This document will instruct and direct you to complete all of the evidence you will need to meet the Achieved level criteria for the following standard.

| **Number** | **Version** | **Title** | **Credits** | **Assessment** |
| --- | --- | --- | --- | --- |
| AS91893 | 1 | Use advanced techniques to develop a digital media outcome | 4 | Internal |
| **Achievement Level Statement** | | | | |
| Use advanced techniques to develop a refined digital media outcome. | | | | |

Please enter the requested evidence in the areas provided.

**3.1 Using information from testing procedures to improve the quality and functionality of the outcome**

Previous Testing (your observations)

The testing from stage 2 will have highlighted areas of the solution that could be developed further. Use the table below to identify at least 3 of these possible changes that will consider making. (you can add more rows if you think you need them).

|  |  |  |
| --- | --- | --- |
| **#** | **What change will you make?** | **Why do you think this is needed** |
| 1 | Instead of mailto link, use form with SQL Database | Form is easier for users to just enterf their details click submit and theyre done instead of writing a whole email + its ezier to organise database then 100s of emails |
| 2 | Make particles appear after making window larger | QOL |
| 3 | Do Colours better | So blind people can see better |
| 4 | Add in-page buttons to link pages | To make the website feel like it links together more |

User Feedback (other observations)

Get two people to use version 2 of your solution and ask them to list 3 changes each that they think would improve it.

|  |  |  |
| --- | --- | --- |
| **#** | **Person 1 – Name : James Wright** | **Person 2 – Name : Matthew Currie** |
| 1 | Replace mail to link with form to make it easier for users to register | Use more different colours for colourful text instead of all the same colours |
| 2 | Don’t have random stuff on details page that is just copyed from contact page or have link to contact page | Add heading to rules page |
| 3 | Add more images on other pages not just the first page | Make navbar text bigger/more readable |

**3.2 The Changes Made**

Now that you have identified a range of changes you **could** focus on, select at least 3 that you **will** focus on and try to implement them.

Remember to copy the entire solution before making changes so that you do not delete the original versions!

Once attempted, complete the table below to indicate what you have done and how successful you believe you have been. (you can add more rows if you think you need them).

|  |  |  |
| --- | --- | --- |
| **#** | **What change did you attempt?**  **How did you do this?** | **How successful were you?**  **Why is this change better?** |
| 1 | Add in-page buttons to link pages  I just added buttons wrapped in anchor elements | Very successful  It hints to users where they should go next |
| 2 | Instead of mailto link, use form with SQL Database  I moved server to apache becuz the live server extension for vscode don’t support html forms and the $\_POST php var  Then I just HTML form and simple SQL table | Very successful  Form is easier for users to just enterf their details click submit and theyre done instead of writing a whole email + its ezier to organise database then 100s of emails. Admins can also go to ./registrations.php to view all registrations |
| 3 | Make particles appear after making window larger by dynamically adding them in the onresize handler as neccesary | It adds to the aesthetic of the website |
| 4 | Also worked on performance of mousemove event handler as it had the largest effect on performance due to it forcing the entire DOM tree to have to recalculate CSS styles as mose position was previously stored in CSS variables on the <body> element | Moved mouse position CSS variables to only be applied to elements that need them, and to not be inherited so the CSS parser doesn’t have to update child nodes. Now performance: (Note only few red flags) |

# 3.3 Using efficient tools and techniques in the outcome’s production.

In the table below identify the advanced tools / techniques that you believe you have used efficiently and explain why this is the case.

It is quite possible that your explanations will be large in size as an efficient use likely means that a number of areas have been improved, and you will need to identify these to highlight any efficiency you have gained.

|  |  |
| --- | --- |
| Advanced Tool | How has your use of this been efficient? |
| *e.g. Linked CSS documents* | This allows for the same styles to be used across multiple similar-themed pages without this there will have been a lot of duplication of the CSS styles. Which would mean that the development process would have taken a lot longer, additionally now the pages load faster because the browser can cache the CSS stylesheet resource and knows to use this whenever one of the other pages on the site requests it, saving a lot of bandwidth (13KB per page load in my case). Etc. |
| PHP | This allowed for template content (such as the navbar, which wasn’t identical on all pages, but the minor changes (activating the currently selected page) could be automated with php; as well as the carousels, which contained many identical image elements, but with different urls) to be dynamically generated on the server-side, before sending fully-compiled html files to the client. It pretty much acted as a preprocessor for HTML in these cases. I also used PHP for communication with a MySQL database, as well as authentication of the basic webpage used to view this database (phpMyAdmin wasn’t working for my XAMPP install??? idk). Without this, the process for the navs and carousels would have been very much copy-paste, and connectiong the HTML form to the SQL DB would have proved impossible. |
| Batch image processing | This allows for easy management over many image files, to process them for more efficient use. In my case, I used it to reduce the insanely high resolution of the provided images, making them more suitable for web. Without this the image files would have been needlessly high resolution, meaning massive file sizes, and enormous bandwidth usage over the network. It also affects the browser paint time, as the GPU has to copy (blit) the image onto the screen, reading from the texture file and downscaling as necessary. This process can be quite expensive for large images (I remember before doing this that when scrolling across the carousel you could visibly see it pause for a second as the browser has to read the file from (VRAM?) and paint it to the screen). Now, the images are loaded across the network much faster, and painted much faster, making the website much more useable. |
| SQL DB | This allowed for storage and querying of the submitted form data. My solution does not yet use this to it’s full potential; SQL is a VERY advanced and capable language, and many more features could have been added to make navigating the form results easier. However the current solution of storing the form results and then displaying them all for the school to process is plenty enough for the needs of the formal organisers. Without this, I would not have had an easy way to store the form results for the event organisers (shush google forms looks ugly cuz cant style to the same theme). |
| Javascript | This allows for the website to do much more dynamic and functional effects on the client side. I used this to dynamically generate random star particles to fit the space theme, as well as dynamically modifying the page, especially on events like resize, where sometimes small adjustments are needed to be made to make the page fit the new screen layout. It also allows for elements like buttons to have actual functionality associated with them, for example the carousel buttons which scroll the carousel over to the previous or next image. Without this, some of the more fun, appealing and interesting effects of the website would not have been possible. |
| Performance | We like it when our client runs smoothly. Javascript is slow, but I have tried my best, where possible, to minimize the performance hit of effects. Small things like caching the list of elements who need updated on mousemove (line 277 index.js) and using document fragments as an intermediary to appending to the DOM (line 109) can edrastically reduce the work the browser has to do querying and updating the DOM, which can be very expensive, especially when hundreds of elements encounter potential reflows or calculated style changes. Without this, the webpage would be much less smooth, drastically impacting UX. |

**3.4 Refined Testing**

You now need to test the 3rd version of your solution. This time we need to focus more on the accuracy of what you created so we will need to complete a formal test plan. Create, Predict and Apply a range of tests that show your solution works correctly. A real-world test plan will likely have 1000s of test in it, but in this instance we only need a handful for each element you created (20-30 in total). **Extra Help Sheet Available**

Also I want to comment on this “A real-world test plan will likely have 1000s of test in it” because a real world test plan would probably be automated with a test library like JUnit or within some kind of Github Actions or smth.. no one writes thousands of lines of tests in a table in a word doc anymore..

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test No. | Test (include test data if necessary) | Expected Result | Actual Result | Test Result |
| 1 | Input test data in form | Should appear in SQL DB | <- | Pass |
| 2 | Input invalid chars/symbols into name, email, or form class input | Should show error message on submit | <- | Pass |
| 3 | Leave formclass or name box empty | Should not allow submit | <- | Pass |
| 4 | Name, formclass, email input box lengths | Should not let you type any more than 30, 5, 50 chars, respectivly | <- | Pass |
| 5 | Hacking to bypass clientside maxlength or required limits | Shoulod show an error after submitting form and not add to db | <- | Pass |
| 6 | Particles | Should regenerate after resizing window to be bigger | <- | Pass |
| 7 | Submit buttons and button links | Should change bg color on hover | <- | Pass |
| 8 | Submit buttons and button links | Should move box shadow while clicking | <- | Pass |
| 9 | failed attempt when submitting form | All form values should be set to what they were before submit | <- | Pass |
| 10 | Spin animation on particles | Should not be applied to point stars (they are fully rotationally symmetric) for performance | <- | Pass |
| 11 | handleResize function of ParticleManager | Should be called at most every 300ms while resizing | <- | Pass |
| 12 | New particles while resizxing larger | Should only be created in places where no particles are already | <- | Pass |
| 13 | Background image | Should always cover full screen | <- | Pass |
| 14 | Form input elements | Should have red shadow if required and empty, or has other error | <- | Pass |
| 15 | Particles | Should never be too many on screen (< (window.innerWidth - 800) / 6) | <- | Pass |
| 16 | Registrations DB VIEWER page | Should only be accessible with correct username/password (admin/password123) | <- | Pass |
| 17 | Registrations DB VIEWER page | Wrong uname/pwd should retry or show error, not leak contents | <- | Pass |
| 18 | Registrations DB VIEWER page  Log out button | Should logout, sent to index page, and request login when next accessed | <- | Pass |
| 19 | Registrations DB VIEWER page | Should display table with all records in DB  (email records should show with mailto link) | <- | Pass |
| 20 | Post submittion of the form | Should take user to success page | <- | Pass |
| 21 | Keyboard Navigation through form | Should flow as expected | <- | Pass |

# 3.5 The Evidence (Video Recording v3)

Create a 3rd video recording showing your solution in operation. Show the full operation of the solution as before, but be sure to spend time demonstrating the changes you made.

Again, if your chosen technology has any form of validation service available (e.g. html validation) demonstrate this in use as well to both show how effective your use has been and to highlight any areas you may need to develop further.

Take your time and give the viewer of this video and opportunity to fully see your database in action. Save your video recording in a suitable file format (e.g. wmv or mp4 – NOT an ispring file)

|  |
| --- |
| **Below tell us the name of this file and where it is stored.** |
| V3.mp4 in H:/2023/Media/Assessment/html2023/docs  Or maybe on github  Or maybe inside zip handin |