

Acme Entertainment Project

weekly report

ICA DESIGN

Caspian Maclean, Andrew Williamson, Ivan Ng

22/11/2021

Contents

[Contents 1](#_Toc88467240)

[SPRINT ONE 4](#_Toc88467241)

[Source Control 4](#_Toc88467242)

[Project Management 5](#_Toc88467243)

[Kanban screenshots 5](#_Toc88467244)

[Meetings and Communication 8](#_Toc88467245)

[Initial Sprint One Meeting - minutes 8](#_Toc88467246)

[Consultation meetings with Lecturer - minutes 8](#_Toc88467247)

[Second Sprint One Meeting - minutes 8](#_Toc88467248)

[Third Sprint One Meeting - minutes 9](#_Toc88467249)

[Teams chat 9](#_Toc88467250)

[Teams chat 9](#_Toc88467251)

[Testing/Validation Plan - Testing Variables 11](#_Toc88467252)

[Responsiveness 11](#_Toc88467253)

[Data Validation 11](#_Toc88467254)

[Analysis 12](#_Toc88467255)

[CITE business rules for software development 12](#_Toc88467256)

[CITE Managed Services Quality Assurance 14](#_Toc88467257)

[Acme Entertainment Pty Ltd development requirements 15](#_Toc88467258)

[Multiplatform Report - Adaptive vs Responsive Design 17](#_Toc88467259)

[Responsive Design 17](#_Toc88467260)

[Adaptive Design 17](#_Toc88467261)

[What we went for 17](#_Toc88467262)

[Sprint One Testing 18](#_Toc88467263)

[Test Methods 18](#_Toc88467264)

[Test Table 19](#_Toc88467265)

[Screen Captures 22](#_Toc88467266)

[SPRINT TWO 28](#_Toc88467267)

[Scrum Master for Sprint Two: 28](#_Toc88467268)

[Source Control 28](#_Toc88467269)

[Project Management 29](#_Toc88467270)

[Kanban screenshots 29](#_Toc88467271)

[Meetings and Communication 31](#_Toc88467272)

[Initial Sprint Two Meeting - minutes 31](#_Toc88467273)

[Second Sprint Two Meeting – minutes 31](#_Toc88467274)

[Second Sprint Three Meeting – minutes 31](#_Toc88467275)

[Teams 32](#_Toc88467276)

[Software Review Plan 33](#_Toc88467277)

[Criteria Testing 33](#_Toc88467278)

[Performance Report 35](#_Toc88467279)

[Introduction 35](#_Toc88467280)

[Performance Criteria 35](#_Toc88467281)

[Performance Tools 35](#_Toc88467282)

[Code Optimisers 36](#_Toc88467283)

[Sprint Two Testing 37](#_Toc88467284)

[Test Table 37](#_Toc88467285)

[Screen Captures 43](#_Toc88467286)

[SPRINT THREE 59](#_Toc88467287)

[Scrum master for sprint three 59](#_Toc88467288)

[Source Control 59](#_Toc88467289)

[Project Management 60](#_Toc88467290)

[Kanban Screenshots 60](#_Toc88467291)

[Meeting and communication 62](#_Toc88467292)

[Initial meeting minutes 62](#_Toc88467293)

[Second meeting minutes 62](#_Toc88467294)

[Third online meeting minutes 63](#_Toc88467295)

[Optimisation report 64](#_Toc88467296)

[PHP Codesniffer and Beautifier 64](#_Toc88467297)

[CSS Compressor 65](#_Toc88467298)

[Test tables 67](#_Toc88467299)

[Screen captures 75](#_Toc88467300)

[References 107](#_Toc88467301)

# SPRINT ONE

In this sprint we analysed and implemented making the web app more suitable for multiple devices of different screen sizes.

The result has been uploaded to our local server, to be accessed at <http://192.168.56.102/ICA_Design/>

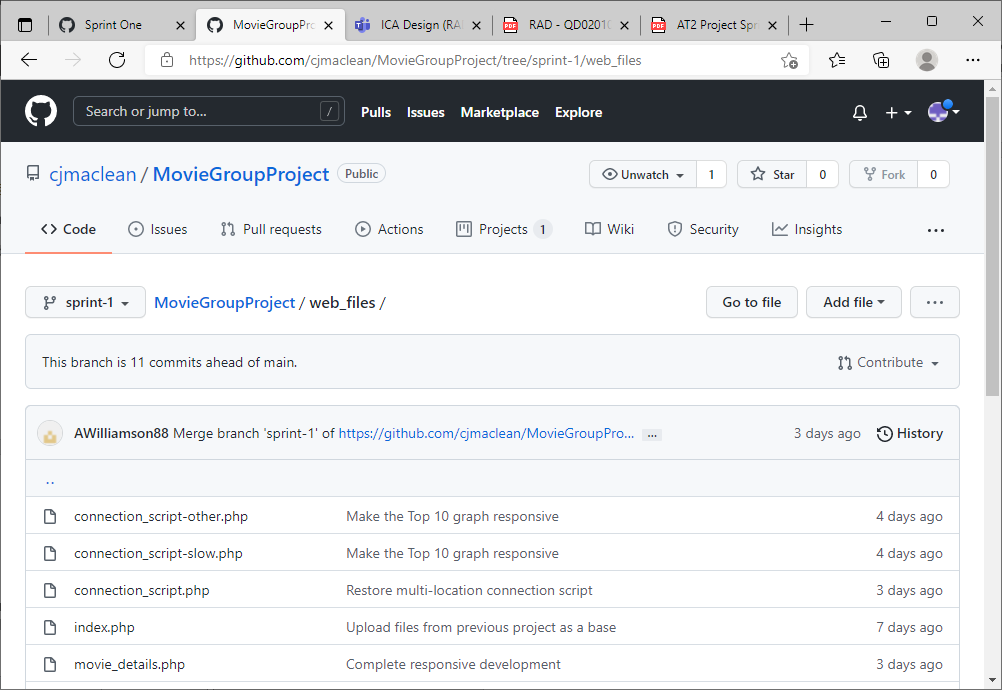
## Source Control

The project is using GitHub for source control.

The project is located at <https://github.com/cjmaclean/MovieGroupProject>

For sprint one, the development will be done under a branch “sprint-1”

Team members are using two git clients: GitHub desktop, and TortoiseGit

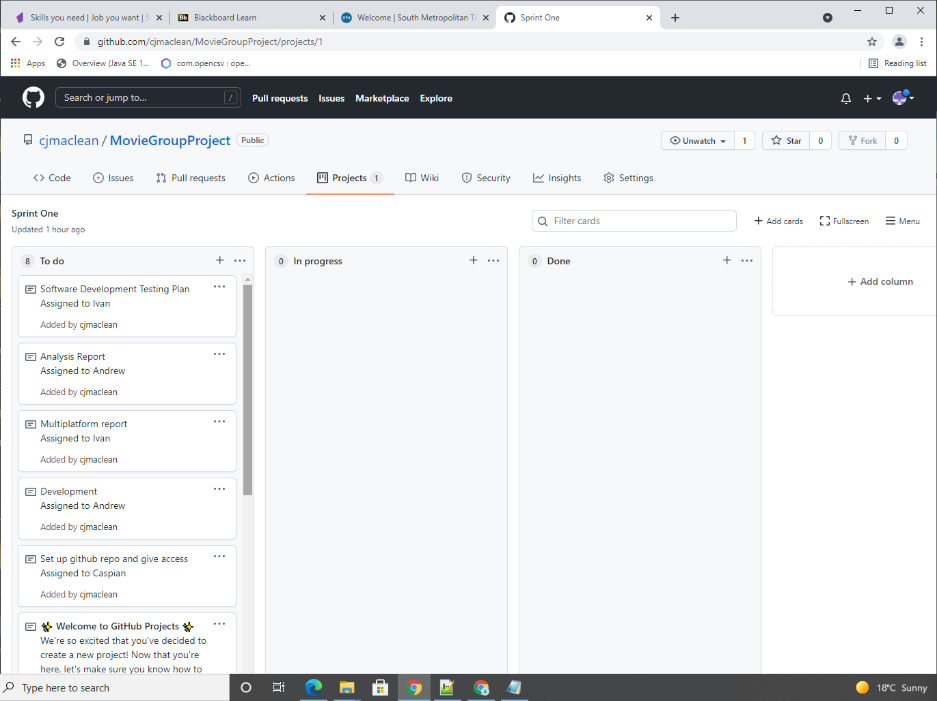


Project GitHub

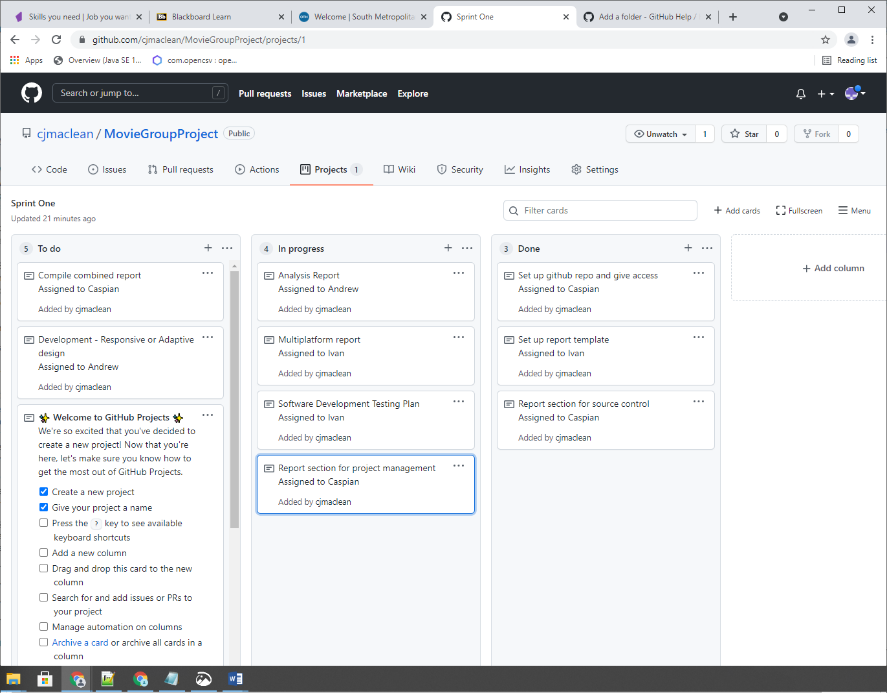
## Project Management

GitHub’s project Kanban feature will be used for project management. The first Kanban is created in a project called Sprint One inside the repository. The scrum master (Caspian for the first sprint) assigns tasks to the other team members, discussing it in scrum meetings and documenting this on the Kanban board.

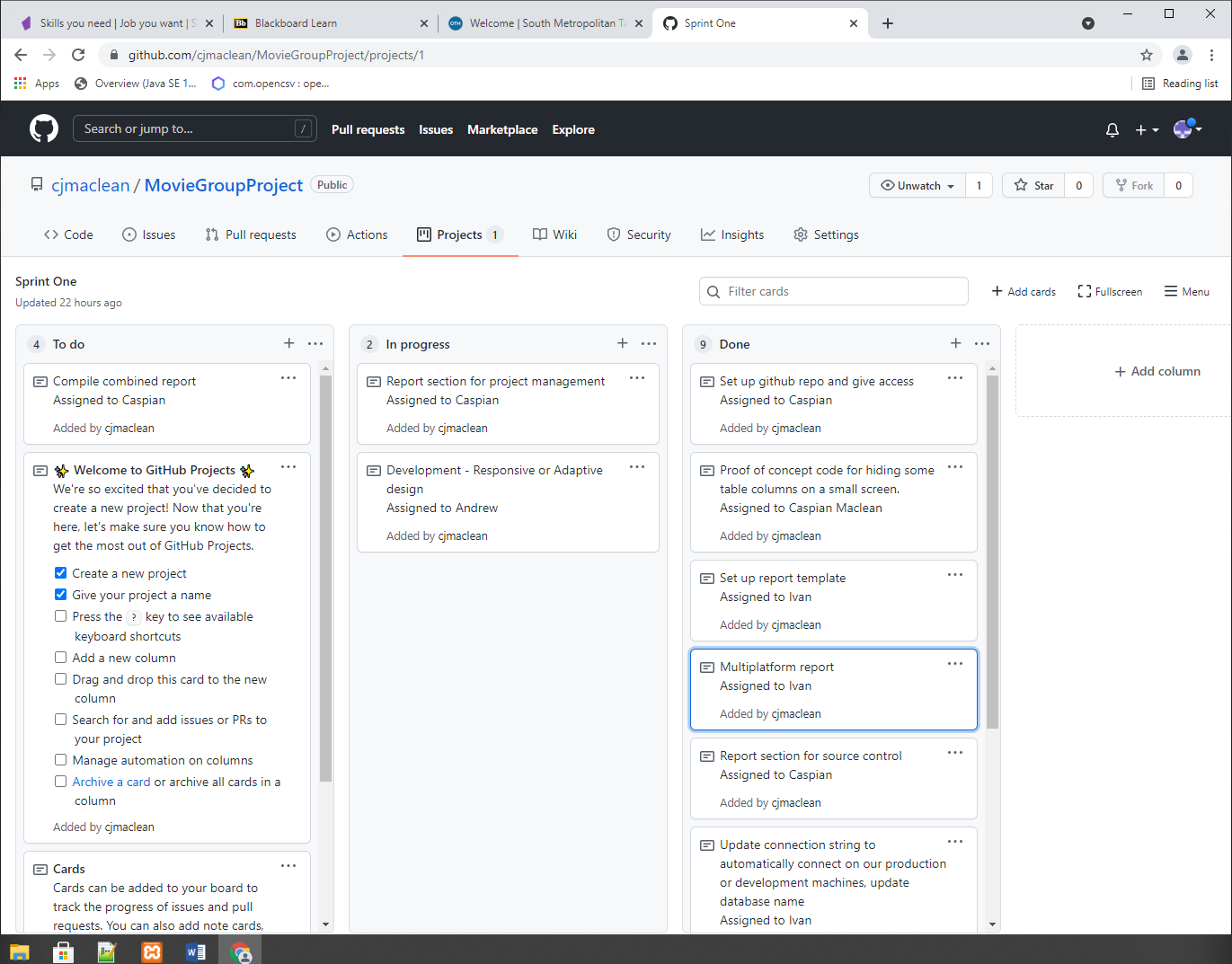
### Kanban screenshots



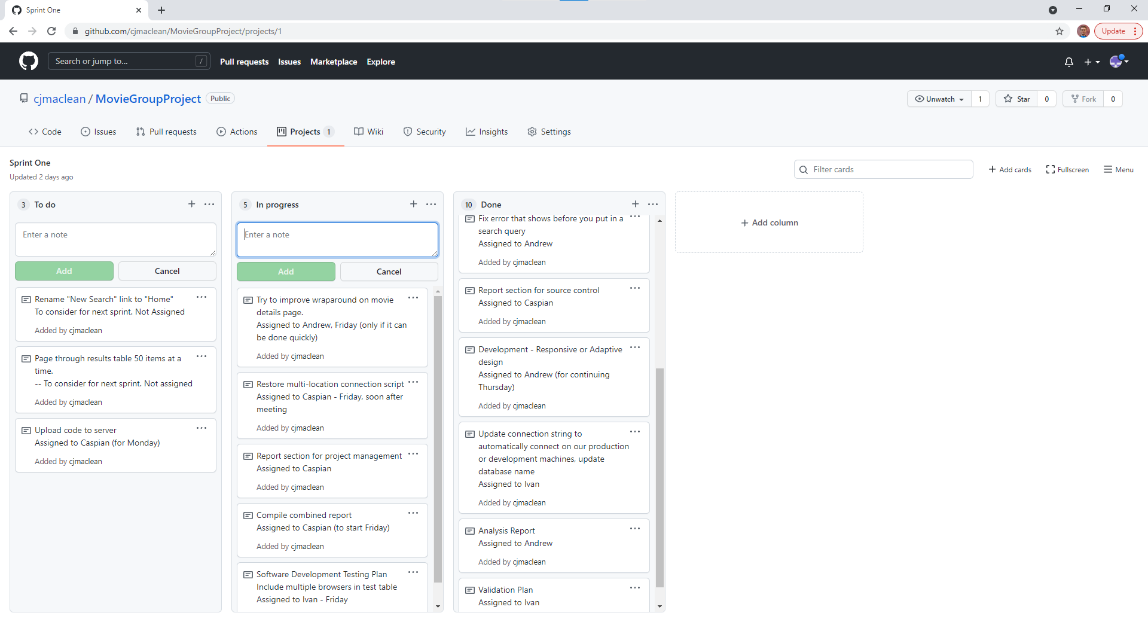
Initial Kanban board with tasks assigned



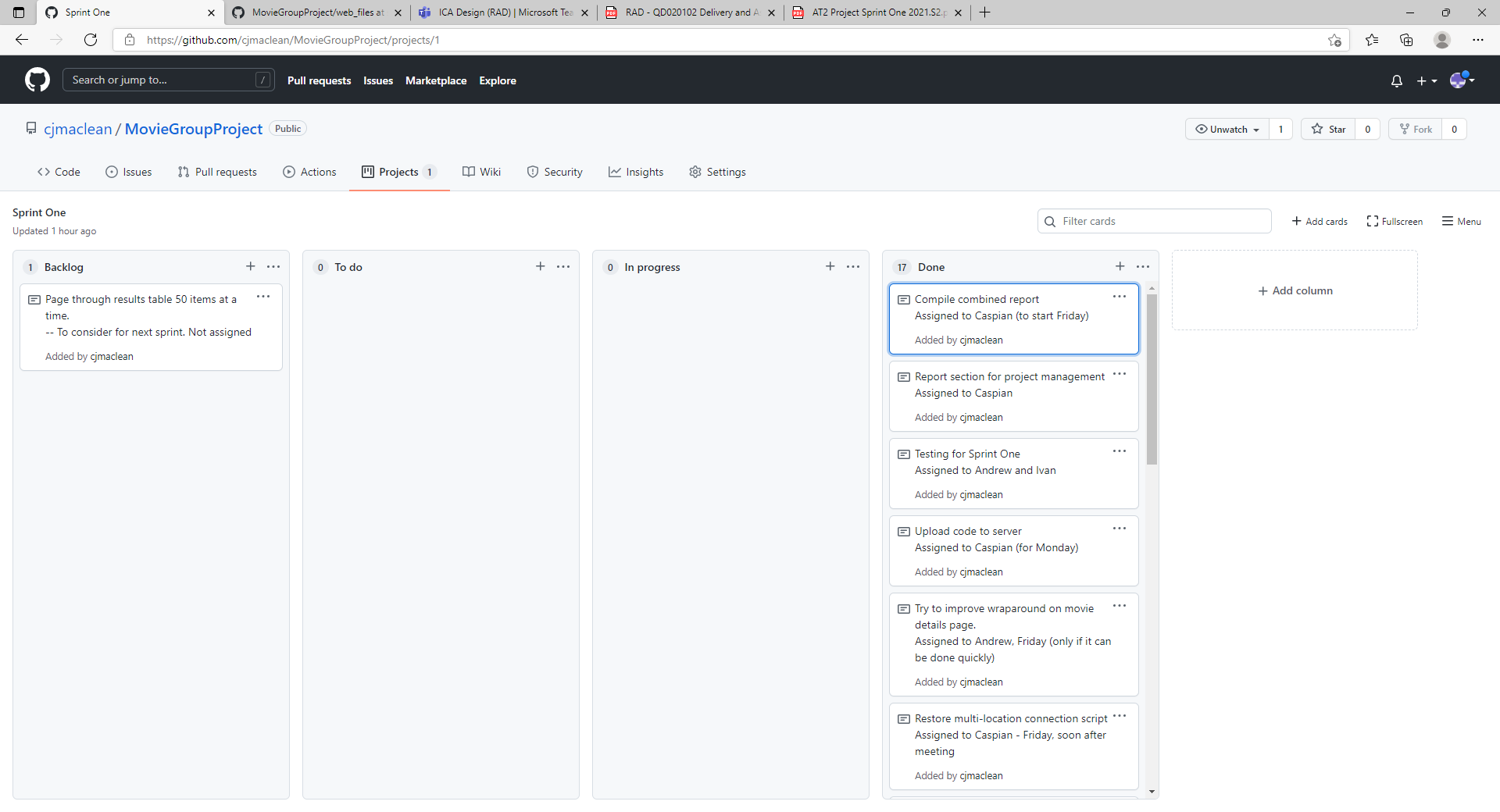
Kanban board later on the first day with some tasks in progress and done



Kanban board Wednesday morning with many tasks complete



Kanban board after Friday meeting



Kanban board after completion on Monday

## Meetings and Communication

Meeting minutes and some screenshots from the Teams chat are included below

1/11/2021

### Initial Sprint One Meeting - minutes

* Scrum master is decided as Caspian Maclean for Sprint One
* Caspian to set up GitHub repo and give access.
* Caspian to Set up Kanban project within GitHub repo (Project Management Plan)
* Team name is decided – ICA Design
* Choice of base project – we’ll look at all.
* We will use responsive design so long as that’s consistent with the Multiplatform report.
* Scrum master assigning tasks
* Master template (pick a built-in style) – Ivan.
* Demonstrating our individual projects
* Picking Caspian’s base project.
* We’ll request to use some Bootstrap.
* Bootstrap use is approved.
* Do we need notes for our other meetings? Only a couple per week. Others don’t need notes.
* We well use Teams for meetings.
* Daily updates, we’ll do in text on teams, in the morning. What we did, what we will do today, any impediments.

### Consultation meetings with Lecturer - minutes

* Can we all just edit the same document instead of the scrum master compiling the parts into one? Answer: no.

3/11/2021

### Second Sprint One Meeting - minutes

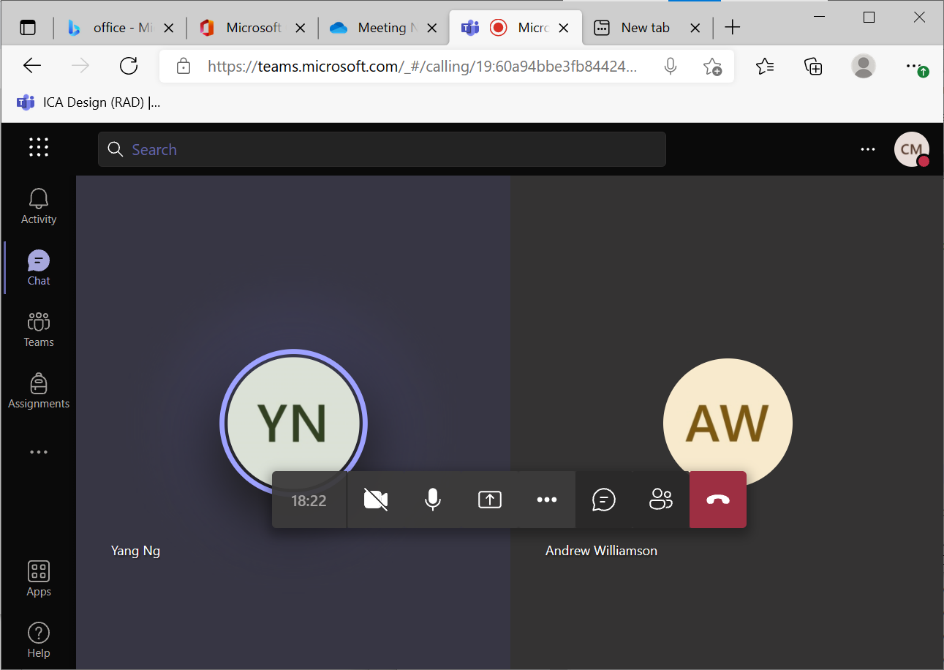
* 9:00 meeting Friday (voice meeting)
* Ivan, no more tasks yet. Maybe test table? Responsiveness. Multi browsers – Edge and Chrome.
* Have a look at the test stuff. Is it enough?
* Possible putting search results on same page as search query. Ivan to look at that briefly today.
* Andrew working on the Responsive code tomorrow.
* Caspian uploading current code to server, to test.

5/11/2021

### Third Sprint One Meeting - minutes

* Ivan worked on the Testing document. Should be able to do today.
* Caspian just tested the new version created by Andrew briefly. Today and Sunday, working on integrating reports. Also taking meeting notes and updating Kanban
* Commit the connection script that works in multiple locations – Caspian doing today soon after meeting.
* Ivan and Caspian looked at Andrew’s updates to web app – looks good and responsive.
* Andrew did the Responsive code, and fixed error in initial search form.
* Ivan proposed two ideas: show pages of results, e.g. 50 at a time. Rename “New search” to “Home”

For the third meeting we used Teams voice chat

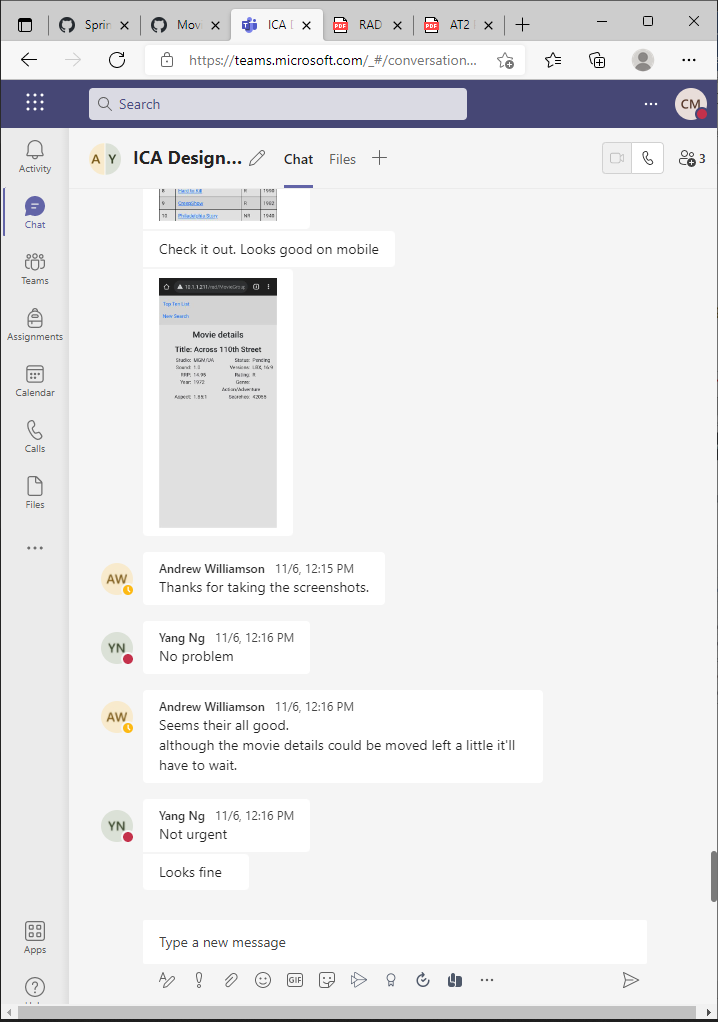
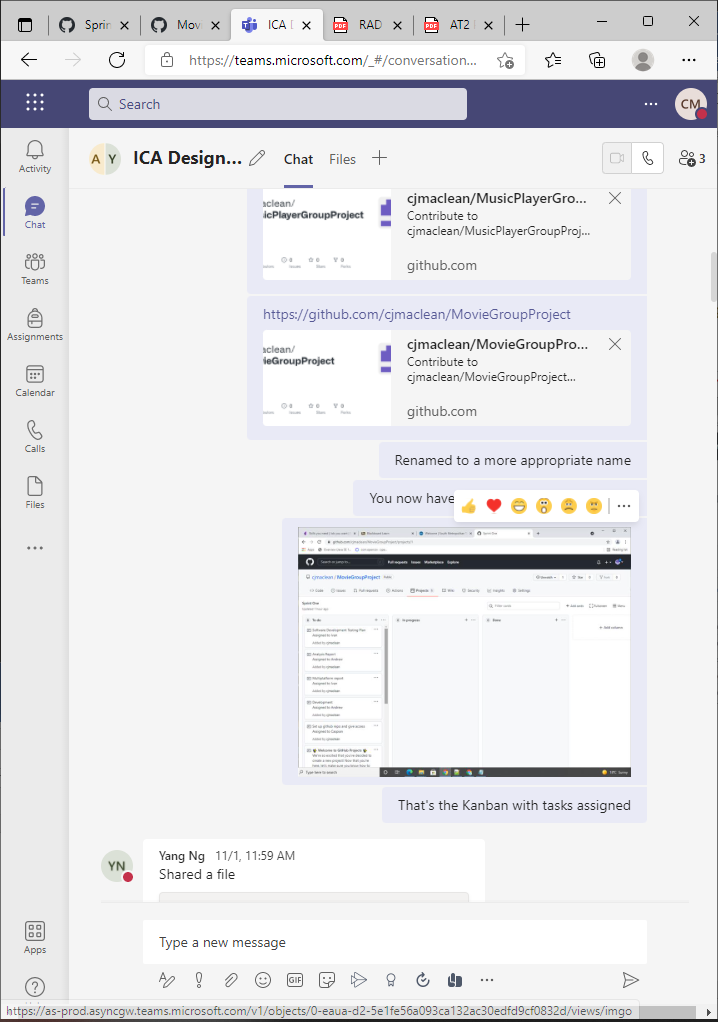


### Teams chat

We also kept in communication with Microsoft Teams

### Teams chat

We also kept in communication with Microsoft Teams



Teams chat

## Testing/Validation Plan - Testing Variables

To validate the website we will be testing our project with the variables below and comparing them with our sprints outcome to hopefully see improvement in design and performance

### Responsiveness

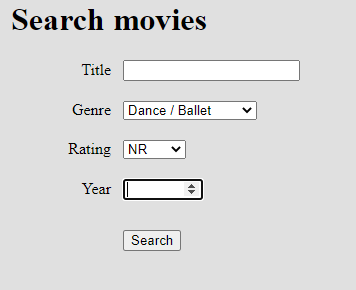
Responsive design is a website that arranges itself based off the width of the screen. A good responsive design is when different width is present different widgets/objects adjust accordingly to fit itself in an ideal arrangement.

For the project we will be making sure that no movie information gets lost and we need to make sure that text are readable in different screen sizes.



### Data Validation

To ensure that the programs search functionality is working correctly in every iteration we will validate various data types to make sure the outcome is accurate to what we’ve searched for.



## Analysis

### CITE business rules for software development

#### Coding Standards

The coding standards will reflect the context of the Language and the Clients requirements. In the absence of these criteria CITE developers will default to the industry standard (C# – Microsoft, Java – Oracle)

Coding standards are collections of coding rules, guidelines, and best practices. A coding standards document tells developers how they must write their code. Instead of each developer coding in their own preferred style, they will write all code to the standards outlined in the document. This makes sure that a large project is coded in a consistent style.

#### Common Aspects of Our Coding Standard:

* Naming Conventions
* File Naming and Organization
* Formatting and Indentation
* Comments and Documentation
* Classes, Functions and Interfaces
* Pointer and Reference Usage
* Testing

All systems and projects will be covered by the following ISO Standard(s)

ISO/IEC/IEEE 12207:2017 Systems and software engineering — Software life cycle processes

#### User Interface Development Guidelines

The user experience is the most important part of the product as it can gives the product a competitive advantage by acting as a differentiator. UX design’s main goal is to create solutions that are directly related to the business strategy which will create a holistic and optimized user experience for the user. The following guidelines should be considered a part of the software development.

* **Visibility of system status**. Users should always be informed of system operations with easy to understand and highly visible status displayed on the screen within a reasonable amount of time.
* **Match between system and the real world**. Designers should endeavour to mirror the language and concepts users would find in the real world based on who their target users are. Presenting information in logical order and piggybacking on user’s expectations derived from their real-world experiences will reduce cognitive strain and make systems easier to use.
* **User control and freedom**. Offer users a digital space where backward steps are possible, including undoing and redoing previous actions.
* **Consistency and standards**. Interface designers should ensure that both the graphic elements and terminology are maintained across similar platforms. For example, an icon that represents one category or concept should not represent a different concept when used on a different screen.
* **Error prevention**. Whenever possible, [design systems](https://www.interaction-design.org/literature/topics/design-systems) so that potential errors are kept to a minimum. Users do not like being called upon to detect and remedy problems, which may on occasion be beyond their level of expertise. Eliminating or flagging actions that may result in errors are two possible means of achieving error prevention.
* **Recognition rather than recall.**Minimize [cognitive load](https://www.interaction-design.org/literature/topics/cognitive-load) by maintaining task-relevant information within the display while users explore the interface. Human attention is limited and we are only capable of maintaining around five items in our short-term memory at one time. Due to the limitations of short-term memory, designers should ensure users can simply employ recognition instead of recalling information across parts of the dialogue. Recognizing something is always easier than recall because recognition involves perceiving cues that help us reach into our vast memory and allowing relevant information to surface. For example, we often find the format of multiple choice questions easier than short answer questions on a [test](https://www.interaction-design.org/literature/topics/test) because it only requires us to recognize the answer rather than recall it from our memory.
* **Flexibility and efficiency of use**. With increased use comes the demand for less interactions that allow faster [navigation](https://www.interaction-design.org/literature/topics/navigation-1). This can be achieved by using abbreviations, function keys, hidden commands and macro facilities. Users should be able to customize or tailor the interface to suit their needs so that frequent actions can be achieved through more convenient means.
* **Aesthetic and minimalist design**. Keep clutter to a minimum. All unnecessary information competes for the user’s limited attentional resources, which could inhibit user’s memory retrieval of relevant information. Therefore, the display must be reduced to only the necessary components for the current tasks, whilst providing clearly visible and unambiguous means of navigating to other content.
* **Help users recognize, diagnose and recover from errors**. Designers should assume users are unable to understand technical terminology, therefore, error messages should almost always be expressed in plain language to ensure nothing gets lost in translation.
* **Help and documentation**. Ideally, we want users to navigate the system without having to resort to documentation. However, depending on the type of solution, documentation may be necessary. When users require help, ensure it is easily located, specific to the task at hand and worded in a way that will guide them through the necessary steps towards a solution to the issue they are facing.

(CITE MANAGED SERVICES, 2021)

### CITE Managed Services Quality Assurance

CITE Managed Services has implemented a Quality Management System (QMS) comprising a complex set of engineering and managerial activities that ensure bespoke quality of delivered software throughout the entire workflow.

#### Quality Management Systems Tasks and Objectives

* Elaboration and implementation of procedures and regulations for software development process based on industry standards and best practices;
* Product lifecycle monitoring to ensure compliance with established processes and guidelines
* Product quality verification and validation to ensure that it complies with clients’ business needs and expectations;
* Establishment of an effective collaboration between all project team members.

#### 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.

#### Independent QA Department

CITE Managed Services QA department is an independent structural subdivision. Our QA team consists of skilled QA engineers who get involved in projects on a dedicated or an on-demand basis. The allocation of QA engineers depends on project size and complexity: they can be easily reallocated from one project to another upon necessity. Such flexibility allows us to optimize efforts and, thus, the overall project budget. Our QA Department is responsible for:

* Full-cycle QA Testing
* Document and Code Reviews
* Defect Tracking
* Configuration Management
* Process Monitoring
* Risk Management

#### QA that is Weaved into the Development Lifecycle

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. A typical QA lifecycle at CITE Managed Services includes the following phases:

1. INITIATION AND PLANNING – Project specification analysis, test plan elaboration and team assignment
2. FIRST REVIEW – Initial testing of first development deliverables, refining the test plan and test items (if necessary).
3. ITERATION AUDITS – Ongoing testing of intermediate iterations builds
4. FINAL VERIFICATION AND VALIDATION – Final product testing to ensure bespoke quality and readiness for deployment.

We utilize the full spectrum of test types as to ensure that your project enjoys in-depth quality assurance:

* Functional and Regression Testing;
* GUI and Usability Testing;
* Accessibility Testing;
* Compatibility Testing;
* Performance Testing;
* Installation / Configuration Testing;
* System / Integration Testing;
* Security Testing;
* Internationalization / Localization Testing;
* User Acceptance Testing (UAT).

(CITE MANAGED SERVICES, 2021)

### Acme Entertainment Pty Ltd development requirements

* Multi-Platform Report on the merits of the two design options currently used; adaptive and responsive.
* Choose the best option and rework the prototype.
* Modified version is to be presented at the Sprint One assessment point by the first Scrum Master.
* The development or migration of the movie database can be hosted on the cloud or suitable local server.
* Ensure your Lecturer is advised on this decision.
* Conduct and record suitable testing of the completed sprint one development, include this information in the Testing Plan.

## Multiplatform Report - Adaptive vs Responsive Design

### Responsive Design

#### Definition

Responsive design is a style in designing a website for the purpose of having elements of websites fit in available space based of the width of the browser. When the browser changes the width content dynamically change in size or rearrange itself to fit in the website.

#### Pros and Cons

|  |  |
| --- | --- |
| **Pros** | **Con** |
| Uniform and seamless | Longer mobile load times |
| SEO friendly | Elements can move around |
| Often easier to implement with many already available CSS frameworks | Ads can get lost on screen |

### Adaptive Design

#### Definition

Adaptive design is when a website has multiple fixed layout sizes. When the browser opens it will choose which layout is appropriate for the space that’s available. When resizing the browser, the design won’t change until it reaches another layouts size.

#### Pros and Cons

|  |  |
| --- | --- |
| **Pros** | **Con** |
| Allows designers to design the best UI/UX for the appropriate device | Can be labor intensive as there’s generally 3 different design per page |
| Increase in load times | Hard to maintain |
| Advertising monetization | Because it’s labor intensive it is also expensive |

### What we went for

As a team we’ve decided for the movie project to go with a responsive design approach. Because of the short amount of time we are given to complete this project it only made sense to go with responsive design as it’s less labor intensive.

## Sprint One Testing

### Test Methods

#### Regression Testing

One of the methods of testing we’ll be using for this project is regression testing. Because we’ll be implementing a lot of features, we’ll need to make sure we test thoroughly to make sure any new feature won’t have broken any previous feature.

#### End to End Testing

Another method we’ll be using will be end to end testing or e2e for short. E2e is a method used to test the functionality and performance of the application by simulating user actions and user experience.

### Test Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test #** | **Browser** | **Test Description** | **Expected Result** | **Actual Result** | **Result Screenshot** |
| 1 | Chrome | Search movie title | Table of movie search results | As Expected | Ref Figure 1 |
| 2 | Chrome | Search movie title and genre | Table of movie search results | As Expected | Ref Figure 2 |
| 3 | Chrome | Test responsiveness with mobile resolution on search page | Shows page responsiveness with necessary information | As Expected | Ref Figure 3 |
| 4 | Chrome | Test responsiveness with tablet resolution on search page | Shows page responsiveness with necessary information | As Expected | Ref Figure 4 |
| 5 | Chrome | Test responsiveness with desktop on search page | Shows page responsiveness with necessary information | As Expected | Ref Figure 5 |
| 6 | Chrome | Test responsiveness with mobile resolution on Top 10 page | Shows page responsiveness with necessary information | As Expected | Ref Figure 6 |
| 7 | Chrome | Test responsiveness with tablet resolution on top 10 page | Shows page responsiveness with necessary information | As Expected | Ref Figure 7 |
| 8 | Chrome | Test responsiveness with desktop on top 10 page | Shows page responsiveness with necessary information | As Expected | Ref Figure 8 |
| 9 | Chrome | Test responsiveness with mobile resolution on movie detail page | Shows page responsiveness with necessary information | As Expected | Ref Figure 9 |
| 10 | Chrome | Test responsiveness with tablet resolution on movie detail page | Shows page responsiveness with necessary information | As Expected | Ref Figure 10 |
| 11 | Chrome | Test responsiveness with desktop on movie detail page | Shows page responsiveness with necessary information | As Expected | Ref Figure 11 |
| 12 | Edge | Search movie title | Table of movie search results | As Expected | Ref Figure 12 |
| 13 | Edge | Search movie title and genre | Table of movie search results | As Expected | Ref Figure 13 |
| 14 | Edge | Test responsiveness with mobile resolution on search page | Shows page responsiveness with necessary information | As Expected | Ref Figure 14 |
| 15 | Edge | Test responsiveness with tablet resolution on search page | Shows page responsiveness with necessary information | As Expected | Ref Figure 15 |
| 16 | Edge | Test responsiveness with desktop on search page | Shows page responsiveness with necessary information | As Expected | Ref Figure 16 |
| 17 | Edge | Test responsiveness with mobile resolution on Top 10 page | Shows page responsiveness with necessary information | As Expected | Ref Figure 17 |
| 18 | Edge | Test responsiveness with tablet resolution on top 10 page | Shows page responsiveness with necessary information | As Expected | Ref Figure 18 |
| 19 | Edge | Test responsiveness with desktop on top 10 page | Shows page responsiveness with necessary information | As Expected | Ref Figure 19 |
| 20 | Edge | Test responsiveness with mobile resolution on movie detail page | Shows page responsiveness with necessary information | As Expected | Ref Figure 20 |
| 21 | Edge | Test responsiveness with tablet resolution on movie detail page | Shows page responsiveness with necessary information | As Expected | Ref Figure 21 |
| 22 | Edge | Test responsiveness with desktop on movie detail page | Shows page responsiveness with necessary information | As Expected | Ref Figure 22 |

### Screen Captures

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

# SPRINT TWO

In this sprint we implemented a user registration system where people who use the web application can register for the newsletter and/or the news updates. The web application was also required to allow the users to de-register and an admin to login to view a list of registered users. The admin also needed the ability to register and de-register the users.

Scrum Master for Sprint Two: Andrew Williamson

The result has been uploaded to our local server, to be accessed at <http://192.168.56.102/ICA_Design/>

## Source Control

For this sprint the source control was transferred from Caspian Maclean’s GitHub account to Andrew Williamson’s. This was to facilitate project management as Andrew Williamson was the scrum master for this sprint.

The main branch for this sprint will be sprint-2 however as the requirements for this sprint can be more easily slit among multiple team members most of the work will be done on other branches. This allows different people to work on different features on different branches.

A screenshot of a computer

Description automatically generated

Project on GitHub

## Project Management

For this sprint we used the same GitHub Kanban board system that we used in the first sprint.

For this sprint we created a new project titled sprint 2 to contain the Kanban boards.

### Kanban screenshots

Graphical user interface, application

Description automatically generated

Kanban board Monday 8th with most tasks assigned and some in progress.

A screenshot of a computer

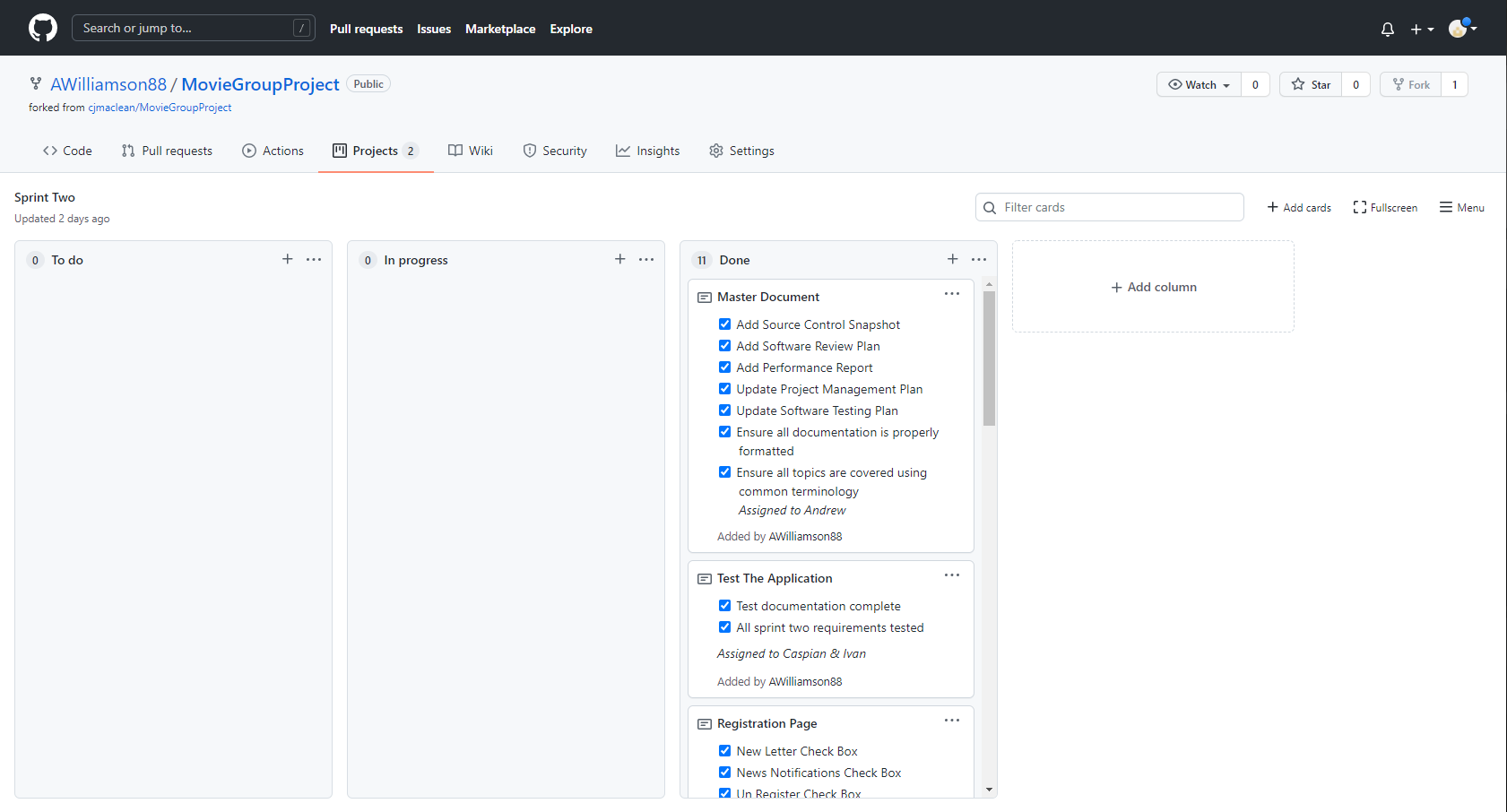
Description automatically generated

Kanban board Wednesday 10th, some jobs complete and more in progress.

Graphical user interface, application

Description automatically generated

Kanban board after Friday's voice meeting.



Kanban board Monday morning, all work complete

## Meetings and Communication

Meeting minutes and some screenshots from the Teams chat are included below.

### Initial Sprint Two Meeting - minutes

08/11/2021

* New scrum master – Andrew Williamson
* Software Review Plan assigned to Andrew Williamson (Scrum master)
* Performance Report assigned to Andrew Williamson (Scrum master)
* New colour scheme will be decided on together and implemented as we progress.
* Creating the membership database was assigned to Caspian Maclean
* Creating the admin page and it’s functionality was assigned to Caspian Maclean
* The home page was assigned to Ivan Ng (Yang Beng)
* Creating the registration page and its functionality was assigned to Ivan Ng (Yang Beng)
* Updating the Test plan was assigned to Ivan Ng (Yang Beng)

### Second Sprint Two Meeting – minutes

10/11/2021

* Planned another voice meeting on Friday 12th at 9.30AM
* Caspian has completed the admin form and layout.
* Caspian started working on the admin functionality
* Ivan completed the home page, navigation bar upgrade, registration form layout
* Ivan has started working on the registration functionality
* Andrew has completed the Software Review Plan
* Andrew has started the Performance Report

### Second Sprint Three Meeting – minutes

12/11/2021

* Ivan has completed the registration page including the functionality and validation
* Ivan has begun updating the Test plan for the sections he worked on
* Caspian has completed the admin login, registered users list, and updating the test document for his sections of the app
* Caspian currently creating the admin controls for de-registering a user.
* Andrew has completed the Performance Report and began updating the master document

### Teams

Graphical user interface, text, application, Teams

Description automatically generatedTeams chat after the third sprint two meeting.

## Software Review Plan

This plan will outline what criteria the final Movie Database application will be tested on. This is to make sure the final application meets the appropriate standard of quality, design, and functionality.

### Criteria Testing

|  |  |  |
| --- | --- | --- |
| Appearance | Yes | No |
| Good use of colour? |  |  |
| Is it pleasant to look at? |  |  |
| Is the structure properly organized? |  |  |
| Is there a header and a footer? |  |  |
| Is there a logo? |  |  |
| Is the navigation well designed? |  |  |

|  |  |  |
| --- | --- | --- |
| Functionality | Yes | No |
| Does the navigation work? |  |  |
| Can you search by any combination of movie criteria including a partial title? |  |  |
| Does the application show all the movie’s details? |  |  |
| Does it have a Top 10 chart? |  |  |
| Can users register/unregister for an email newsletter? |  |  |

|  |  |  |
| --- | --- | --- |
| User Experience | Yes | No |
| Is the application easy to use? |  |  |
| Is the text large enough to read comfortably? |  |  |
| Is the navigation and layout intuitive? |  |  |
| Is the design interesting? |  |  |

|  |  |  |
| --- | --- | --- |
| Responsiveness | Yes | No |
| Is the application visually appealing on mobile screens? |  |  |
| Does it still work on a mobile sized interface? |  |  |
| Is the application visually appealing on tablet screens? |  |  |
| Does it still work on a tablet sized interface? |  |  |
| Is the application visually appealing on desktop monitors? |  |  |
| Does it still work on desktop computer? |  |  |

|  |  |  |
| --- | --- | --- |
| Accessibility and Readability | Yes | No |
| Does the application work on different browsers? |  |  |
| Does the application work on different types of devices? |  |  |
| Does the text remain large enough to read on different screen sizes? |  |  |
| Do any images have alt text to explain what they are? |  |  |

This software review plan is designed to give the reviewed application a score out of 25. This score can then be divided by five and the resulting number can be considered the applications star rating.

## Performance Report

### Introduction

When it comes to quality assurance for web-based applications, performance testing is a vital step. Performance testing reveals how efficient the application is and how well works under pressure. Today websites are expected to respond quickly function normally regardless of how many people may be using them at the same time. Therefore, proper testing is an important part of any quality assurance process.

### Performance Criteria

Although there are many aspects that can affect performance, when it comes to web applications the main considerations are speed and reliability.

#### Performance Speed

Today websites are expected to respond within a maximum of 0.3 seconds, any longer than that and people begin to feel that the website is simply too slow. It’s quite normal for people to leave a website with a slightly longer response time and go looking for a faster one. When testing the speed of a web application it is important to verify how long each aspect of the application takes to perform its functions. This means testing the html, CSS, bootstrap, PHP, database and any other code to see how long they take to operate.

#### Performance Reliability

Load testing is the method for determining the reliability of a web application. When load testing it is important to estimate the maximum number of people that would be using the web servers at any single time and then determine how fast it would be expected to respond. Testing under those conditions will show reveal how reliable the web application truly is.

### Performance Tools

#### Developer Tools

When developing a web application, the first and most convenient performance analysis tool available is the developer tools built into some web browsers. For instance some of Google Chrome’s performance tools allow you to test and record how long your website takes to load and how it operates while you are using it, among many other things. This data can show you what features on your site are the greatest load-time wasters.

#### Third-Party Software

There are also many other applications out there that are designed for performance testing. Many of these can not only test the website but also the server side of the application by creating tens of thousands of simulated users to access the site at the same time.

### Code Optimisers

Code optimizer are another way to improve the performance of an application by improving the efficiency and removing any ambiguous or log winded code. Some code optimisers can also spot errors and/or potential errors that may cause problems further down the line.

#### CSS Optimisers

CSS optimisers often include the ability to:

* Simplify the code making sure there are no unnecessary or duplicate commands.
* Remove all the white space. The computer doesn’t need the white space to read and understand the CSS commands so removing them can reduces the file size and increase load times.
* Compress the file into a much smaller size. Often this can decrease the size of the file to only a tenth of what it was.

#### Code Sniffer

Code sniffer is a very popular tool for optimizing code as it not only works for PHP, CSS and JavaScript but is can also be adjusted to use different coding standards. Code Sniffer also comes with XAMPP, potentially the most popular free development server.

## Sprint Two Testing

### Test Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test # | Browser | Test Description | Expected Result | Actual Result | Result Screenshot |
| 1 | Chrome | Search movie title | Table of movie search results | As expected | Figure 1 |
| 2 | Chrome | Search movie title and genre | Table of movie search results | As expected | Figure 2 |
| 3 | Chrome | Test responsiveness with mobile resolution on search page | Shows page responsiveness with necessary information | As expected | Figure 3 |
| 4 | Chrome | Test responsiveness with tablet resolution on search page | Shows page responsiveness with necessary information | As expected | Figure 4 |
| 5 | Chrome | Test responsiveness with desktop on search page | Shows page responsiveness with necessary information | As expected | Figure 5 |
| 6 | Chrome | Test responsiveness with mobile resolution on Top 10 page | Shows page responsiveness with necessary information | As expected | Figure 6 |
| 7 | Chrome | Test responsiveness with tablet resolution on top 10 page | Shows page responsiveness with necessary information | As expected | Figure 7 |
| 8 | Chrome | Test responsiveness with desktop on top 10 page | Shows page responsiveness with necessary information | As expected | Figure 8 |
| 9 | Chrome | Test responsiveness with mobile resolution on movie detail page | Shows page responsiveness with necessary information | As expected | Figure 9 |
| 10 | Chrome | Test responsiveness with tablet resolution on movie detail page | Shows page responsiveness with necessary information | As expected | Figure 10 |
| 11 | Chrome | Test responsiveness with desktop on movie detail page | Shows page responsiveness with necessary information | As expected | Figure 11 |
| 12 | Edge | Search movie title | Table of movie search results | As expected | Figure 12 |
| 13 | Edge | Search movie title and genre | Table of movie search results | As expected | Figure 13 |
| 14 | Edge | Test responsiveness with mobile resolution on search page | Shows page responsiveness with necessary information | As expected | Figure 14 |
| 15 | Edge | Test responsiveness with tablet resolution on search page | Shows page responsiveness with necessary information | As expected | Figure 15 |
| 16 | Edge | Test responsiveness with desktop on search page | Shows page responsiveness with necessary information | As expected | Figure 16 |
| 17 | Edge | Test responsiveness with mobile resolution on Top 10 page | Shows page responsiveness with necessary information | As expected | Figure 17 |
| 18 | Edge | Test responsiveness with tablet resolution on top 10 page | Shows page responsiveness with necessary information | As expected | Figure 18 |
| 19 | Edge | Test responsiveness with desktop on top 10 page | Shows page responsiveness with necessary information | As expected | Figure 19 |
| 20 | Edge | Test responsiveness with mobile resolution on movie detail page | Shows page responsiveness with necessary information | As expected | Figure 20 |
| 21 | Edge | Test responsiveness with tablet resolution on movie detail page | Shows page responsiveness with necessary information | As expected | Figure 21 |
| 22 | Edge | Test responsiveness with desktop on movie detail page | Shows page responsiveness with necessary information | As expected | Figure 22 |
| 23 | Chrome | Open admin.php | Form displays without showing members | As expected | Figure 23 |
| 24 | Chrome | Click button to show members on admin page | Form displays and members shown in table | As expected | Figure 24 |
| 25 | Chrome | Test responsiveness with mobile resolution on admin page | Shows page responsiveness with necessary information | As expected | Figure 25 |
| 26 | Chrome | Test responsiveness with desktop on admin page | Shows page responsiveness with necessary information | As expected | Figure 26 |
| 27 | Chrome | Unsubscribe member using table link on the admin page | All subscriptions cleared for member | As expected | Figure 27 |
| 28 | Chrome | Unsubscribe member using e-mail field on the admin page | All subscriptions cleared for member | As expected | Figure 28 |
| 29 | Chrome | Unsubscribe non-existent member using e-mail field on the admin page | Web page says there is no member with the e-mail address. | As expected | Figure 29 |
| 30 | Edge | Test responsiveness with mobile resolution on admin page | Shows page responsiveness with necessary information | As expected (acceptable but not great, due to wrapping of word “E-mail”) | Figure 30 |
| 31 | Edge | Test responsiveness with desktop on admin page | Shows page responsiveness with necessary information | As expected | Figure 31 |
| 32 | Chrome | Register valid user | User gets registered | As expected | Figure 32 |
| 33 | Chrome | Attempt register with bad email | Error message | As expected | Figure 33 |
| 34 | Chrome | Attempt to register with missing first name | Error message | As expected | Figure 34 |
| 35 | Chrome | Attempt register with missing last name | Error message | As expected | Figure 35 |
| 36 | Edge | Register valid user | User gets registered | As expected | Figure 36 |
| 37 | Edge | Attempt register with bad email | Warning displayed | As expected | Figure 37 |
| 38 | Edge | Attempt to register with missing first name | Warning displayed | As expected | Figure 38 |
| 39 | Edge | Attempt register with missing last name | Warning displayed | As expected | Figure 39 |

### Screen Captures

|  |  |  |
| --- | --- | --- |
| Figure 1 |  |  |
| Figure 2 |  |  |
| Figure 3 |  |  |
| Figure 4 |  |  |
| Figure 5 |  |  |
| Figure 6 |  |  |
| Figure 7 |  |  |
| Figure 8 |  |  |
| Figure 9 |  |  |
| Figure 10 |  |  |
| Figure 11 |  |  |
| Figure 12 |  |  |
| Figure 13 |  |  |
| Figure 14 |  |  |
| Figure 15 |  |  |
| Figure 16 |  |  |
| Figure 17 |  |  |
| Figure 18 |  |  |
| Figure 19 |  |  |
| Figure 20 |  |  |
| Figure 21 |  |  |
| Figure 22 |  |  |
| Figure 23 |  |  |
| Figure 24 |  |  |
| Figure 25 |  |  |
| Figure 26 |  |  |
| Figure 27 |  |  |
| Figure 28 |  |  |
| Figure 29 |  |  |
| Figure 30 |  |  |
| Figure 31 |  |  |
| Figure 32 |  |  |
| Figure 33 |  |  |
| Figure 34 |  |  |
| Figure 35 |  |  |
| Figure 36 |  |  |
| Figure 37 |  |  |
| Figure 38 |  |  |
| Figure 39 |  |  |

# SPRINT THREE

For the final spring we’ve added basic login functionality which allows admins to register new Acme Personnel members and choose if they’re admin or not. Admins have more features in the admin page such as being able to register more people and unsubscribe users from newsletter and news flash. We have also updated the Top10 page from Top10 most searched to Top10 highest rated movies which will update when our new rating feature gets updated. In the movie search page.

## Scrum master for sprint three

The scrum master for this sprint is Yang Beng Ng (Ivan)

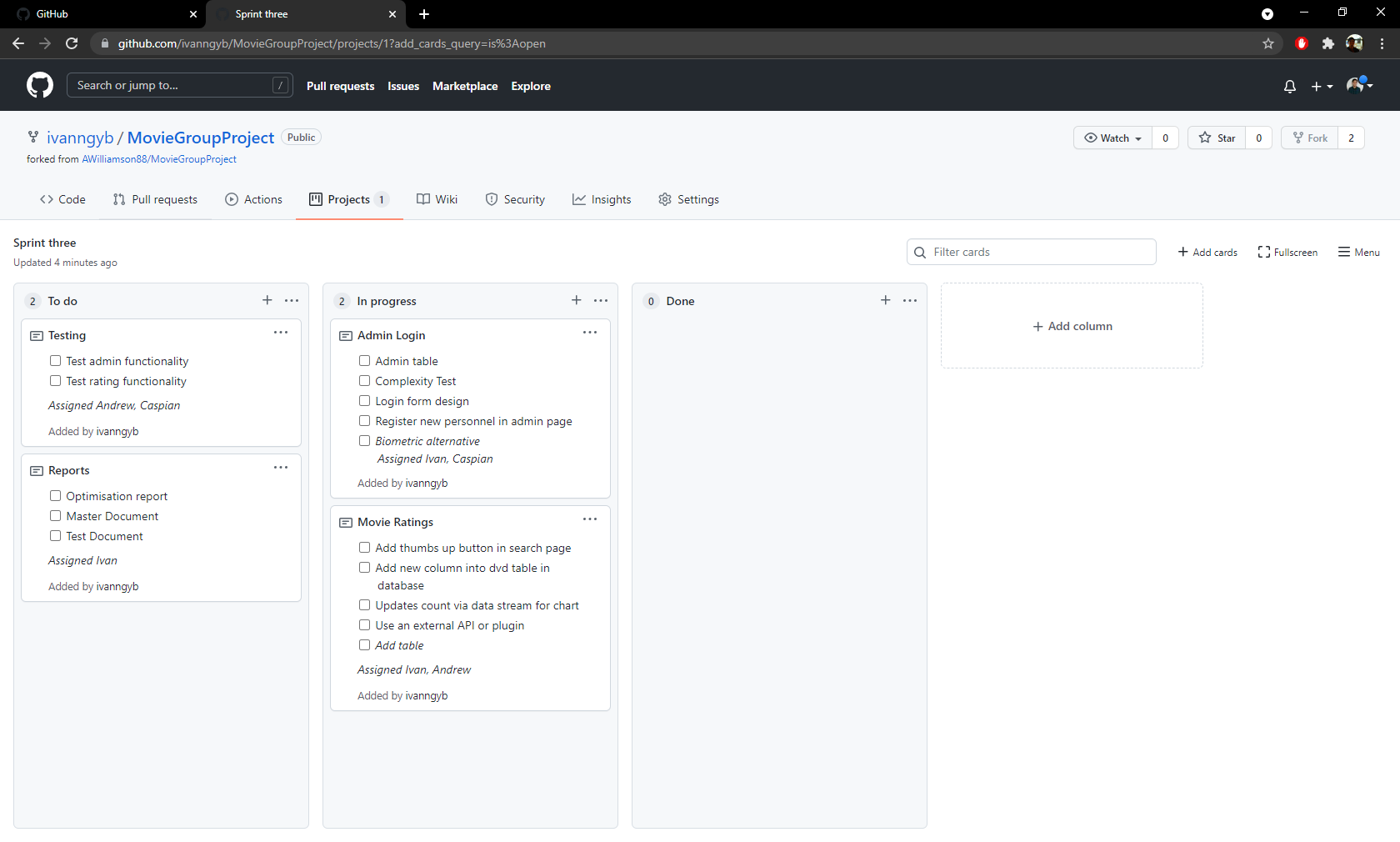
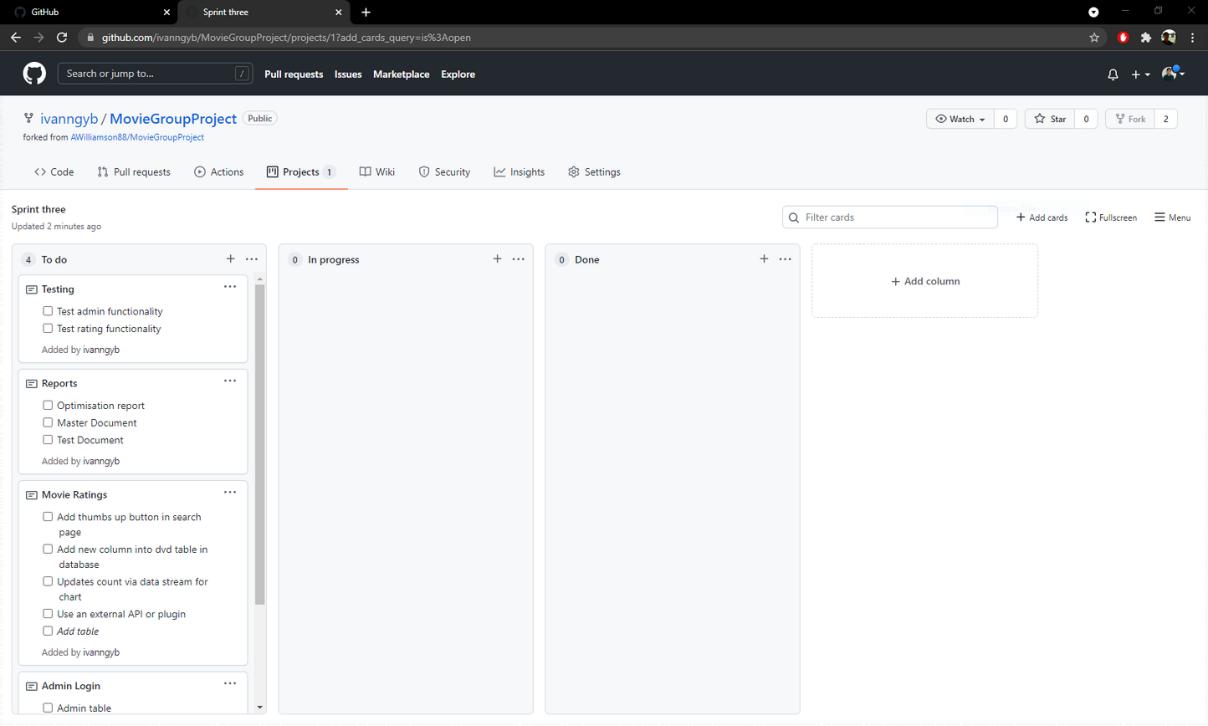
## Source Control

After finishing our last sprint the project was forked and transferred over to <https://github.com/ivanngyb/MovieGroupProject>



## Project Management

### Kanban Screenshots

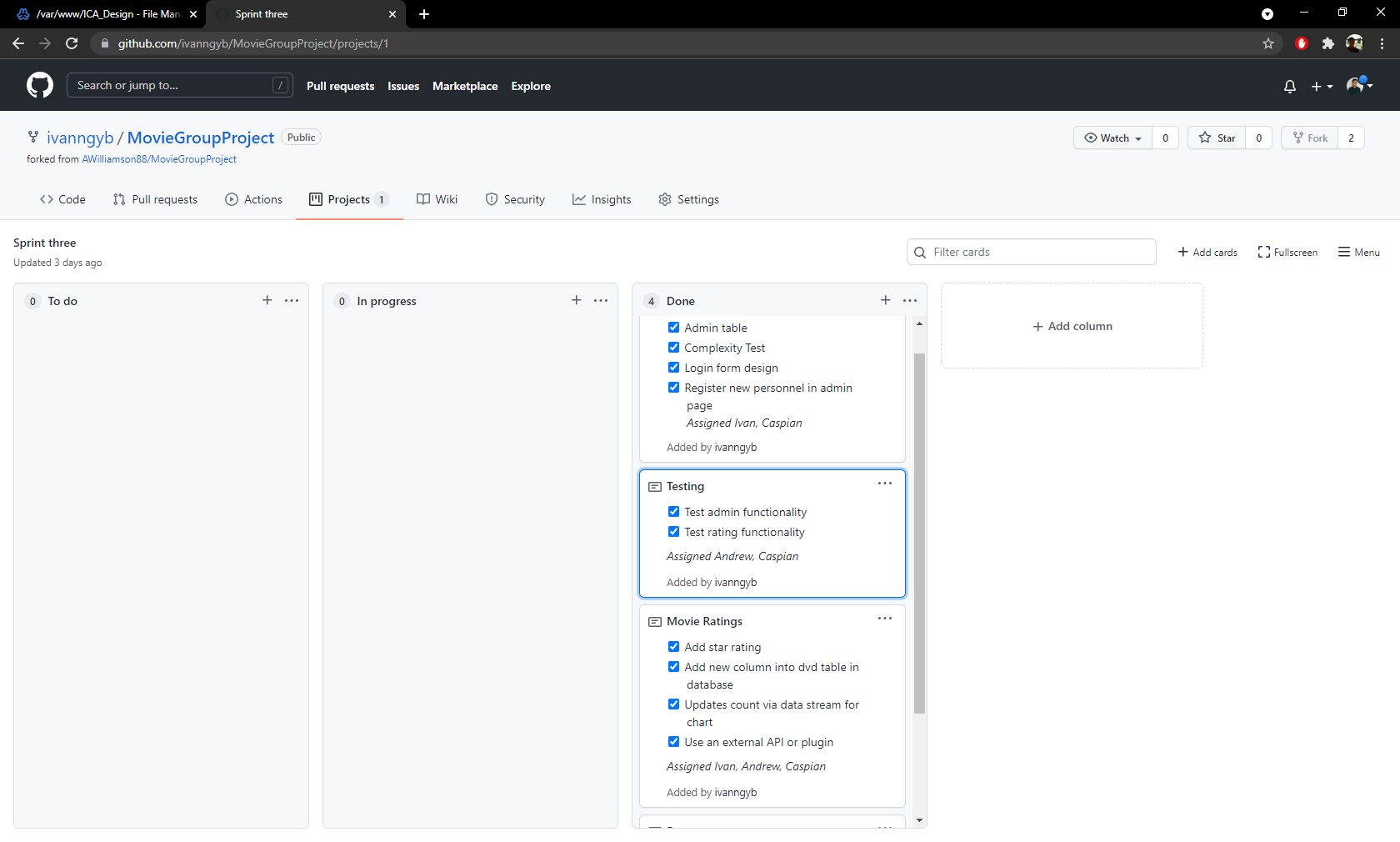


A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence



## Meeting and communication

### Initial meeting minutes

15/11/2021

* New scrum master – Yang Beng Ng (Ivan)
* Ivan assigned to working on login functionality and star rating front end
* Andrew got assigned to do top 10 graphs
* Caspian assigned to updating tables in database to take star ratings and doing back end for updating star rating

### Second meeting minutes

17/11/2021

* Caspian completed backend for star rating
* Caspian assigned to making registering functionality
* Ivan completed login functionality
* Ivan to work on star rating front end
* Andrew managed to get graph working now assigned to linking it up with database

### Third online meeting minutes

Graphical user interface, application, Teams

Description automatically generated

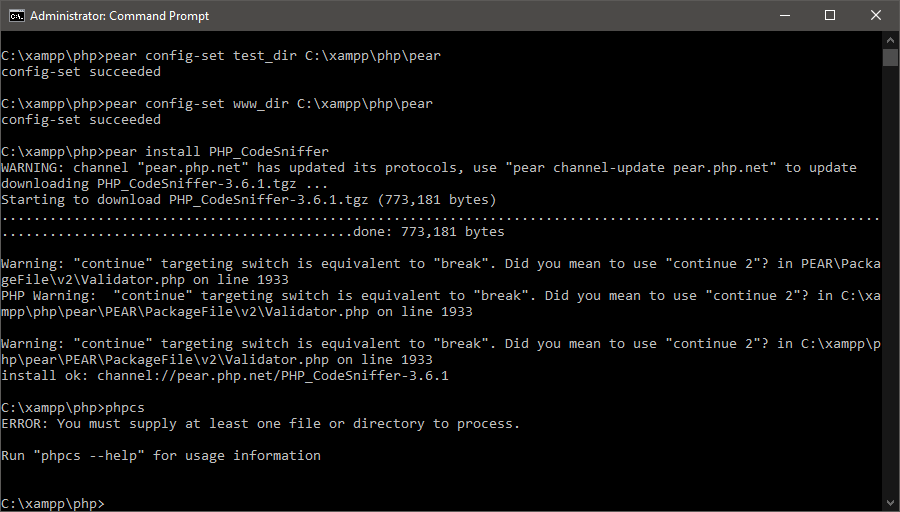
19/11/2021

* Caspian completed admin page to work with admin and non admin
* Caspian assigned to potentially start optimizing scripts
* Ivan completed star rating
* Ivan to work on helping with optimization
* Andrew is very close to finishing the graph

## Optimisation report

### PHP Codesniffer and Beautifier

For our php files we’ve used PHP Codesniifer and its beautifier functions to optimise our scripts.



Installation of PHP\_CodeSniffer shown above

Text

Description automatically generated

Running PHPCS on our folder. Getting a lot of errors.

Text

Description automatically generated

Running PHP Code Beautifier on the folder to optimise all php and javascripts within folder.

### CSS Compressor

A screenshot of a computer

Description automatically generated with medium confidence

CSS Compressor is a Visual Studio Code extension that optimizes CSS files by minifying them.

A screenshot of a computer

Description automatically generated with medium confidence

Before using CSS Compressor the code has a lot of comments and length of file is very long.

A computer screen capture

Description automatically generated with medium confidence

After using CSS Compressor the code is minify and comments are removed. The document went from having 240 lines of code to 40 lines making it smaller and more optimised.

## Test tables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test # | Browser | Test Description | Expected Result | Actual Result | Result Screenshot |
| 1 | Chrome | Search movie title | Table of movie search results | As expected | Figure 1 |
| 2 | Chrome | Search movie title and genre | Table of movie search results | As expected | Figure 2 |
| 3 | Chrome | Test responsiveness with mobile resolution on search page | Shows page responsiveness with necessary information | As expected | Figure 3 |
| 4 | Chrome | Test responsiveness with tablet resolution on search page | Shows page responsiveness with necessary information | As expected | Figure 4 |
| 5 | Chrome | Test responsiveness with desktop on search page | Shows page responsiveness with necessary information | As expected | Figure 5 |
| 6 | Chrome | Test responsiveness with mobile resolution on Top 10 page | Shows page responsiveness with necessary information | As expected | Figure 6 |
| 7 | Chrome | Test responsiveness with tablet resolution on top 10 page | Shows page responsiveness with necessary information | As expected | Figure 7 |
| 8 | Chrome | Test responsiveness with desktop on top 10 page | Shows page responsiveness with necessary information | As expected | Figure 8 |
| 9 | Chrome | Test responsiveness with mobile resolution on movie detail page | Shows page responsiveness with necessary information | As expected | Figure 9 |
| 10 | Chrome | Test responsiveness with tablet resolution on movie detail page | Shows page responsiveness with necessary information | As expected | Figure 10 |
| 11 | Chrome | Test responsiveness with desktop on movie detail page | Shows page responsiveness with necessary information | As expected | Figure 11 |
| 12 | Edge | Search movie title | Table of movie search results | As expected | Figure 12 |
| 13 | Edge | Search movie title and genre | Table of movie search results | As expected | Figure 13 |
| 14 | Edge | Test responsiveness with mobile resolution on search page | Shows page responsiveness with necessary information | As expected | Figure 14 |
| 15 | Edge | Test responsiveness with tablet resolution on search page | Shows page responsiveness with necessary information | As expected | Figure 15 |
| 16 | Edge | Test responsiveness with desktop on search page | Shows page responsiveness with necessary information | As expected | Figure 16 |
| 17 | Edge | Test responsiveness with mobile resolution on Top 10 page | Shows page responsiveness with necessary information | As expected | Figure 17 |
| 18 | Edge | Test responsiveness with tablet resolution on top 10 page | Shows page responsiveness with necessary information | As expected | Figure 18 |
| 19 | Edge | Test responsiveness with desktop on top 10 page | Shows page responsiveness with necessary information | As expected | Figure 19 |
| 20 | Edge | Test responsiveness with mobile resolution on movie detail page | Shows page responsiveness with necessary information | As expected | Figure 20 |
| 21 | Edge | Test responsiveness with tablet resolution on movie detail page | Shows page responsiveness with necessary information | As expected | Figure 21 |
| 22 | Edge | Test responsiveness with desktop on movie detail page | Shows page responsiveness with necessary information | As expected | Figure 22 |
| 23 | Chrome | Open admin.php | Form displays without showing members | As expected | Figure 23 |
| 24 | Chrome | Click button to show members on admin page | Form displays and members shown in table | As expected | Figure 24 |
| 25 | Chrome | Test responsiveness with mobile resolution on admin page | Shows page responsiveness with necessary information | As expected | Figure 25 |
| 26 | Chrome | Test responsiveness with desktop on admin page | Shows page responsiveness with necessary information | As expected | Figure 26 |
| 27 | Chrome | Unsubscribe member using table link on the admin page | All subscriptions cleared for member | As expected | Figure 27 |
| 28 | Chrome | Unsubscribe member using e-mail field on the admin page | All subscriptions cleared for member | As expected | Figure 28 |
| 29 | Chrome | Unsubscribe non-existent member using e-mail field on the admin page | Web page says there is no member with the e-mail address. | As expected | Figure 29 |
| 30 | Edge | Test responsiveness with mobile resolution on admin page | Shows page responsiveness with necessary information | As expected (acceptable but not great, due to wrapping of word “E-mail”) | Figure 30 |
| 31 | Edge | Test responsiveness with desktop on admin page | Shows page responsiveness with necessary information | As expected | Figure 31 |
| 32 | Chrome | Register valid user | User gets registered | As expected | Figure 32 |
| 33 | Chrome | Attempt register with bad email | Error message | As expected | Figure 33 |
| 34 | Chrome | Attempt to register with missing first name | Error message | As expected | Figure 34 |
| 35 | Chrome | Attempt register with missing last name | Error message | As expected | Figure 35 |
| 36 | Edge | Register valid user | User gets registered | As expected | Figure 36 |
| 37 | Edge | Attempt register with bad email | Warning displayed | As expected | Figure 37 |
| 38 | Edge | Attempt to register with missing first name | Warning displayed | As expected | Figure 38 |
| 39 | Edge | Attempt register with missing last name | Warning displayed | As expected | Figure 39 |
| 40 | Chrome | Search for a movie and add a star rating. | The rating updates. | As expected | Figure 40 |
| 41 | Chrome | Attempt to rate the same movie again. | Unable to rate the same movie twice. | As expected | Figure 41 |
| 42 | Chrome | Attempt to rate different movie. | The rating of that movie updates. | As expected | Figure 42 |
| 43 | Chrome | Load up the top ten chart and change the ratings of the movies. | The top ten chart changes as the movie ratings are changed. | As expected | Figure 43 |
| 44 | Chrome | Change the ratings of the movies in the chart. | The Table below the top ten chart changes with the star ratings. | As expected | Figure 44 |
| 45 | Edge | Search for a movie and add a star rating. | The rating updates. | As expected | Figure 45 |
| 46 | Edge | Attempt to rate the same movie again. | Unable to rate the same movie twice. | As expected | Figure 46 |
| 47 | Edge | Attempt to rate different movie. | The rating of that movie updates. | As expected | Figure 47 |
| 48 | Edge | Load up the top ten chart and change the ratings of the movies. | The top ten chart changes as the movie ratings are changed. | As expected | Figure 48 |
| 49 | Edge | Change the ratings of the movies in the chart. | The Table below the top ten chart changes with the star ratings. | As expected | Figure 49 |
| 50 | Edge | Login with wrong username “caspianmaclean” | Cannot log in | As expected | Figure 50 |
| 51 | Edge | Login as admin but wrong password “whatever” | Cannot log in | As expected | Figure 51 |
| 52 | Edge | Login as admin with correct password | Shows admin page, nav bar shows admin page | As expected | Figure 52 |
| 53 | Edge | Admin page as admin, click show members | We can see the page, with unsubscribe email field, the members, with unsubscribe link on the table rows | As expected | Figure 53 |
| 54 | Edge | Admin add user no name | Fails, warns about no username | As expected | Figure 54 |
| 55 | Edge | Add user caspianmaclean / ezpassword | Fails, password not complex enough | As expected | Figure 55 |
| 56 | Edge | Add user caspianmaclean / passWORD99 / not admin | Succeeds, entry in database | As expected | Figure 56 |
| 57 | Edge | Log out | Goes to home page | As expected | Figure 57 |
| 58 | Edge | Go to admin page while logged out | Redirected to home page | As expected | Figure 58 |
| 59 | Edge | Log in as caspianmaclean, click list members | Shows as non-admin, no option to create new accounts, or unsubscribe members (button or table column) | As expected | Figure 59 |
| 60 | Chrome | Admin page responsive, desktop size vs phone size | Looks okay in both sizes | As expected | Figure 60 |
| 61 | Edge | Admin page responsive, desktop size vs phone size | Looks okay in both sizes | As expected | Figure 61 |
| 62 | Chrome | Login responsive, desktop size vs phone size | Looks okay in both sizes | As expected | Figure 62 |
| 63 | Edge | Login responsive, desktop size vs phone size | Looks okay in both sizes | As expected | Figure 63 |

## Screen captures

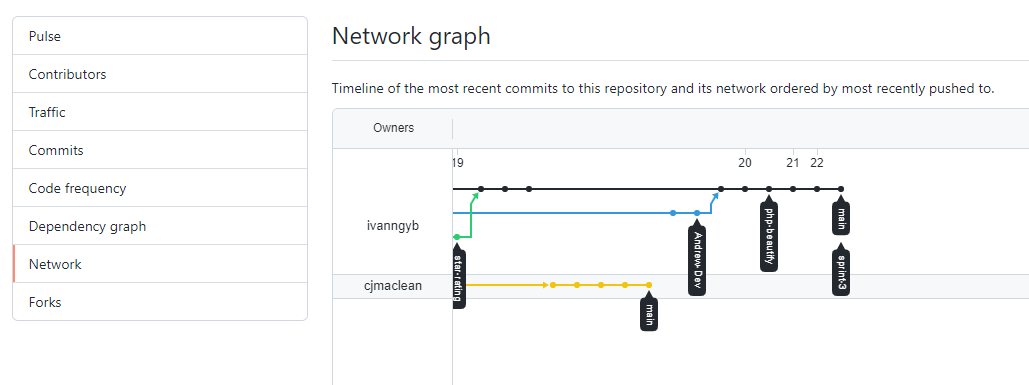
|  |  |  |
| --- | --- | --- |
| Figure 1 |  |  |
| Figure 2 |  |  |
| Figure 3 |  |  |
| Figure 4 |  |  |
| Figure 5 |  |  |
| Figure 6 |  |  |
| Figure 7 |  |  |
| Figure 8 |  |  |
| Figure 9 |  |  |
| Figure 10 |  |  |
| Figure 11 |  |  |
| Figure 12 |  |  |
| Figure 13 |  |  |
| Figure 14 |  |  |
| Figure 15 |  |  |
| Figure 16 |  |  |
| Figure 17 |  |  |
| Figure 18 |  |  |
| Figure 19 |  |  |
| Figure 20 |  |  |
| Figure 21 |  |  |
| Figure 22 |  |  |
| Figure 23 |  |  |
| Figure 24 |  |  |
| Figure 25 |  |  |
| Figure 26 |  |  |
| Figure 27 |  |  |
| Figure 28 |  |  |
| Figure 29 |  |  |
| Figure 30 |  |  |
| Figure 31 |  |  |
| Figure 32 |  |  |
| Figure 33 |  |  |
| Figure 34 |  |  |
| Figure 35 |  |  |
| Figure 36 |  |  |
| Figure 37 |  |  |
| Figure 38 |  |  |
| Figure 39 |  |  |
| Figure 40 |  |  |
| Figure 41 |  |  |
| Figure 42 |  |  |
| Figure 43 |  |  |
| Figure 44 |  |  |
| Figure 45 |  |  |
| Figure 46 |  |  |
| Figure 47 |  |  |
| Figure 48 |  |  |
| Figure 49 |  |  |
| Figure 50 |  |  |
| Figure 51 |  |  |
| Figure 52 |  |  |
| Figure 53 |  |  |
| Figure 54 |  |  |
| Figure 55 |  |  |
| Figure 56 |  |  |
| Figure 57 |  |  |
| Figure 58 |  |  |
| Figure 59 |  |  |
| Figure 60 |  |  |
| Figure 61 |  |  |
| Figure 62 |  |  |
| Figure 63 |  |  |

# Handover

## Software Review Report

### How team ensured quality of software?

Throughout the sprints we followed very closely to our standard of procedure. For development we made sure to make individual changes on various branches and we would then merge it to the current sprint branch before merging into the main branch.



When testing we made sure to follow our test cases which gets updated throughout our three sprints. Because we chose to do regression testing for every sprint we re-tested every functionality of our program to ensure that everything in previous sprints were still working.

### Processes utilized for adequate control of development process

As a team we made sure to have frequent meetings to check where everyone is at on the program. As mentioned before we would also work on various branches of our GIT and we would review changes before merging.

Another process we utilized was making sure we tested each other’s work to make sure that it had good a user experience and useability.

### Code testing process

For testing we made sure to test every function of our program for every sprint. The step goes as follows:

1. Update testing plans with additional new requirements from clients
2. Re-test every function
3. Take screenshots and update plan as we go
4. Sprint master combines everyone’s test into master document

### Future modification and refinements

When handing over our software we will ensure that our program is adequately commented and well documented. We will also provide help files such as documentations for where and how to modify the website.

### Criteria and Requirement Testing

|  |  |  |
| --- | --- | --- |
| Appearance | Yes | No |
| Good use of colour? |  |  |
| Is it pleasant to look at? |  |  |
| Is the structure properly organized? |  |  |
| Is there a header and a footer? |  |  |
| Is there a logo? |  |  |
| Is the navigation well designed? |  |  |

|  |  |  |
| --- | --- | --- |
| Functionality | Yes | No |
| The application functions correctly on multiple platforms and screen sizes. |  |  |
| Users can register / unregister for a newsletter or news updates. |  |  |
| Registration requires First name, Last name, and a valid email. |  |  |
| Application has password login system for administrators and Acme personnel |  |  |
| The password must pass a minimum complexity test. |  |  |
| The application has a rating system for the movies. |  |  |
| The ratings on the screen are updated as the ratings change. |  |  |
| The top ten page chart and table update as the ratings change. |  |  |
| Does the navigation work? |  |  |
| Can you search by any combination of movie criteria? |  |  |
| Does the application show all the movie’s details? |  |  |
| Does it have a Top 10 chart? |  |  |

|  |  |  |
| --- | --- | --- |
| User Experience | Yes | No |
| Is the application easy to use? |  |  |
| Is the text large enough to read comfortably? |  |  |
| Is the navigation and layout intuitive? |  |  |
| Is the design interesting? |  |  |

|  |  |  |
| --- | --- | --- |
| Responsiveness | Yes | No |
| Is the application visually appealing on mobile screens? |  |  |
| Does it still work on a mobile sized interface? |  |  |
| Is the application visually appealing on tablet screens? |  |  |
| Does it still work on a tablet sized interface? |  |  |
| Is the application visually appealing on desktop monitors? |  |  |
| Does it still work on desktop computer? |  |  |

|  |  |  |
| --- | --- | --- |
| Accessibility and Readability | Yes | No |
| Does the application work on different browsers? |  |  |
| Does the application work on different types of devices? |  |  |
| Does the text remain large enough to read on different screen sizes? |  |  |
| Do necessary images have alt text to explain what they are? |  |  |
| The application is perceivable using sight and/or sound. |  |  |
| The application is operable using just the keyboard. |  |  |
| The application structure and functionality are easy to comprehend and understand. |  |  |

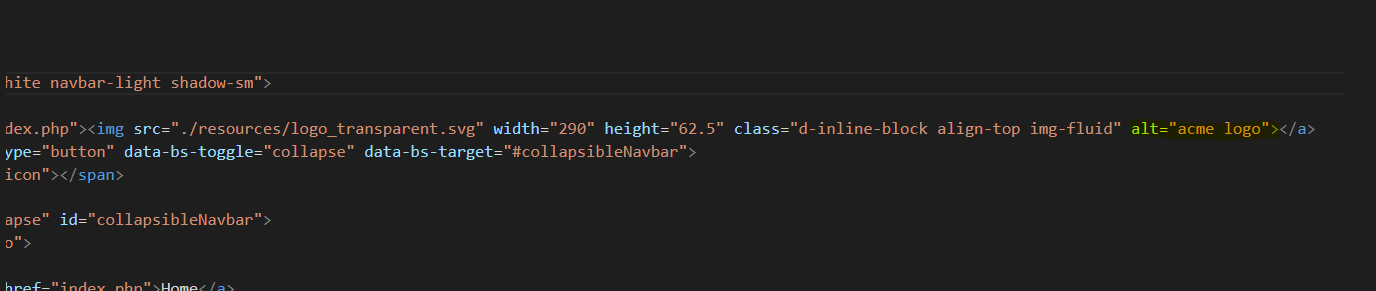
## Software Testing Plan

No further testing was required as client was satisfied with the all the functionality of the website.

## WCAG Requirements

To ensure that our website met WCAG requirements we made sure to include alt-text on our images, ensured that the website can be navigated with only the use of a keyboard and that every element and form on the page can be read by a screen reader.

In our case we used Microsoft Narrator and NVDA to test the screen reader to make sure that the info being read is accurate and met WCAG requirements.



# References

CITE MANAGED SERVICES. (2021). *CODING POLICIES AND STANDARDS*. Retrieved November 01, 2021, from citems: https://www.citems.com.au/services/application-development/standards/

CITE MANAGED SERVICES. (2021). *QUALITY MANAGEMENT*. Retrieved November 01, 2021, from citems: https://www.citems.com.au/services/application-development/quality-management/