# Testing Report

**Name:** Reeve Jarvis

**Project #:** 8

**Project title:** App Development

**Project Name:** GGDB - Good Game Database

## Testing Summary

Throughout the development of my final product, I aimed to achieve the best possible results for a wide audience. My goal was to produce a web-application that would:

* Pass standard code validation tests
* Be responsive to various devices
* Maintain accessibility for a wide audience
* Meet efficiency/speed test expectations
* Pass a Quality Assurance review

In order to meet this goal, I used various techniques and tools available to web developers. Below I have outlined the strategies I used to test my application, alongside the tools that assisted me in doing so, and the related reports. For a comprehensive list of testing reports I have included an Appendix at the end of the document with links to all external reports and resources.

## Testing Tool Sources/Documentation:

MDN DevTools: [What are browser developer tools? - Learn web development | MDN](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_are_browser_developer_tools)

Chrome DevTools: [Chrome DevTools](https://developer.chrome.com/docs/devtools/)

W3C - HTML Markup Validator: [The W3C Markup Validation Service](https://validator.w3.org/)

W3C - CSS Validator: [The W3C CSS Validation Service](https://jigsaw.w3.org/css-validator/)

Lighthouse Chrome Extension: [Lighthouse | Tools for Web Developers](https://developers.google.com/web/tools/lighthouse)

Lighthouse FireFox Extension: [Google Lighthouse – Get this Extension for 🦊 Firefox](https://addons.mozilla.org/en-CA/firefox/addon/google-lighthouse/)

PageSpeed Insights (Lighthouse): [PageSpeed Insights](https://pagespeed.web.dev/)

WAVE Accessibility Tool: [WAVE Web Accessibility Evaluation Tool](https://wave.webaim.org/)

ESLint NPM Package: [ESLint](https://eslint.org/)

## General Testing During Implementation

**Testing Tools Used:**

Browser DevTools

**Strategies Used:**

Throughout the implementation of my web application, I relied heavily upon the browser dev-tools to ensure everything was functioning as expected. As I created functions to handle the passing of data between my API sources and the page, I used console.log() calls to analyze the output and assist me in making programming decisions. I tracked error messages through the call stack of my console outputs, to assist me finding problematic code and developing solutions. Additionally, when experiencing issues I set breakpoints to navigate through the steps of my code and find the cause of my problems. The browser dev tools have become one of my most-used tools as a prospective web developer.

**Outcomes:**

Fully functional application across various devices.

## Quality Assurance Review

**Testing Tools Used:**

User Testing of Site Functionality

**Strategies Used:**

To ensure that my application functions as intended, I did a Quality Assurance check through user testing. To do so, I clicked through every possible link, ensuring they produced the expected results. I also did a check to make sure that all external resources were opened in new browser tabs to maintain a consistent user experience. Next, I performed some interactive tests on the search functionality with various inputs, to gauge the accuracy of results. I tried these activities on all of the devices I used for my Responsive Tests, to ensure functionality across multiple platforms. I also distributed my application to other users amongst my peers, to avoid user-bias.

**Outcomes:**

Positive results, with accurate response and expected data.

## Code Validation and Best Practices

**Testing Tools Used:**

ESLint Javascript Code Validation, Lighthouse Browser Extension, W3C Validators

**Strategies Used:**

I have done my best to follow best practices in the implementation of my JavaScript code. To assist me in doing so, I leveraged the ESLint NPM Package. ESLint provided real-time JavaScript validation and provided me with clear indication of code that could prove to be problematic or did not meet modern coding style best practices.

To validate the HTML and CSS code that I produced, or that was dynamically generated from my application, I used the W3C Code Validation tools. These tools are standard practice in web-development to ensure that all produced markup and style rules are valid. Reviewing my code with these validators allowed me to check for semantic code implementation, structural hierarchy issues, or unsupported CSS styles. If any of my HTML or CSS code was invalid, these reports gave me clear indication of the issues so I could fix my mistakes.

Upon final deployment of my application, I used the Lighthouse Browser Extension to continue validating my code. Lighthouse provides an in-depth report on various best practices used in your code. This tool will also be used for page speed insights in later tests.

**Outcomes:**

The final version of my application passed all validation tests with flying colours.

**W3C Validations:**

[Validate - CSS - Homepage.pdf](https://drive.google.com/file/d/1-vOFpCPXKh6gQeSssmG9zqIOcwx6i8_0/view?usp=sharing)

[Validate - CSS - GameDetailsPage.pdf](https://drive.google.com/file/d/1fa9CnHzoJk_1VtFPpOqj_Q9FipZOBGmy/view?usp=sharing)

[Validate - CSS - ProfilePage.pdf](https://drive.google.com/file/d/1gQ-eQ8m_SDpUwMMWozP60H3GilUvwxic/view?usp=sharing)

[Validate - HTML - Homepage.pdf](https://drive.google.com/file/d/1LSkRdEAiu1fgs9yx1g2bVttUv4s521Xs/view?usp=sharing)

[Validate - HTML - GameDetailsPage.pdf](https://drive.google.com/file/d/1_EQa0FTG1mKwtKQLnZUcFC9ki9E5za3_/view?usp=sharing)

[Validate - HTML - ProfilePage.pdf](https://drive.google.com/file/d/1tAKoaryyXn8w8Lfikh5IjUjG475ylDeJ/view?usp=sharing)

## Responsive Testing

**Testing Tools Used:**

Browser DevTools, User-Testing on various devices and browsers

**Strategies Used:**

Throughout the implementation stages, I worked to ensure that I coded my application structure using modern flexbox and css-grid layouts. On top of these layout techniques, I used responsive units to ensure page elements would adapt to screen size changes appropriately. This provided me with the framework to produce a responsive application with ease. I used the Browser DevTools device emulators throughout my coding stages to check that my application screens were displaying as intended along the way.

After deploying my application, I took the liberty to perform some various user-testing activities across a range of devices available to me. I was able to test my application’s appearance and functionality on the following devices and browsers:

* iOS (Mobile) - iPhone 8
* Android (Mobile) - Samsung Galaxy Note10+
* Android (Tablet) - Galaxy Tab A 8
* Desktop - Chrome Browser
* Desktop - FireFox Browser

**Outcomes:**

While there are some slight variations in background appearance between iOS and Android Devices, this is not any severe issue in functionality or presentation to page elements.

**Responsive Device Screenshots:**

[Responsive Testing Screenshot Slideshow](https://docs.google.com/presentation/d/18cwNek7Jpln_2sL0tRb7oJdZ2rjqDI0KvHAzmCjZuSM/edit?usp=sharing)

## Accessibility Testing

**Testing Tools Used:**

WAVE Accessibility Tool, Lighthouse Browser Extension

**Strategies Used:**

When considering the design of my application, I wanted to ensure I met modern accessibility standards. In order to achieve an accessible application, for a wide audience of varying abilities, there are a few key elements I focussed on:

* Semantic markup usage and proper page hierarchy
* Correct labelling and alt text for form fields, buttons, and images
* Appropriate color contrast between elements
* Font-selection with a focus on readability
* Clear and operable interface

For more in-depth detail on how I addressed these concerns see my [accessibility plan](https://docs.google.com/document/d/1Qm1WscEZdhvI8lohHO98kFCGqysQ8t2GSSKcy_WF4nI/edit?usp=sharing).

To test the effectiveness of my accessibility plans, I used the WAVE accessibility tool. This tool provided me with insights on how I might improve my page for a wider audience, and I made adjustments accordingly. On top of the WAVE tool, my Lighthouse Tests also performed a check for accessibility concerns.

**Outcomes:**

My page achieved perfect ratings on both the WAVE accessibility test, and Lighthouse Test respectively.

**WAVE Accessibility Reports:**

[AccessibilityTest - Homepage.PNG](https://drive.google.com/file/d/1vAWZrYJzjMqnQPEECM3X71gcOsVsqL_5/view?usp=sharing)

[AccessibilityTest - GameDetailsPage.PNG](https://drive.google.com/file/d/11RUwnPFRV7fhhNh7jv9SA8PyRRnPZtxF/view?usp=sharing)

[AccessibilityTest - ProfilePage.PNG](https://drive.google.com/file/d/1zTyxWvmJdgMrvVLWyYtplF6dPXD_nC9v/view?usp=sharing)

**Lighthouse Reports:**

[LighthouseReport - Homepage.pdf](https://drive.google.com/file/d/1JGLs-5QWxJ_xSTyiZ97fbzp9Sn6MveAb/view?usp=sharing)

[LighthouseReport - GameDetailsPage.pdf](https://drive.google.com/file/d/1iHMBweHD9VyCAWLL8aK9OOw-q-lUZFDE/view?usp=sharing)

[LighthouseReport - ProfilePage.pdf](https://drive.google.com/file/d/1I8GBihqINjkImCdVhVe92hEVBBD38hvU/view?usp=sharing)

## Page Speed / Efficiency Testing

**Testing Tools Used:**

Lighthouse Browser Extension

**Strategies Used:**

I have done my best to address page-speed and code efficiency throughout the coding of my application. I have built my application to populate page content and dynamically generate markup following modern practices, and tried to ensure fast processing times. The resulting page load is rather fast. There is a slight delay (1-2 seconds) for the page to display API data after load. I have worked to address this, and improve the overall user-experience using a Loading Animation overlay screen.

To test my deployed application for Speed and Efficiency I used the Lighthouse browser extension. This gave me a detailed report, and pointed out issues in need of addressing. There were some limitations to what I was able to achieve based on the API resources I am serving to my page, particularly when it comes to images. With that said, my tests presented positive results. There are a few items I will plan to address in future updates, to improve the overall speed and quality of my application as best I can.

**Outcomes:**

Page speed tests reported as mid-high level, 87/100.

**Lighthouse Reports:**

[LighthouseReport - Homepage.pdf](https://drive.google.com/file/d/1JGLs-5QWxJ_xSTyiZ97fbzp9Sn6MveAb/view?usp=sharing)

[LighthouseReport - GameDetailsPage.pdf](https://drive.google.com/file/d/1iHMBweHD9VyCAWLL8aK9OOw-q-lUZFDE/view?usp=sharing)

[LighthouseReport - ProfilePage.pdf](https://drive.google.com/file/d/1I8GBihqINjkImCdVhVe92hEVBBD38hvU/view?usp=sharing)

# APPENDIX:

## Testing Tool Sources/Documentation:

MDN DevTools: [What are browser developer tools? - Learn web development | MDN](https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_are_browser_developer_tools)

Chrome DevTools: [Chrome DevTools](https://developer.chrome.com/docs/devtools/)

W3C - HTML Markup Validator: [The W3C Markup Validation Service](https://validator.w3.org/)

W3C - CSS Validator: [The W3C CSS Validation Service](https://jigsaw.w3.org/css-validator/)

Lighthouse Chrome Extension: [Lighthouse | Tools for Web Developers](https://developers.google.com/web/tools/lighthouse)

Lighthouse FireFox Extension: [Google Lighthouse – Get this Extension for 🦊 Firefox](https://addons.mozilla.org/en-CA/firefox/addon/google-lighthouse/)

PageSpeed Insights (Lighthouse): [PageSpeed Insights](https://pagespeed.web.dev/)

WAVE Accessibility Tool: [WAVE Web Accessibility Evaluation Tool](https://wave.webaim.org/)

ESLint NPM Package: [ESLint](https://eslint.org/)

## W3C Validations:

[Validate - CSS - Homepage.pdf](https://drive.google.com/file/d/1-vOFpCPXKh6gQeSssmG9zqIOcwx6i8_0/view?usp=sharing)

[Validate - CSS - GameDetailsPage.pdf](https://drive.google.com/file/d/1fa9CnHzoJk_1VtFPpOqj_Q9FipZOBGmy/view?usp=sharing)

[Validate - CSS - ProfilePage.pdf](https://drive.google.com/file/d/1gQ-eQ8m_SDpUwMMWozP60H3GilUvwxic/view?usp=sharing)

[Validate - HTML - Homepage.pdf](https://drive.google.com/file/d/1LSkRdEAiu1fgs9yx1g2bVttUv4s521Xs/view?usp=sharing)

[Validate - HTML - GameDetailsPage.pdf](https://drive.google.com/file/d/1_EQa0FTG1mKwtKQLnZUcFC9ki9E5za3_/view?usp=sharing)

[Validate - HTML - ProfilePage.pdf](https://drive.google.com/file/d/1tAKoaryyXn8w8Lfikh5IjUjG475ylDeJ/view?usp=sharing)

## Lighthouse Reports:

[LighthouseReport - Homepage.pdf](https://drive.google.com/file/d/1JGLs-5QWxJ_xSTyiZ97fbzp9Sn6MveAb/view?usp=sharing)

[LighthouseReport - GameDetailsPage.pdf](https://drive.google.com/file/d/1iHMBweHD9VyCAWLL8aK9OOw-q-lUZFDE/view?usp=sharing)

[LighthouseReport - ProfilePage.pdf](https://drive.google.com/file/d/1I8GBihqINjkImCdVhVe92hEVBBD38hvU/view?usp=sharing)

## Responsive Device Screenshots:

[Responsive Testing Screenshot Slideshow](https://docs.google.com/presentation/d/18cwNek7Jpln_2sL0tRb7oJdZ2rjqDI0KvHAzmCjZuSM/edit?usp=sharing)

## WAVE Accessibility Reports:

[AccessibilityTest - Homepage.PNG](https://drive.google.com/file/d/1vAWZrYJzjMqnQPEECM3X71gcOsVsqL_5/view?usp=sharing)

[AccessibilityTest - GameDetailsPage.PNG](https://drive.google.com/file/d/11RUwnPFRV7fhhNh7jv9SA8PyRRnPZtxF/view?usp=sharing)

[AccessibilityTest - ProfilePage.PNG](https://drive.google.com/file/d/1zTyxWvmJdgMrvVLWyYtplF6dPXD_nC9v/view?usp=sharing)