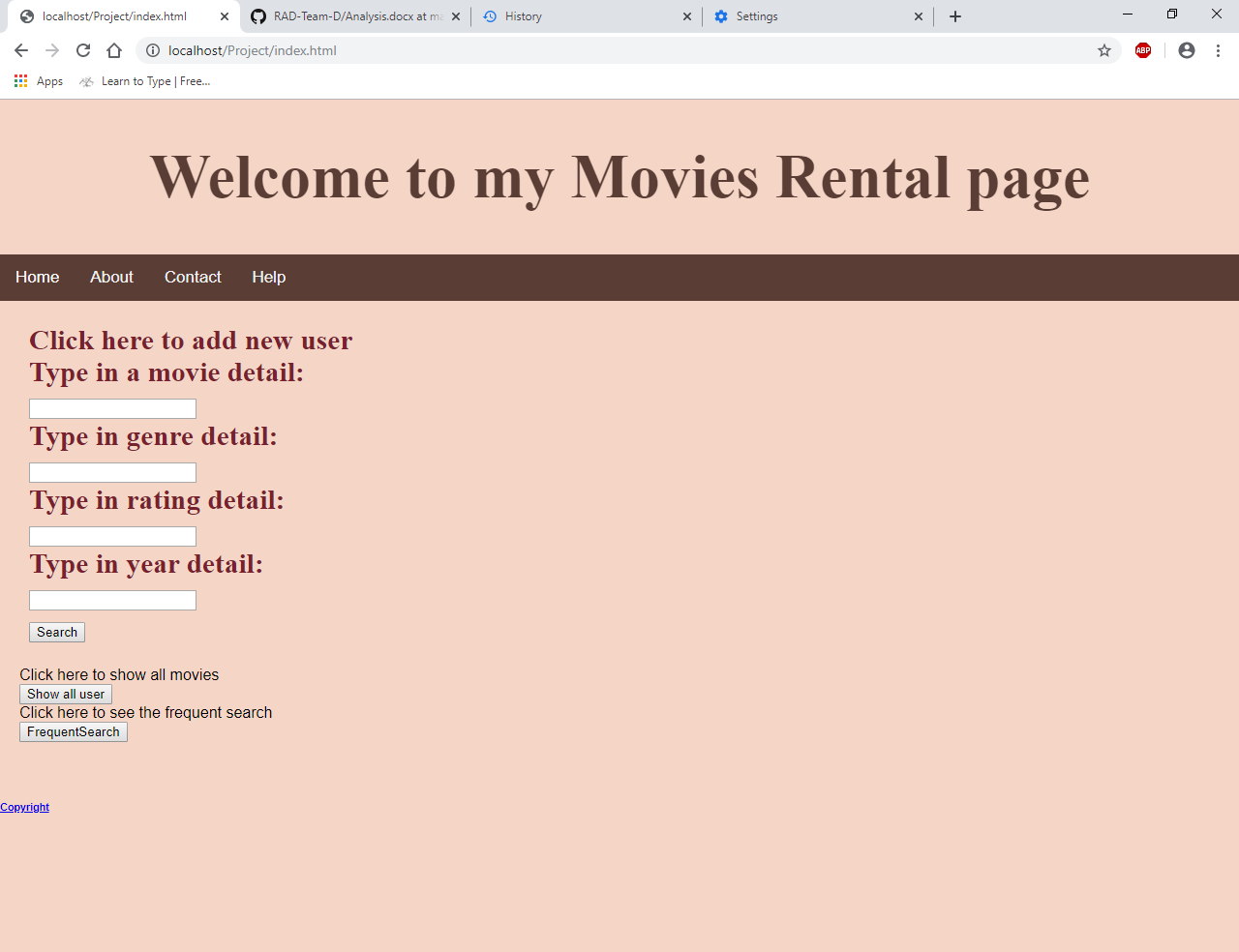
|  |  |  |
| --- | --- | --- |
| Case | Description | Screenshot |
| 1 | “Fast” entered into movie detail box returns only movie titles that include “Fast” |  |
| 2 | “Comedy” entered genre detail box returns only comedy movies. |  |
| 3 | “1995” entered into the year detail box returns only movies that were released in “1995”. |  |
| 4 | “NR” was entered into the rating detail box and returned only movies with a rating of NR. |  |

## Responsive Layout

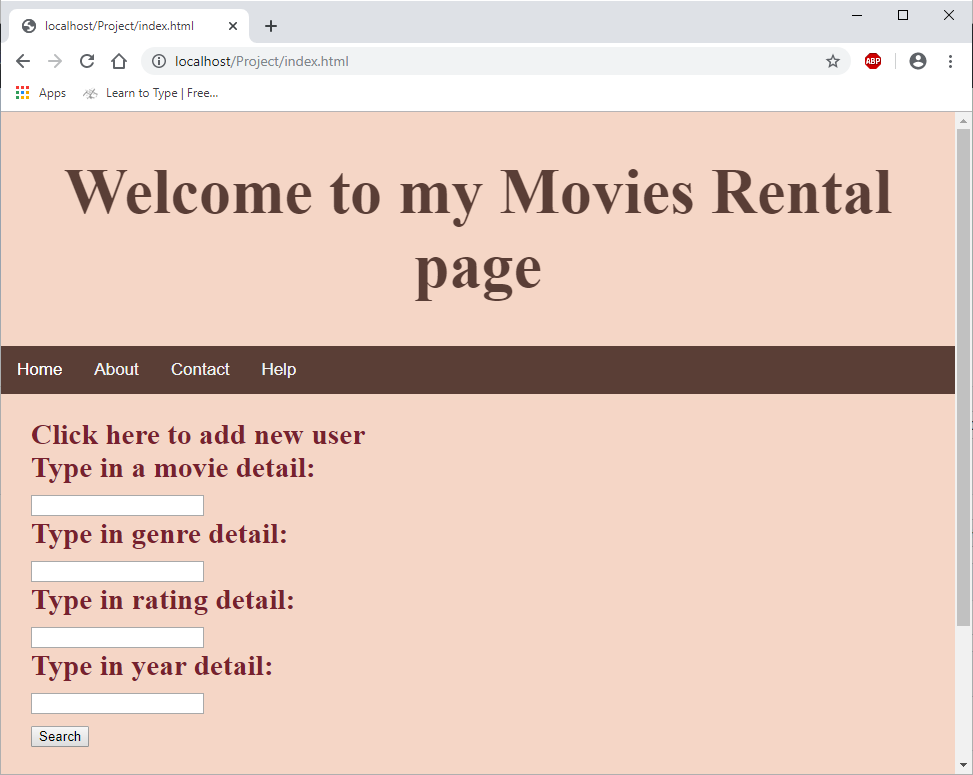
With the responsive layout it won’t necessarily look like the tablet version if the tablet screen is small. If your mobile screen is huge than it will scale around your device screen etc. The database connection will work the same as long as you have an internet connection queries will be sent through.

## Home Page

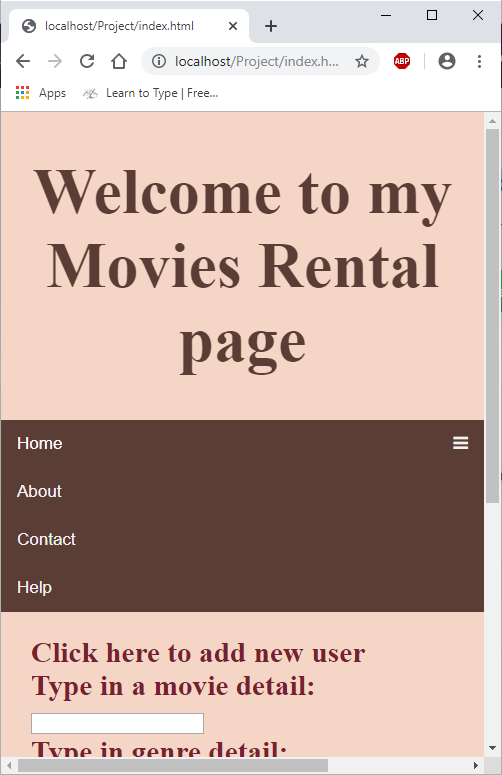
### Desktop



### Tablet

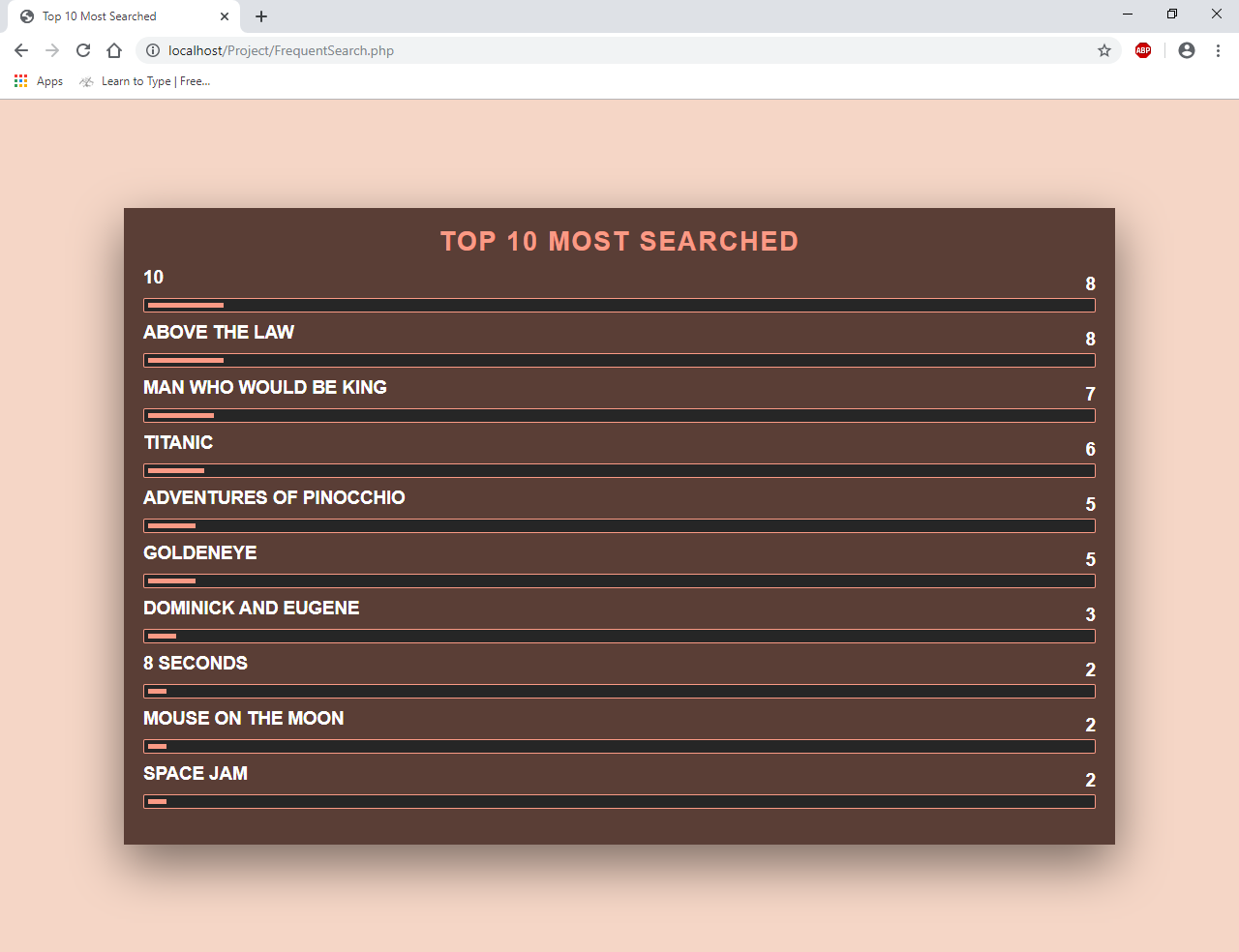


### Mobile



## Frequent Search Chart

### Desktop



# **Sprint Two**

# Software Review Plan

## Purpose and objective

This software review plan is a plan to review the software for users, management and peer reviews. CITE Managed Services develops software for a range of clients and from time to time they will utilise external contractors when appropriate. We as Team D have been employed to implement a Movie Database for Acme Entertainment Pty Ltd.

## Software Review Criteria

|  |  |  |
| --- | --- | --- |
| Review Detail | Check  list | Comments  (e.g. feedback on design and requirements) |
| Application can search for a lot of variation of specific search e.g. can search multiple headings with different variations and not only one.  (example) |  | Application can only search one variation (only movie title, genre type, rating and year)  (example) |
| Web main page loads successfully with no errors |  |  |
| The Movie Database Application functions correctly on multiple platforms |  |  |
| Software Includes a Help feature from help feature button and displays a helpful descriptions |  |  |
| Has good loading time of webpage |  |  |
| Movie database can be seen from the “show all movies” button. |  |  |
| The movie database application has a frequently searched term page |  |  |
| Application includes a search feature that can search for movies by any heading e.g. title, name, aspect ratio, rating etc. |  |  |
| Members can subscribe to application on monthly week letters |  |  |

Additional Comments

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Performance Report

## Code Optimisers

Code optimization is any method of code modification to improve code quality 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.

## Types and Levels of Optimization

Optimization can be performed by automatic optimizers, or programmers. An optimizer is either a specialized software tool or a built-in unit of a compiler (the so-called optimizing compiler). Modern processors can also optimize the execution order of code instructions.

Optimizations are classified into high-level and low-level optimizations. High-level optimizations are usually performed by the programmer, who handles abstract entities (functions, procedures, classes, etc.) and keeps in mind the general framework of the task to optimize the design of a system. Optimizations performed at the level of elementary structural blocks of source code – loops and branches.

## What to Optimize

With manual code optimization, one faces another problem: one doesn't just need to know how exactly optimization should be done, but also what particular part of the program should be optimized. Due to various reasons (slow input operations, the difference in the working speed of a human operator and a computer, and so on), 90% of the execution time of a program is spent executing only 10% of the code since optimization takes additional time aside from the time you've spent on developing the program, you'd better focus on optimizing this time-critical 10% of code rather than try to optimize the whole program. These code fragments are known as bottlenecks, and can be detected by special utilities - profilers - which can measure the time taken by various parts of the program to execute.

## Good and Bad Outcomes of Optimization

In programming, almost everything should be treated from the viewpoint of rationality - optimization is no exception. There is a belief that code written by an inexperienced Assembler programmer is 3-5 times slower than code generated by the compiler (Zubkov). Widely known is a phrase by Knuth regarding early low-level optimizations (such as attempts to save on operators or variables): "Premature optimization is the root of all evil".

Most programmers don't complain about optimizations performed by the optimizer, some of which are conventional and obligatory. Such as, for instance, tail call optimization in functional languages (tail call is a special case of recursion, which can be represented as a loop).

However, one should understand that multiple complex optimizations at the level of machine code may cause a great slow-down of compilation. The benefit they allow you to gain may be much too insignificant, when compared to general system design optimizations (Wirth). One should also keep in mind that modern languages, with all their syntactic and semantic "frills", have many nuances and subtleties, so that a programmer who isn't familiar with them may be surprised by an outcome of optimization.

## Performance Tools

### WebLOAD

Enterprise-grade load and performance testing tool for web applications. WebLOAD is the tool of choice for enterprises with heavy user load and complex testing requirements. It allows you to perform load and stress testing on any internet application by generating load from the cloud and on-premises machines.

WebLOAD’s strengths are its flexibility and ease of use – enabling you to quickly define the tests you need with features like DOM-based recording/playback, automatic correlation, and JavaScript scripting language.

The tool provides a clear analysis of your web application performance, pinpointing issues and bottlenecks that may stand in the way of achieving your load and response requirements.

WebLOAD supports hundreds of technologies – from web protocols to enterprise applications and has built-in integration with Jenkins, Selenium and many other tools to enable continuous load testing for DevOps.

### SmartMeter.io

This load and performance testing tool provides advanced testing functions. With JMeter at its core, it will be instantly familiar to any of its users.

Creating a test in SmartMeter.io is very simple. You can make test scenarios without scripting just by clicking in an embedded browser. There’s also no proxy setup or browser plugin necessary.

It features automatically generated reports with all details about the test and its results. The results contain auto-evaluated acceptance criteria, statistics, graph comparison tool, and trend analysis of multiple test runs.

The tool is also strong in distributed testing, CI integration, and offers unparalleled performance testing support for Vaadin apps.

## Apache JMeter

**Open source load testing tool:** It is a Java platform application. It is mainly considered as a performance testing tool and it can also be integrated with the test plan. In addition to the load Test plan, you can also create a functional test plan. This tool has the capacity to be loaded into a server or network so as to check on its performance and analyse its working under different conditions.

Initially, it was introduced for testing the web applications, but later its scope had widened. It is of great use in testing the functional performance of the resources such as Servlets, Perl Scripts and JAVA objects. Need JVM 1.4 or higher to run.

# Test Plan

## Introduction

This testing plan is the documentation for Assessment Task 2 project for Rapid Application Documentation. In this project, we develop a web site that will show movies information. The website will show movies information including ID, title, studio, status, sound, versions, rec ret price, rating, year, genre, and aspect of the movie. In this website, the user also can search for the name, title, year, and rating and will show the top 10 most searched movies in the bar chart.

This testing plan documentation will explain about the testing that our group will develop in this project. Our strategies will be based on the quality assurance lifecycle at CITE Managed Services including initiation and planning, first review, iteration audits, final verification and validation. The methodology that we will use is rapid application development. The plan identify the items to be tested, the features to be tested, the types of testing to be performed, the personel reponsible for testing, the resources and schedule required to complete testing, and the risks associated with the plan.

### Scope

**In Scope**

The website will be opened on various sizes should be able to responds to or resizes itself depending on the type of device it is being seen through. The website working correctly including the website can be searched using title, rating, year, and genre, the website showing the top 10 most searched movies in bar chart, and showing all the movies information correctly. The features that will be mainly used to test if the administrator portal works and when signed in as an admin, it has administrative functions.

**Out of Scope**

The features that will not be tested are the website security and performance, high quality website, high performance because they are not included in the software requirements

### Quality Objective

The test objectives are to verify the functionality of the movie database website for our group (team D), the project should focus on testing the responsive functionality that will responds to or resizes itself depending on the type of device it is being seen through and to guarantee all the requirements working correctly as it should be. The other objectives of our testing project should be able to meet all the functional and non-functional requirements, ensure the application under tets meets the quiality specifiations definied by the client, ensure all the bugs or issues are identified and fixed before go live, and ensure all the code meet all the quality assurance standard of CITE.

### Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| No. | Member | Tasks |
|  | Testers (Christina T, Kyle C, Jason K) | Test the software at different levels and record the results, develop the test specifications, conduct the test and record the results for different items of the software |
|  | Test manager (Jason K) | Prepares test strategy, define test levels and test cycles, coordinate with other teams for software releases, gather the updates on testing progress |
|  | Test designer (Christina T) | The person responsible for creating the test scripts, scenarios, test lives, and so on that make the tests to be performed. |
|  | Test approver (Kyle C) | The person that responsible for reviewing, validating, and approving that test materials created by the test designer. |
|  | Reviewer | The person responsible for reviewing reports from the testers and determining what subsequent actions will be taken. |

## Test Methodology

### Overview

The test methodology for the project we will use is rapid application development. We are using the rapid application development because rapid application development can be used with minimal planning in favor of rapid prototyping. Rapid application development was conceived for this purpose – to develop prototypes rapidly for testing functions and features, without having to worry about how the end product will be affected.

The reasons why we choose rapid application development also because if there are changes of requirements then we can change it any time. This methodologies also focuses on speed, compared to other models which usually focus on bringing a working product to the customer.

### Test Levels

The test levels or test types we will be conducting in our web application under testing are UI testing, grey-box testing, rule based testing and database testing which falls all under RAD methodology testing.

The RAD testing methodology has lots of testing as it requires a test every sprint to ensure that the application is working as expected. We will test the database first if it connects, SQL statements works correctly and running database tests just like we would run our applications.

We would run the grey box testing by testing the web application with a mix of black box testing (testing the UI) and white box testing (testing the source code) to search for defects.

### Bug Triage

Our bug triage will have a process of each bug is prioritized based on its severity and frequrency in our movie application. We will evaluate the bug next and think of a resolution of the defects. We notify our team members on the bugs that occurred on the program and how we fix it. Generally we try to fix the bugs during development of the application.

### Suspension Criteria and Resumption Requirements

Our suspension criteria mainly depend on the working days that we can do the testing procedures such as holidays or weekends. A critical defect is on suspension criterion that cannot allow further testing until the defect is treated and out of the way.

Resumption requirements of the testing procedure may only happen when the holiday period ends or it is a working day, or the fix is successfully implemented and working again making it available for the testing team to continue the testing procedures.

### Test Completeness

The testing criteria will deem the testing complete if the bugs that we found are fixed and all test cases are followed and executed. After testing we work on the code again to fix all the bugs that were found this mainly determines if the test is deemed to be achieved/completed.

## Test Deliverables

The test artifacts that will be delivered during the different testing lifecycles are:

* Test Plan
* Test Documents/Test Cases
* Bug Reports

|  |  |
| --- | --- |
| Test Plan (this document) | the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements. |
| Test Document | documentation of artifacts created before or during the testing of software. It helps the testing team to estimate testing effort needed, test coverage, resource tracking, execution progress, etc. It is a complete suite of documents that allows you to describe and document test planning, test design, test execution, test results that are drawn from the testing activity. |
| Bug Report | Reports that contains information needed to reproduce and fix problems. |

## Resource & Environment Needs

### Testing Tools

Testing tools used to test the application:

* XAMPP Control Panel
* PhpMyAdmin
* Apache
* MySQL
* PHP, CSS, HTML
* Office 365 mainly Microsoft Word
* Notepad++ or Adobe Dreamweaver as a source code editor

### Test Environment

Minimum hardware and software requirements for test environment

1. OS: Microsoft Windows 7
2. Processor: Intel Core i5-4590 CPU @3.30GHz Quad core
3. storage that has a free space of greater than 1MBs
4. RAM: greater than 2GBs
5. Separate GPU is not needed for this game as not much graphics processing is needed and some CPUs already integrate graphics the processor.
6. Hard Drive disk or SSD with a free space of at least 20Mbs.

# Testing Document (Sprint Two)

## Scope

CITE Managed Services develops software for a range of clients and from time to time they will utilise external contractors when appropriate. Team D (us) have been employed to implement a Movie Database for Acme Entertainment Pty Ltd.

The requirements have been updated to create a membership page to add new records to a membership database, to show all current user, and to remove the records from membership database. This version will add the user full name and email address and can remove the user using just the email address. All the information also have validation functions that will filter out incorrect or erroneous information.

## Changelog

|  |  |  |
| --- | --- | --- |
| Version | Team Member | Description |
| 2.1 | Jason | Header changed to be responsive |
| 2.2 | Jason | Body textboxes and information formatted to fit the responsive layout |
| 2.3 | Jason | Fixed header webpage location links. |
| 2.4 | Jason | Added resolutions for devices and device requirements |
| 3.1 | Christina | Create a membership page and administrative functions. |

## Testing Case

|  |  |  |
| --- | --- | --- |
| Case | Description | Screenshot |
|  | Enter a valid information. (First Name: Christina, Last Name: Kartika, Email : christina2820@gmail.com) |  |
|  | Enter an invalid information. (First Name: Christina211, Last Name: Tatang, Email: christina@gmail.com) |  |
|  | Enter an empty textbox |  |
|  | Enter a user that already exists |  |
|  | Show all user |  |
|  | Remove a user using a valid information (Email: christina2820@gmail.com) |  |
|  | Remove a user using an invalid information (Email: christina@gmail.com) |  |
|  | Log in as admin using a valid information (Username: TeamD, Password: TeamD123) |  |

# **Sprint Three**

# Test Plan (Sprint Three)

## Introduction

This testing plan is the documentation for Assessment Task 2 project for Rapid Application Development. In this project, we develop a web site that will show movies information. The website will show movies information including ID, title, studio, status, sound, versions, rec ret price, rating, year, genre, and aspect of the movie. In this website, the user also can search for the name, title, year, and rating and will show the top 10 most searched movies in the bar chart.

This testing plan documentation will explain about the testing that our group will develop in this project. Our strategies will be based on the quality assurance lifecycle at CITE Managed Services including initiation and planning, first review, iteration audits, final verification and validation. The methodology that we will use is rapid application development. The plan identify the items to be tested, the features to be tested, the types of testing to be performed, the personel reponsible for testing, the resources and schedule required to complete testing, and the risks associated with the plan.

### Scope

**In Scope**

The website will be opened on various sizes using Google’s inspect element toggle device toolbar. It should be able to responds to or resizes itself depending on the type of device it is being seen through. The website will be including the website can be searched using title, rating, year, and genre, the website showing the top 10 most searched movies in bar chart, and showing all the movies information correctly. The features for sprint two that will be mainly used to test if the administrator portal works and when signed in as an admin, it has administrative functions. In addition for the Sprint three features we will test the top 10 information if the rating is responsive to the changes of rating, correctly using the administrator password, sign-in with different users.

**Out of Scope**

The features that will not be tested are the website security and performance, high quality website, high performance because they are not included in the software requirements

### Quality Objective

The test objectives are to verify the functionality of the movie database website for our group (team D), the project should focus on testing all the requirements from sprint 1-3.The other objectives of our testing project should be able to meet all the functional and non-functional requirements, ensure the application under tets meets the quiality specifiations definied by the client, ensure all the bugs or issues are identified and fixed before go live, and ensure all the code meet all the quality assurance standard of CITE.

### Roles and Responsibilities

|  |  |  |
| --- | --- | --- |
| No. | Member | Tasks |
|  | Testers (Christina T, Kyle C, Jason K) | Test the software at different levels and record the results, develop the test specifications, conduct the test and record the results for different items of the software |
|  | Test manager (Jason K) | Prepares test strategy, define test levels and test cycles, coordinate with other teams for software releases, gather the updates on testing progress |
|  | Test designer (Christina T) | The person responsible for creating the test scripts, scenarios, test lives, and so on that make the tests to be performed. |
|  | Test approver (Kyle C) | The person that responsible for reviewing, validating, and approving that test materials created by the test designer. |
|  | Reviewer | The person responsible for reviewing reports from the testers and determining what subsequent actions will be taken. |

## Test Methodology

### Overview

The test methodology for the project we will use is rapid application development. We are using the rapid application development because rapid application development can be used with minimal planning in favor of rapid prototyping. Rapid application development was conceived for this purpose – to develop prototypes rapidly for testing functions and features, without having to worry about how the end product will be affected.

The reasons why we choose rapid application development also because if there are changes of requirements then we can change it any time. This methodologies also focuses on speed, compared to other models which usually focus on bringing a working product to the customer.

### Test Levels

The test levels or test types we will be conducting in our web application under testing are UI testing, grey-box testing, rule based testing and database testing which falls all under RAD methodology testing.

The RAD testing methodology has lots of testing as it requires a test every sprint to ensure that the application is working as expected. We will test the database first if it connects, SQL statements works correctly and running database tests just like we would run our applications.

We would run the grey box testing by testing the web application with a mix of black box testing (testing the UI) and white box testing (testing the source code) to search for defects.

### Bug Triage

Our bug triage will have a process of each bug is prioritized based on its severity and frequrency in our movie application. We will evaluate the bug and think of a resolution of the defects. We notify our team members on the bugs that occurred on the program and how we fix it. Generally we try to fix the bugs during development of the application.

### Suspension Criteria and Resumption Requirements

Our suspension criteria mainly depend on the working days that we can do the testing procedures such as holidays or weekends. A critical defect is on suspension criterion that cannot allow further testing until the defect is treated and out of the way.

Resumption requirements of the testing procedure may only happen when the holiday period ends or it is a working day, or the fix is successfully implemented and working again making it available for the testing team to continue the testing procedures.

### Test Completeness

The testing criteria will deem the testing complete if the bugs that we found are fixed and all test cases are followed and executed. After testing we work on the code again to fix all the bugs that were found this mainly determines if the test is deemed to be achieved/completed.

## Test Deliverables

The test artifacts that will be delivered during the different testing lifecycles are:

* Test Plan
* Test Documents/Test Cases
* Bug Reports

|  |  |
| --- | --- |
| Test Plan (this document) | the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements. |
| Test Document | documentation of artifacts created before or during the testing of software. It helps the testing team to estimate testing effort needed, test coverage, resource tracking, execution progress, etc. It is a complete suite of documents that allows you to describe and document test planning, test design, test execution, test results that are drawn from the testing activity. |
| Bug Report | Reports that contains information needed to reproduce and fix problems. |

## Resource & Environment Needs

### Testing Tools

Testing tools used to test the application:

* XAMPP Control Panel
* PhpMyAdmin
* MySQL
* PHP, CSS, HTML
* Office 365 mainly Microsoft Word
* Notepad++ or Adobe Dreamweaver as a source code editor

### Test Environment

Minimum hardware and software requirements for test environment

1. OS: Microsoft Windows 7
2. Processor: Intel Core i5-4590 CPU @3.30GHz Quad core
3. storage that has a free space of greater than 1MBs
4. RAM: greater than 2GBs
5. Separate GPU is not needed for this game as not much graphics processing is needed and some CPUs already integrate graphics the processor.
6. Hard Drive disk or SSD with a free space of at least 20Mbs.

# Software Review Plan (Sprint Three)

## Purpose and objective

This software review plan is a plan to review the software for users, management and peer reviews. CITE Managed Services develops software for a range of clients and from time to time they will utilise external contractors when appropriate. We as Team D have been employed to implement a Movie Database for Acme Entertainment Pty Ltd.

## Software Review Criteria

|  |  |  |
| --- | --- | --- |
| Review Detail | Check  list | Comments  (e.g. feedback on design and requirements) |
| Application can search for a lot of variation of specific search e.g. can search multiple headings with different variations and not only one.  (example) |  | Rating: 6/10  Application can only search one variation (only movie title, genre type, rating and year)  (example) |
| Web main page loads successfully with no errors |  | Rating: |
| The Movie Database Application functions correctly on multiple platforms |  | Rating: |
| Software Includes a Help feature from help feature button and displays a helpful descriptions |  | Rating: |
| Has good loading time of webpage |  | Rating: |
| Movie database can be seen from the “show all movies” button. |  | Rating: |
| The movie database application has a frequently searched term page |  | Rating: |
| Application includes a search feature that can search for movies by any heading e.g. title, name, aspect ratio, rating etc. |  | Rating: |
| Members can subscribe to application on monthly week letters |  | Rating: |
| Members can rate a movie from 1-5 |  | Rating: |
| Shows the top 10 rated movies and top 10 searched movies |  | Rating: |
| All pages are responsive and response to the user’s resolution |  | Rating: |

Additional Comments

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Optimisation report

## Introduction

The purpose of this document is to optimise the movie database application by analysing the webpages of the application. Code optimization is any method of code modification to improve code quality 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.

Optimization can be performed by automatic optimizers, or programmers. An optimizer is either a specialized software tool or a built-in unit of a compiler (the so-called optimizing compiler). Modern processors can also optimize the execution order of code instructions

## Performance tool used

### PHP\_CodeSniffer and PHP Beautifier

PHP\_CodeSniffer is a set of two PHP scripts; the main phpcs script that tokenizes PHP, JavaScript and CSS files to detect violations of a defined coding standard, and a second phpcbf script to automatically correct coding standard violations.

PHP Beautifier gives the PHP code proper indentation with spaces or tabs. It also supports various indentation styles such as K&R style, Allman style, Whitesmiths style and GNU style.

### WebLOAD

Enterprise-grade load and performance testing tool for web applications. WebLOAD is the tool of choice for enterprises with heavy user load and complex testing requirements. It allows you to perform load and stress testing on any internet application by generating load from the cloud and on-premises machines. <https://www.youtube.com/watch?v=nffOrOtVvT0&feature=share>

## WebLOAD Summary

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Start** |  | 11/27/19 11:48 PM |  | **End** |  | 11/28/19 12:29 AM |  | **Duration** |  | 00:41:21 |

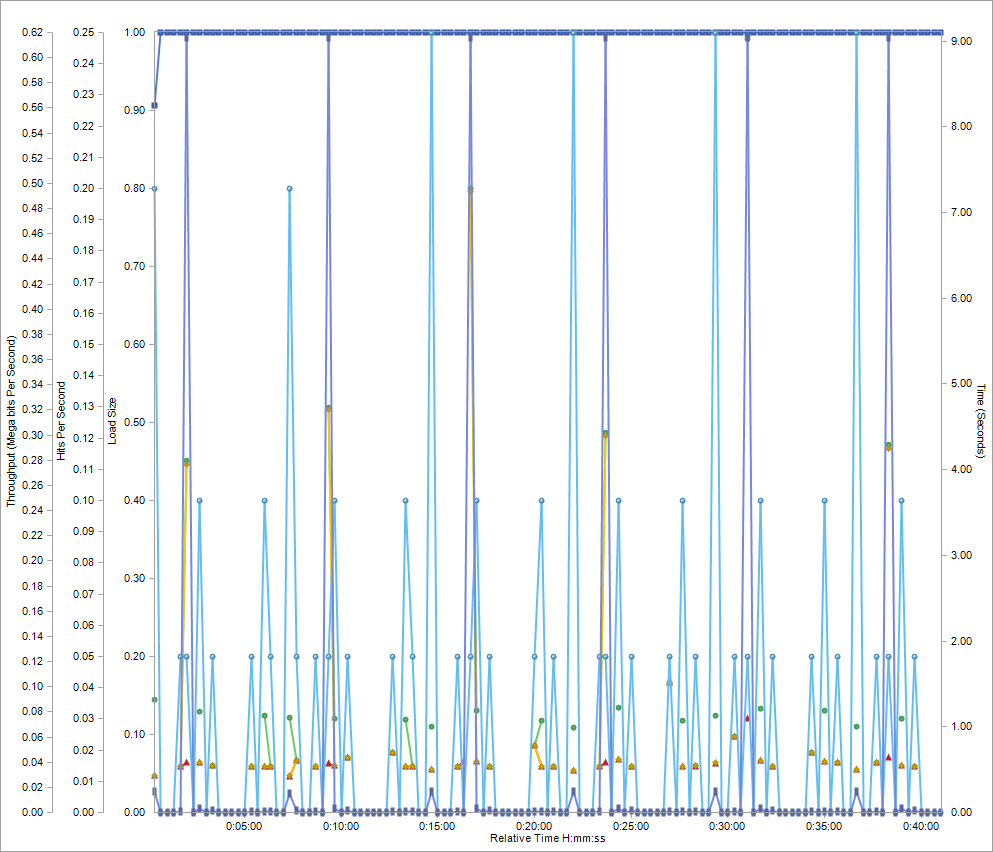
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test passed** | **Max Virtual Clients** | 1 | **Total Throughput (MB)** | 3.84 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total Errors** |  | 0 |  | **Total Warnings** |  | 0 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Failed Rounds** |  | 0 |  | **Total Rounds** |  | 5 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total Failed Hits** |  | 0 |  | **Total Hits** |  | 79 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Total Failed TXs** |  | 0 |  | **Total TXs** |  | 38 |  |  |



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | **Measurement Name** | **Min** | **Max** | **Average** | **StDev** |  |
|  |  |  | Load Size | 0.907 | 1 | 0.999 | 0.008 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  | Page Time | 0.533 | 9.057 | 1.468 | 1.775 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  | Time To First Byte | 0.425 | 1.514 | 0.602 | 0.174 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  | Response Time | 0.428 | 9.02 | 1.247 | 1.826 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  | Hits Per Second | 0 | 0.25 | 0.032 | 0.056 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  | Throughput (Mega bits Per Second) | 0 | 0.616 | 0.031 | 0.133 |  |
|  |  |  |  |
|  |  |  |  |

This chart displays the main performance indicators and changes in Load Size, over time:

- Load Size – The number of Virtual Clients running during the last reporting interval.

- Page Time – The time it takes to complete a successful upper level request, in seconds.

- Time to First Byte – The time it takes from when a request is sent until the Virtual Client receives the first byte of data.

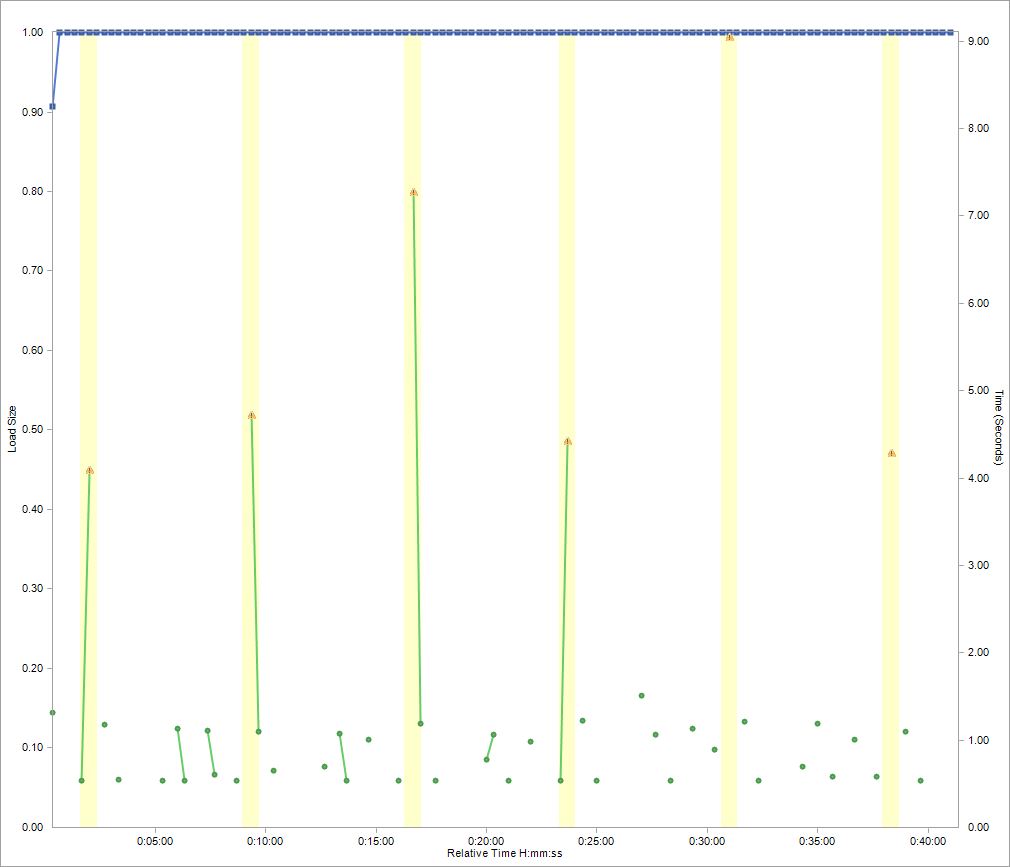
- Response Time –The time it takes the SUT to send the object of an HTTP request back to a Virtual Client, in seconds.

- Hits Per Second – The number of times the Virtual Clients made an HTTP request, divided by the elapsed time.

- Throughput – The average number of Mega bits per second, transmitted to the Virtual Clients running the Agenda.

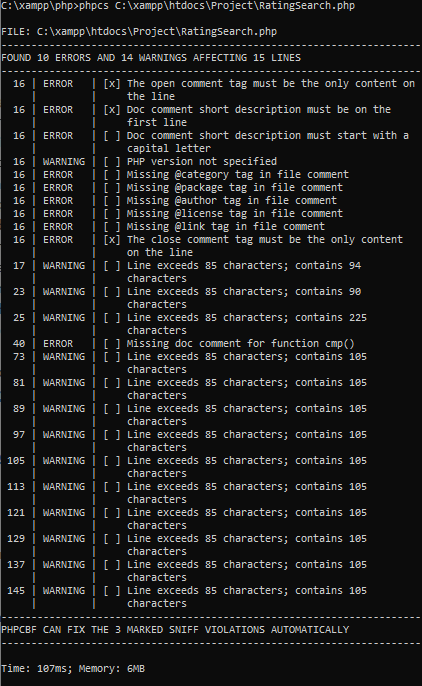
**Slowest Transactions**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Transaction Name** | **Transaction Time** | | | | | | |
| **Count** | **Avg.** | **Percentile 90.0%** | **Max** | **Min** | **Std Dev** | **TPS** |
| Transaction Add User | 5 | 126.422 | 127.091 | 127.091 | 126.113 | 0.344 | 0.002 |
|  |  |
| Transaction Remove User | 5 | 57.435 | 57.470 | 57.470 | 57.358 | 0.040 | 0.002 |
|  |  |
| Transaction Log In Admin | 5 | 39.423 | 39.508 | 39.508 | 39.383 | 0.048 | 0.002 |
|  |  |
| Transaction Frequent Search | 6 | 32.603 | 32.665 | 32.665 | 32.531 | 0.052 | 0.003 |
|  |  |
| Transaction Show All Movies | 6 | 31.640 | 35.050 | 35.050 | 30.097 | 1.865 | 0.003 |
|  |  |

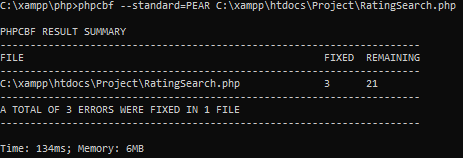


## PHP\_CodeSniffer and PHP Code beautifier summary

Phpcs on ratingSearch.php



Phpcbf to code proper indentation and spaces. Note that the remaining errors (21) is not required for indentation and spaces and is only required for comments documenting such as specifying tags in file.



A better example of this phpcbf having an effect is fixing 163 errors out of 169 in the AddUser.php below

|  |  |  |
| --- | --- | --- |
| File | PHP\_CodeSniffer | PHP Beautifier |
| AddUser.php |  |  |
| AdminPanel.php |  |  |
| AdminPanelLoggedIn.php |  |  |
| clientPage.php |  |  |
| FrequentSearch.php |  |  |
| RemoveUser.php |  |  |
| Search.php |  |  |

PHP\_CodeSniffer and PHP Beautifier testing (**NOT** all tests are in the table, only major files)

## Optimisation conclusion

We used WebLOAD as a profiler for our movie database to analyse our webpages by the loading times of each pages and we use that statistics to improve our pages. A Table below shows the transaction time after we optimise our code.

For phpcs and phpcbf improved our overall maintainability of code by having the phpcbf to fix indentations and spaces which is a must for coding conventions to make code more readable. The phpcs just shows the errors and warnings of the violated code conventions while phpcbf fixes indentations and spaces. This also helps with the runtime of the php depending on how to programmer firstly wrote the code e.g. if the programmer writes a program as close or as one-line run-time will be faster than PHP Beautifier but reading and maintaining the code will be significantly harder.

# Testing Document

## Scope

CITE Managed Services develops software for a range of clients and from time to time they will utilise external contractors when appropriate. Team D (us) have been employed to implement a Movie Database for Acme Entertainment Pty Ltd.

It is required that the website has a client log in area, admin log in area and has a way to rate movies.

## Device Requirements

### Computers

* Windows 7, Windows 8, Windows 8.1, Windows 10 or later
* An Intel Pentium 4 processor or later that's SSE2 capable
* OS X Yosemite 10.10 or later
* 64-bit Ubuntu 14.04+, Debian 8+, openSUSE 13.3+, or Fedora Linux 24+

## Tablets and Mobiles

### Apple

* iPad, iPhone, and iPod Touch
* iOS 10 or later
* All languages supported by the App Store

### Android

* Chrome is available on phones and tablets running Android 4.4+ (KitKat).

## Testing Case

Testing Case for database searching

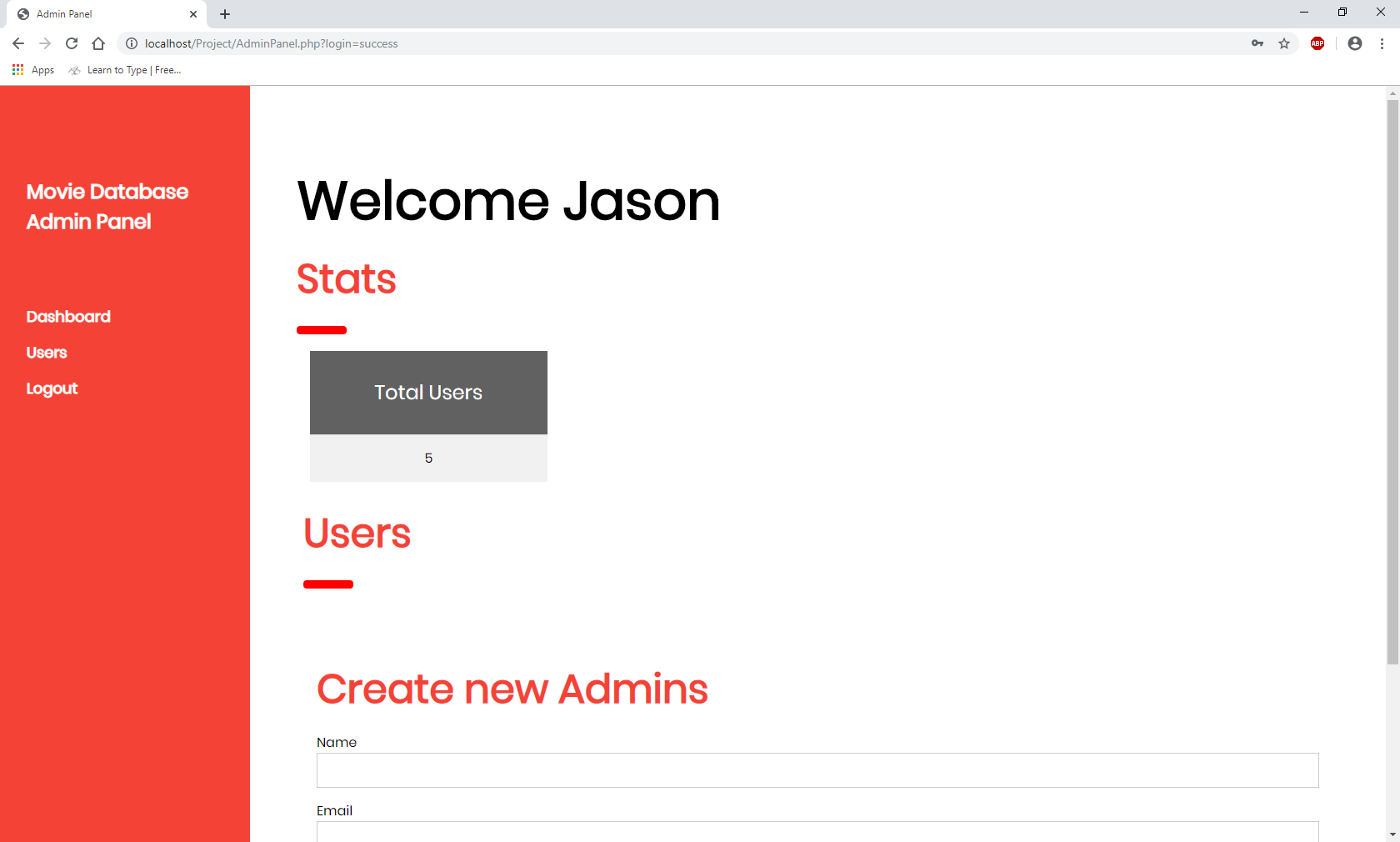
|  |  |  |
| --- | --- | --- |
| Case | Description | Screenshot |
| 1 | Admin can login to the admin panel |  |
| 2 | Admin can reset there password |  |
| 3 | Admin can create other admins |  |
| 4 | Client can login |  |
| 5 | Client can signup |  |
| 6 | Client can rate movies |  |
| 7 | Clients can like/dislike movies |  |
| 8 | Clients can see the top 10 most liked movies |  |
| 9 | Clients can see the likes and dislikes of the movies |  |

## Responsive Layout

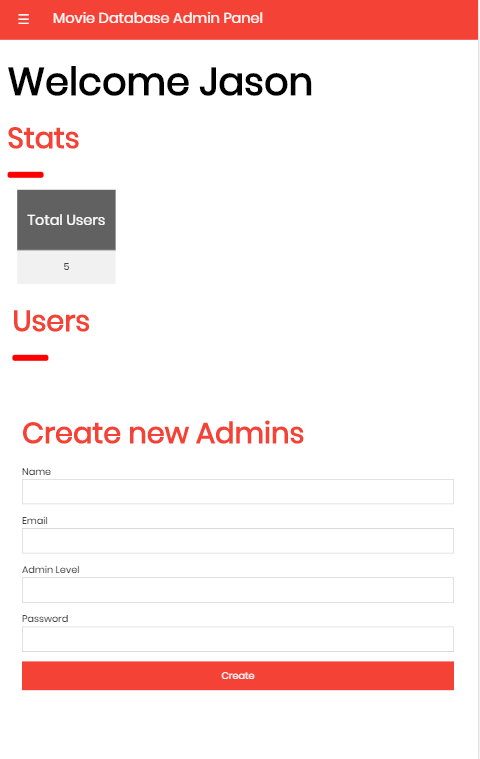
With the responsive layout it won’t necessarily look like the tablet version if the tablet screen is small. If your mobile screen is huge than it will scale around your device screen etc. The database connection will work the same as long as you have an internet connection queries will be sent through.

## Admin Page

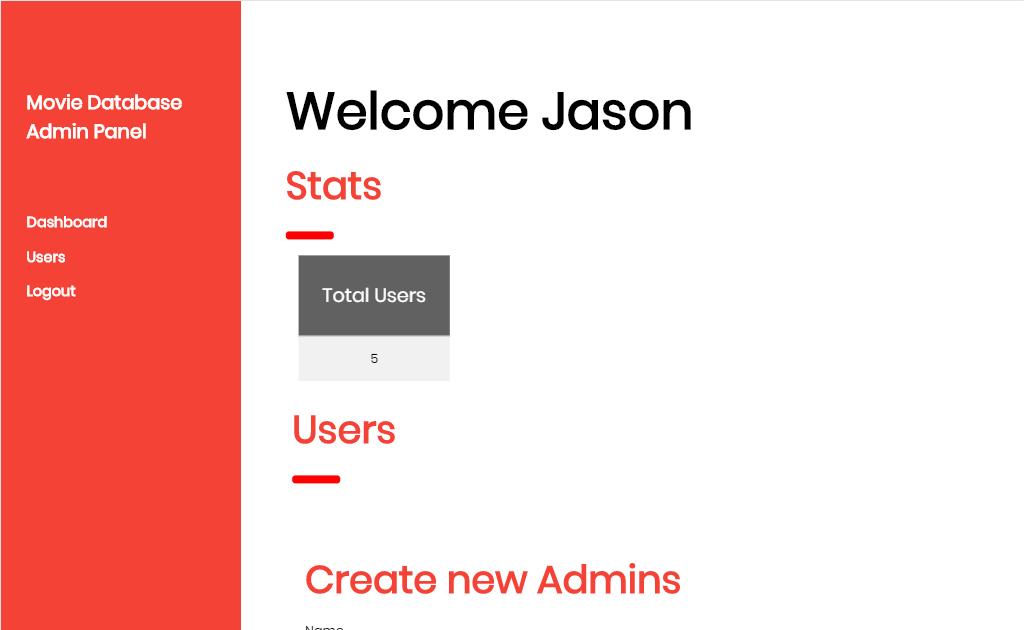
### Desktop 1920x1080



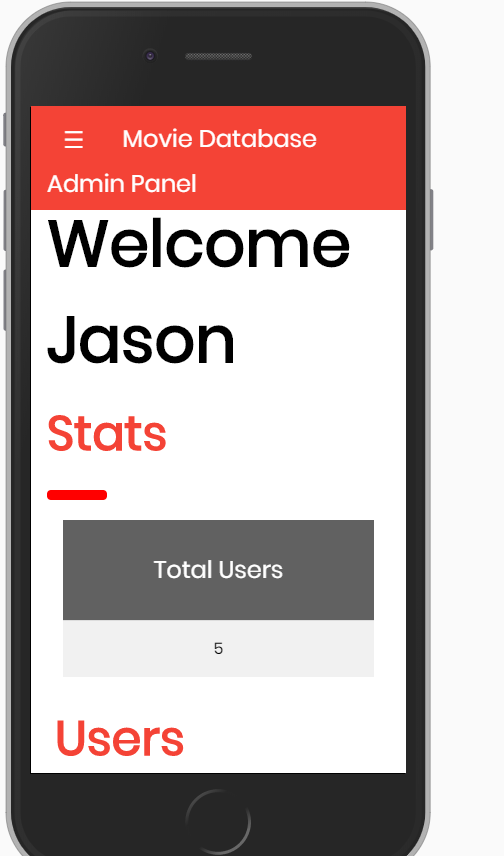
### Tablet 800 x 1280 Portrait



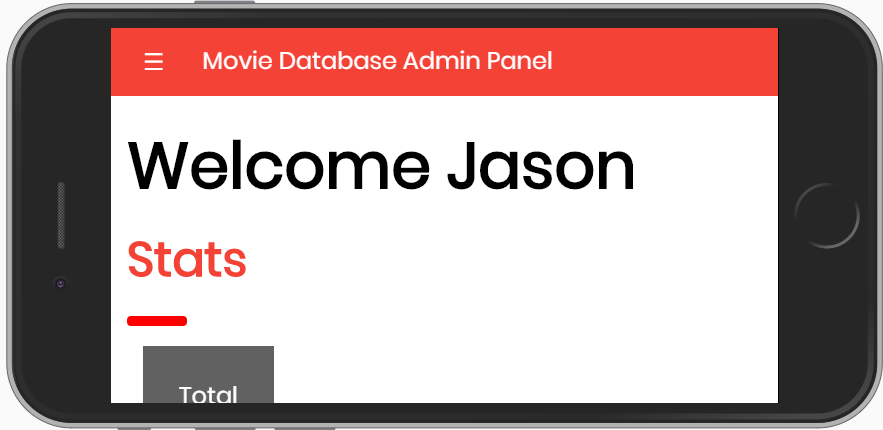
### Tablet 1280x800 Landscape



### Mobile 375x667 Portrait e.g. Iphone 6

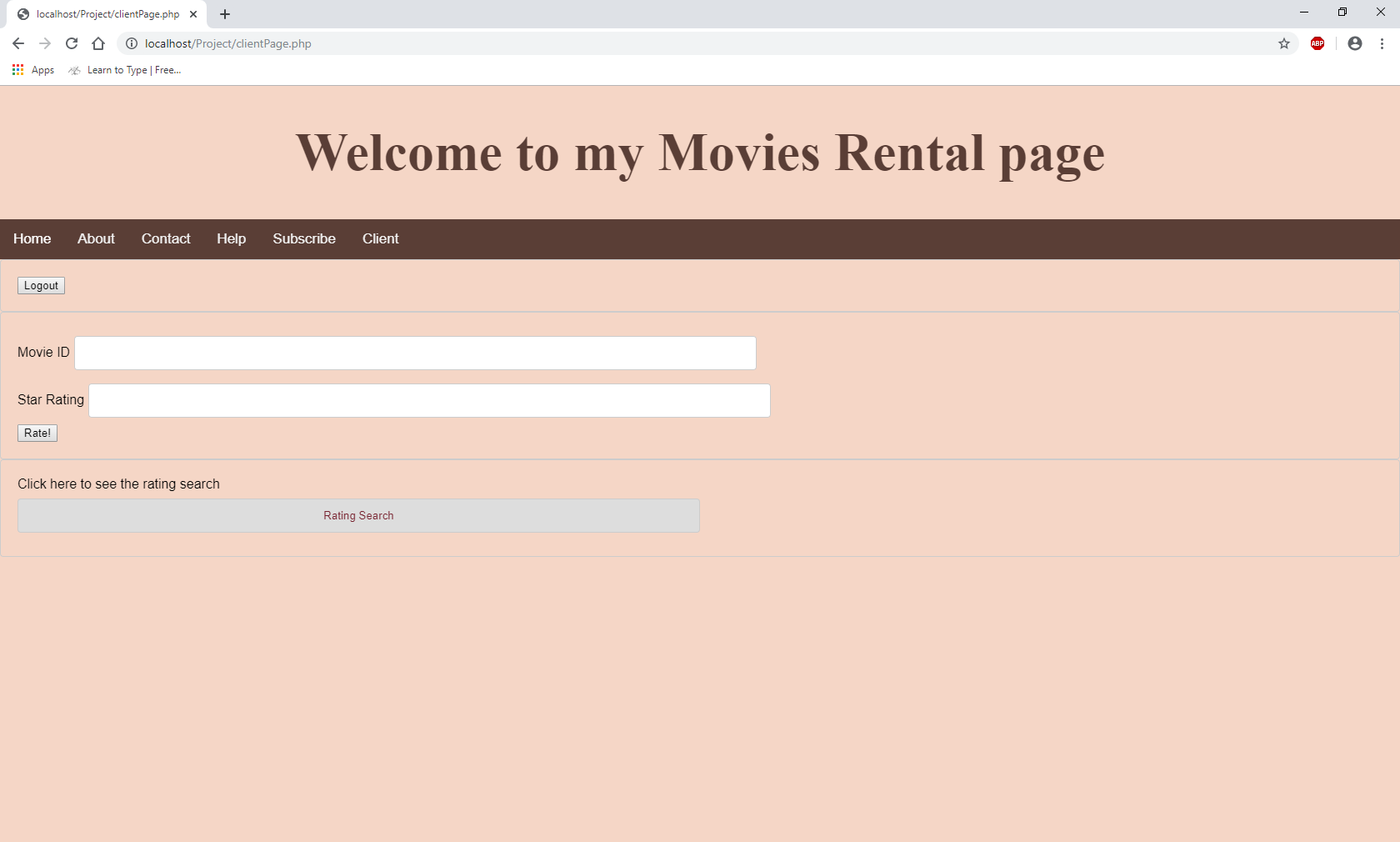


### Mobile 667x375 Landscape e.g. Iphone 6

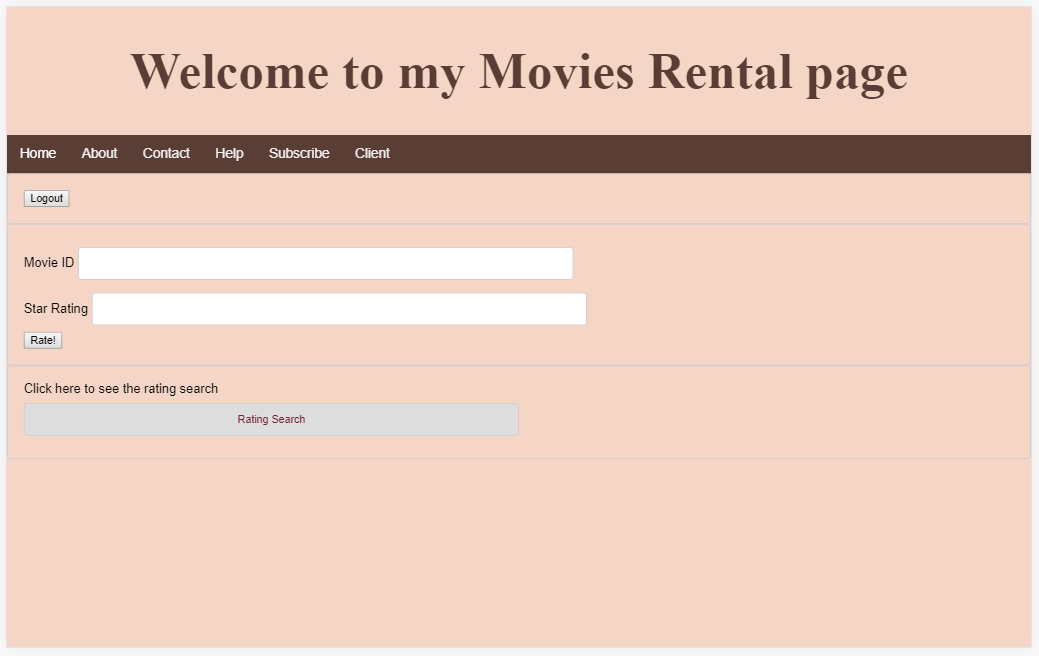


## Frequent Search Chart

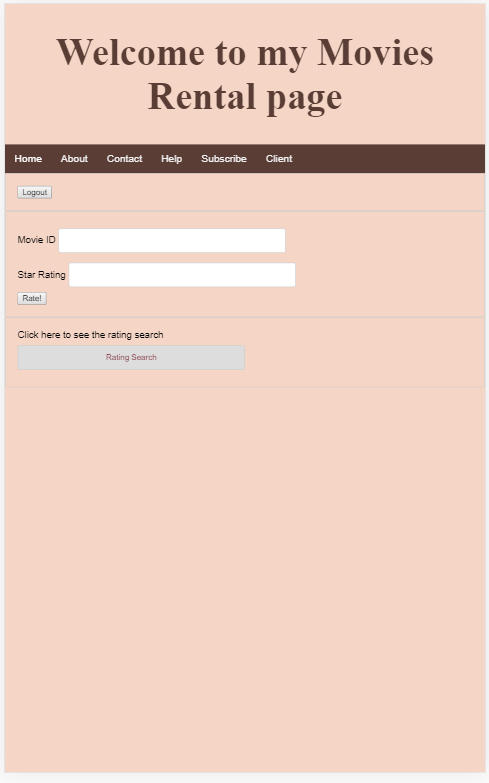
### Desktop 1920x1080



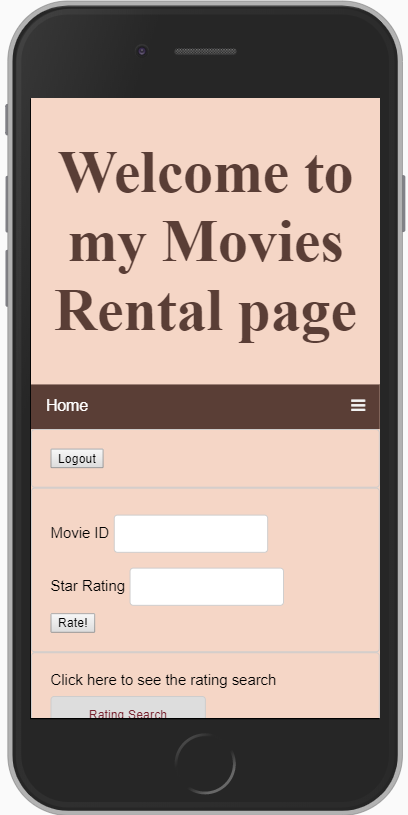
### Tablet 1280x800 Landscape



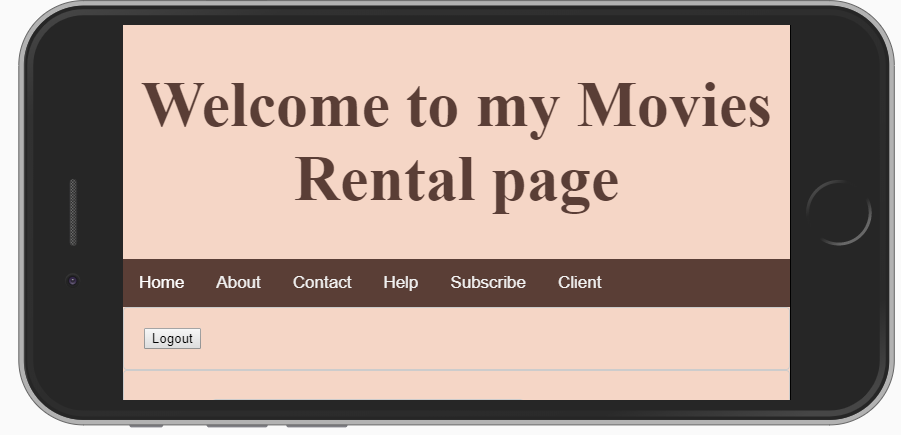
### Tablet 1280x800 Portrait



### Mobile 375x667 Portrait e.g. Iphone 6

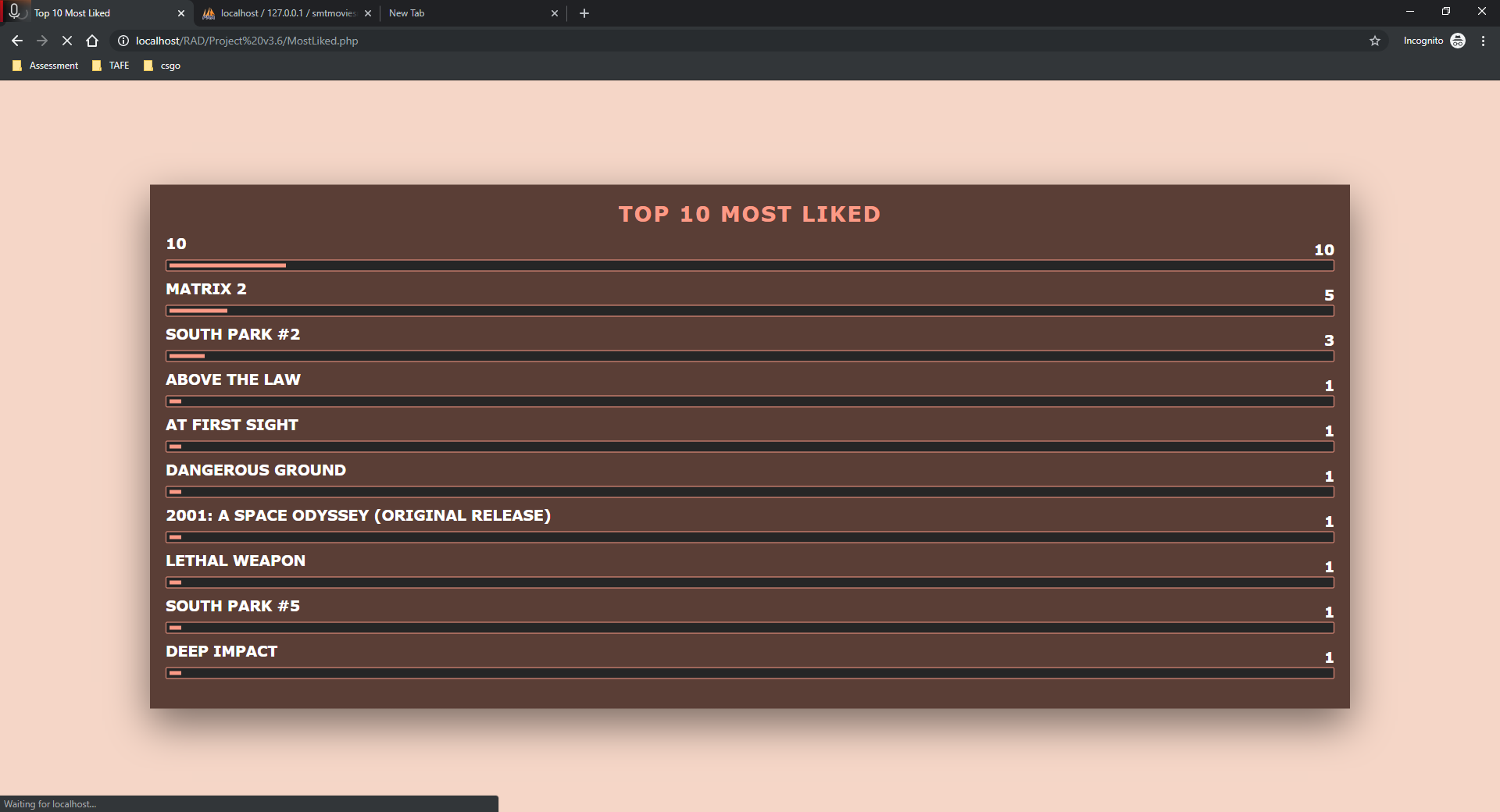


### Mobile 667x375 Landscape e.g. Iphone 6



## Top 10 most Liked

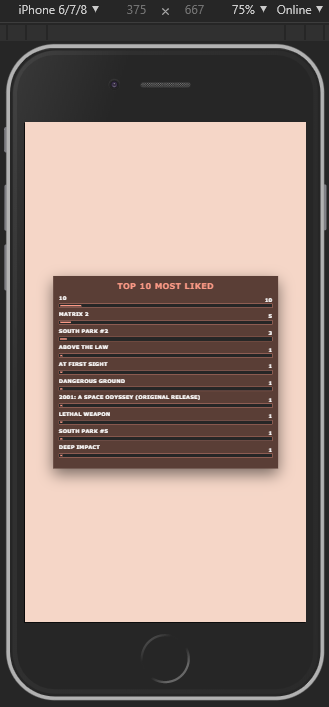
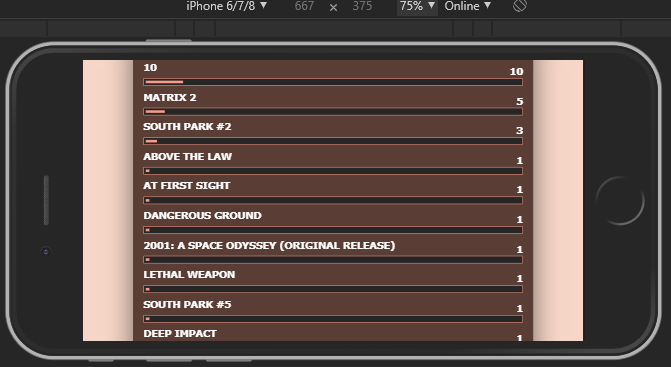
### Desktop 1920x1080



### Ipad



### Iphone 6/7/8



# Glossary

| TERM/ACRONYM | DEFINITION |
| --- | --- |
| API | Application Program Interface |
| AUT | Application Under Test |
| CSS | Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language like HTML. |
| HTML | Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser. |
| PHP | Hypertext Preprocessor is a general-purpose programming language originally designed for web development. |
| SQL | a domain-specific language used in programming and designed for managing data held in a relational database management system, or for stream processing in a relational data stream management system. |
| RAD | The RAD (Rapid Application Development) model is based on prototyping and iterative development with no specific planning involved. The process of writing the software itself involves the planning required for developing the product. |
| phpcs | PHP\_CodeSniffer is a set of two PHP scripts; the main phpcs script that tokenizes PHP, JavaScript and CSS files to detect violations of a defined coding standard, and a second phpcbf script to automatically correct coding standard violations. |
| phpcb | PHP\_CodeSniffer is a set of two PHP scripts; the main phpcs script that tokenizes PHP, JavaScript and CSS files to detect violations of a defined coding standard, and a second phpcbf script to automatically correct coding standard violations. |