Sorting Algorithm – Design and Implementation

1. Introduction

1.1 Product Scope  
Who this project is aimed at. Includes a general PACT and two specific Personas.

1.2 Goals and Requirements  
Addresses the following goals and functional requirements for designing a multimedia web application. Demonstrates and explains the implementation of Search Algorithms within systems.

2. Design Overview

2.1 Wireframes  
Wireframes of each web page and how this design satisfies the needs of the personas and scenarios

2.2 Navigation Maps  
Navigation Maps of each web page to show they interact and interconnect with one another and how this design satisfies the needs of the personas and scenarios.

3. Prototype

4. Testing

5. Sources

# Introduction

This design project is a HTML+CSS web page that aims to demonstrate and explain how sorting algorithms work within computational systems. The project is to contain JavaScript animations and images to provide a colourful and easy-to-understand design for the user. The aim of this project is to provide an easily accessible and understandable visual representation of a software concept by representing the problem with conceptual visual cues for the user and then relating them to the relevant practical code.

The user will be able to select a search algorithm of their choosing using clickable buttons on the web page to navigate to different pages. They will be shown JavaScript animations of the search algorithm where they can interact with the web page by selecting the values of the example to be searched from and which value they want to find.

## Product Scope

This project is intended for educational purposes to be used to teach users who want to learn about Computing algorithms.

The target users have been identified and evaluated using High-Level Design Theories such as PACT (People, Activities, Contexts, Technologies). Two likely Personas that can be used are Computing Science students at University and the Computing Science lecturers.

* **P**eople – Users are most likely Computing Science students/teachers who are looking to educate themselves/others about computational sorting algorithms. They most likely have a basic understanding of computer systems and will possess a mental model that is more sophisticated than that of the average person. It is assumed that they have an understanding of the code provided in the web page and how a program is typically structured, as a result there is no information provided on the basic principles of programming or code structure. Students/lecturers may not be from an English-speaking country so language options will be present at the start of the webpage to accommodate those who do not speak English or prefer another language.
* **A**ctivities – Users who are viewing the webpages are most likely sat down at their computer when interacting with the webpage and are focussed on the information given (it is not to be used in conjunction with another software).
  + Temporal Aspects –
    - Students (for example) may visit this web page several times across their learning period as a revision strategy, whereas teachers/lecturers may revisit this website every year to familiarise themselves with the concepts and use it as a teaching resource. As a result, the web page needs to be easily accessible and quick to learn as the tasks on the web page may be used infrequently
    - The project is designed to be flexible with any time pressures the user may have. It will have a built-in slider that will speed up or slow down the animation times and will also have summaries of pieces of text to accommodate those who are pressured by time but also provide the information and context for users who are willing to take the time to digest the information given
    - The simplistic and easy-to-use design of the web page ensure that users who are interrupted during their interaction with the web page can easily go back to where they were in just a few clicks - this is an advantage of having a low amount of navigation within the web page and also having a “home” page where every navigation button is displayed
  + Cooperation –
    - The project does not need to be run in conjunction with any other piece of software but must be accessed by a sufficient connection to the internet, nor does it need to be used by more than one user at once to operate as it is designed for one sole user to explore themselves
  + Complexity –
    - Some concepts within the webpage may be considered vague or complex but the availability of customisation and essentially “playing around” with the webpage allows the users to explore and ultimately understand the concepts a lot better. An example of this is the sorting algorithms themselves, even though the user may not understand the description or explanation of the algorithm, being able to choose their own inputs to see how each algorithm reacts will help with their general understanding of the problem.
  + Safety-Critical –
    - Measures are put in place in the webpage for when the user puts unexpected values for the inputs, the JavaScript will output a message telling the user where they went wrong and will ask them to try again. At no point should the webpage produce an error message when an unknown value is inputted.
* **C**ontexts –
  + Physical Environment – The activity is most likely to occur in the classroom or in any educational environment where it is being demonstrated to a group of pupils or through individual learning. This requires the webpage to be clearly readable and understandable to be used in the classroom setting. The webpage should not require any documentation or explanation to how it works and the mental model of most users will be detailed to an extent on how the webpage operates (they may understand how the navigation of the site works and what the buttons do but not necessarily how the JavaScript is implemented).
  + Social Context – The webpage will not breach any private information about the user as they will have access to all the material without having to answer any questions or sign in. This means that the site can be accessed in public and at home without any concern for their privacy. There will contain playable videos on the site that will contain audio but will not automatically play until the user clicks on the video. This ensures that the use of the site is accepted by social norms as there is no unexpected audio or video playing when in use.
* **T**echnology –
  + Input – User is most likely to be using a home/school computer, in which they will be using a mouse to interact with the web page by clicking on links and buttons within the webpage. If the user wishes to customise the values inputted into the sorting algorithm, then the user will require a physical or virtual keyboard to output decimal values to the system. Future versions of this project may aspire to be accessible on mobile systems such as phones or tablets, in which the user will be able to input using the in-built keyboard and touch screen functions.
  + Output – All output will be through the user’s monitor display and with the occasional integrated video playing audio through speakers/headphones. The webpage will make the use of JavaScript visual animations and lines of code to show the methodology and output of each sorting algorithm.

## Personas

**Nikolas**

* + - * Age 19
      * 2nd year University student studying Computing Science
      * Part-time job at Morrison’s
      * Plays video games and writes code in his free time
      * Youngest child
      * Leaves assignments until the last minute
      * Transfer student originally from Poland

1. Studying for his assignment about sorting algorithms
2. Given a task by his lecturer to implement a sorting algorithm into his code but has left it until the day before it is due
3. Has not caught up on lecture slides and does not have time to take notes of them all
4. Frantically searches Google to look for code that he can copy and paste
5. Finds code but it does not implement correctly, and he does not know why
6. After more Googling Nikolas stumbles across the Sorting Algorithm Design project and scans that page
7. Sees the sorting algorithm he is looking for in a bold button and clicks
8. He is taken through an animation that tells him how each step of the sorting algorithm works
9. He reads on and changes the input values to somewhat match those of his assignment
10. Suddenly the sorting algorithm makes sense, and he does not have to copy and paste the code and it works perfectly after some adjustments
11. He receives 72% for his assignment grade. He is relieved to have finished and done well

**John**

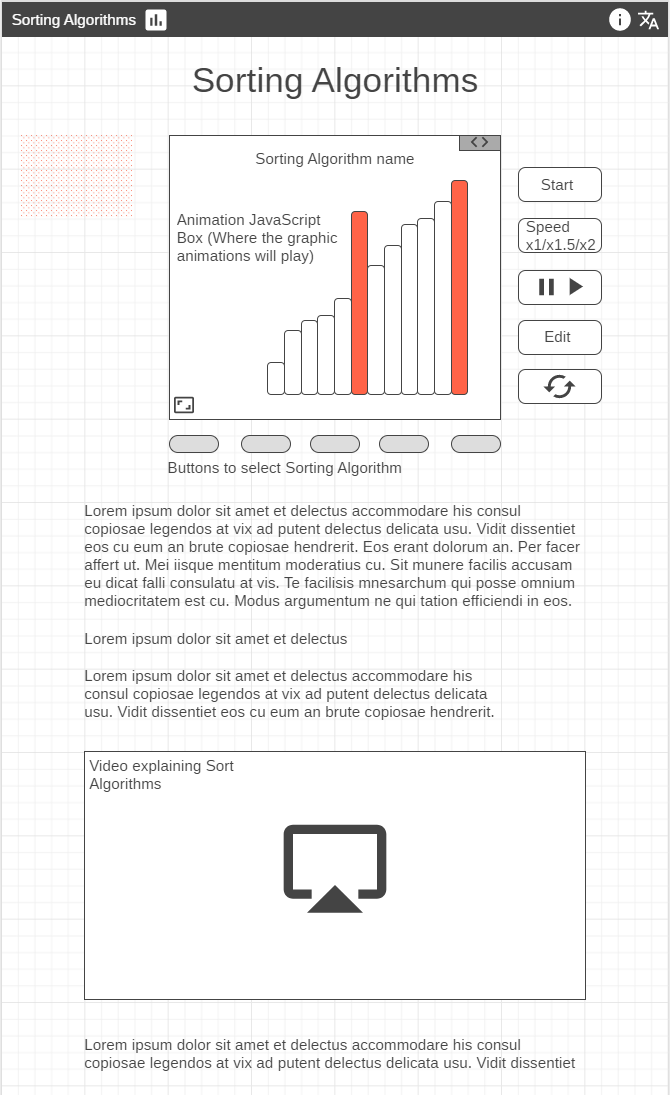
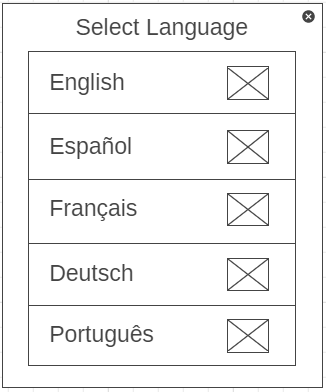
* + - * Age 47
      * Computing Science teacher at a High School
      * Lives with his wife and two children
      * Returning to work after a 4-month sick leave
      * Does not have a lot of time to come up with a lesson plan before term starts

1. Sitting in his workplace the night before term starts and he is trying to finish the last of his lesson plans
2. Working on his sorting algorithms topic and is struggling to find the time to explain every algorithm in his class
3. Recommended a website by one of his colleagues in the department, Sorting Algorithm Design
4. Thinks it is fantastic. Provides a lot of information and allows the user to decide the inputs for each sorting algorithm
5. Assigns 10 minutes of his lesson to go briefly go over the website and then set it as homework to learn
6. Teaches the pupils how each algorithm works and inputs his own values into the algorithm. Asks the students what they think will happen and to write it down
7. By the end of the lesson most students either understand or are beginning to grasp the concept of sorting algorithms and next week they will be made to implement them

# Design Overview

## Wireframe

**Home Page**

Once the user opens the webpage, they are greeted with this home page. At the top of the webpage is the banner that is displayed on every page of the website and acts as an anchor for navigating the website.  
Clicking on  at the top bring the user back to the Home Page.  
The  icon in the top-right corner brings up a window detailing the information about the webpage (e.g. author, date created, date updated) that can be closed by an ‘x’ on the corner of the window.  
The  icon in the top-right corner brings up another window that lets the user select a language. (Images will be filled with the corresponding Flag of each country)  


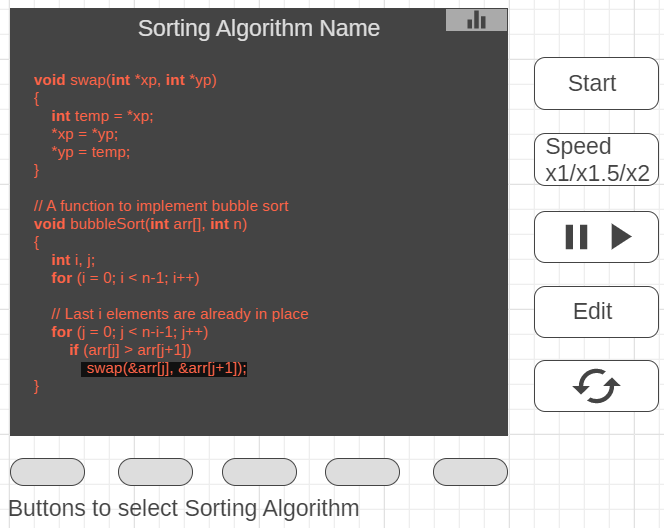
The central box contains the JavaScript multimedia that plays the animation for each sorting algorithm. The coloured bars seen here are the current bars being sorted in the algorithm and to the left of the graph will be a description of what is happening at this current stage.

The small grey box in the corner of the JavaScript box  is used to switch to the code view of the problem

Justifications:

* Holding the in-depth information that requires a lot of time to read over and understand below the animations and short explanations means that the users’ attention is drawn to the quicker information (e.g. short animations and summarised explanations) so that the user’s attention is grabbed and they can get a quick idea of how each algorithm operates without having to go in-depth.
* The text below this is acting as additional or more thorough information if the user wants to read but they will understand the basics from the animations alone. This prevents the user from being overwhelmed by a wall of information that will instantly turn them off from the website. Instead, a colourful animation is displayed and the short descriptions of each step going along side it. Users will want to consume this bite-size pieces of information before they decide they want to learn more about the algorithm.
* The buttons to the right of the animation are carefully sequence in order of most likely to be used and importance. For example, the most likely button to be used by the user is the ‘Start’ button as for every sorting algorithm they will need to press this button to activate the animation and so on. This is for the user’s benefit as they have clear access to the button, they are looking for without them knowing. Without having to look at the other buttons, the user will scan from the top and see the ‘Start’ button which will be the next button they were requiring if they are new to the site. After their initial scan and a run of the animation, the user will be familiar with how the animations work and the consequent buttons and where to find them.
* The ‘Code View’ button in the top right of the animation box may appear small and hard to find but has been coloured grey to contrast with the bright orange and white background. This may be changed to have its own larger in later iterations if users find this difficult to navigate
* Metaphors are used throughout the webpage to represent buttons and functions. The symbols used are default object that are used in many other websites and applications. The metaphors used will be understood by users as their personas dictate that they are very familiar with design in technology and have a vast experience of using websites and applications that will use the same symbols. These are used to describe the domain of each button to help the readability of the website and help for when the user is scanning to find what they are looking for. Buttons that do not contain a metaphor are done so because a metaphor cannot be found for the domain it is trying to describe (e.g. Speed – no images represent x1/x2 without writing it in text).
* Many affordances are used throughout this project in the form of buttons and the navigation bar.
  + Buttons help to suggest to the user that they can click each button and they will have an effect. Buttons are outlined in black and may contain either a white or grey filling to help distinguish them and tell the user that squares with this design are most likely buttons that can be pressed
  + The navigation bar uses white symbols on a black background to distinguish what is clickable. Everything in white on the black navigation bar is clickable and will take the user to a different page
* An embedded YouTube video is included in the information section of each sorting algorithm page to provide an auditory explanation of each algorithm for those who find it easier to consume information through sound and to provide more information that has not already been covered. This is to ensure that all learning techniques of the user are considered – whether it be through text in the information section, visuals through the animations and code, auditory through the embedded video, or through kinesthetics where the user gets to customise and play with the different inputs values of each sorting algorithm. YouTube video will also include built-in captions for users who are hard of hearing or cannot understand the speaker.
* A Fullscreen button is provided in the bottom left-hand corner of the animation box to expand the view of the animation in a bigger display. This is used for the inclusion of people who may be visually impaired or struggle to see smaller shapes. The contrasting orange colour used for the currently active bars in the graph are used to help the user easily distinguish and keep track of the value that is being sorted and where it moves to on the graph

**Code View**

In the code view, the corresponding code for the sorting algorithm is displayed in orange text.   
Once the animation is played, a black box will appear over the line of code that is currently being executed and a description of what that line does to the right of the code. This allows the user to follow along with the code as the algorithm sorts the data. The user can also simultaneously switch between code view and grid view to gather a full understanding of what is happening at each iteration of the algorithm.

The media buttons on the right (e.g. Start, Speed, Pause, etc) still operate as normal in the code view as it would in the grid view. Fullscreen button is functional in both the grid view and the code view – when pressed will bring up another window showing a larger version of either the grid or lines of code.

Justifications:

* Code text is written in bright orange to contrast with the dark background to make the code readable from afar
* Black box highlighting the currently running piece of code allows the user to follow along with each iteration of the algorithm and to easily distinguish, at a glance, which line of code is currently under execution.
* The Code View is used to provide the user with practical examples of how each algorithm would be implemented in code. Each line will be explained and discussed next to the code so that the user has a full understanding of the practical implementation of the algorithm.

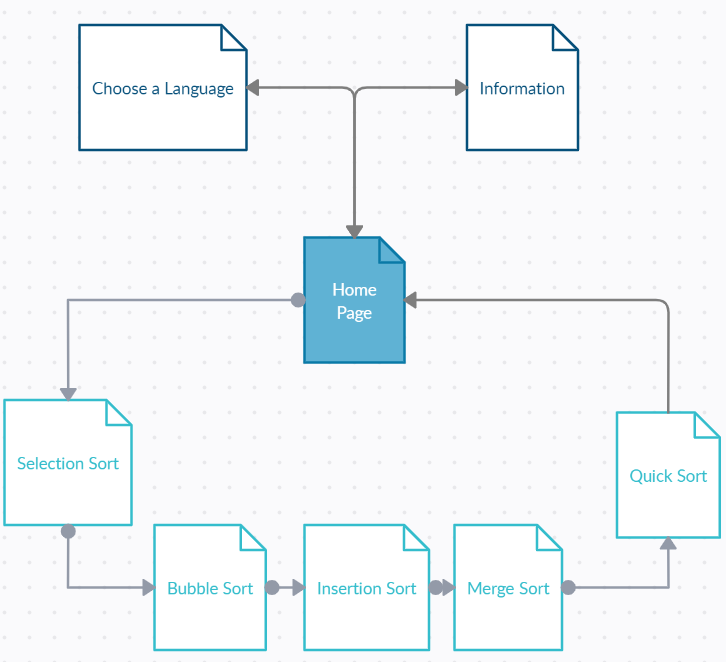
**Media Buttons**

* Start – Starts the sorting algorithm with the default data or the data set by the user. Starts at step 1 and goes through the sorting process until completion
* Speed – Different settings for speed. The button will show only one number in the box and that will be the current speed. Every time the button is pressed it will cycle (e.g. currently shows x1, after clicking it shows x1.5) and the speed of the sorting algorithm will change accordingly. The default speed at x1 may be a 300-millisecond delay between animation changes, x1.5 being 225 milliseconds and x2 at 150. Only three speed settings are chosen because any faster than x2 is unnecessary as you won’t be able to see what is happening in the sorting algorithm and any slower than x1 the user can easily pause the animation at any time they wish to read and understand what is happening.
* Pause/Play – Similar to the speed buttons, only one will display depending on whether the animation is currently running or paused and will change once clicked
* Edit – Allows the user to edit the values of the sorting array. Once clicked, a window will appear displaying each bar and their corresponding number underneath it. Numbers are between 1-100. As the user changes the number the bar above will change in size dynamically to provide the user with a visual representation of what each value means compared to other bars. Input validation will be used to ensure the user cannot input any unexpected characters and can only pick a number within the range. Allowing the user to pick their own values provides a better understanding of each algorithm as they test each algorithm with trial and error to see how each handles the given values
* Restart – Once clicked, animation will cancel and return to step 1 where it will replay with the same values. This is so the user can go back to the beginning of the sorting algorithm without having to refresh the website or wait until the algorithm finishes its last step

**Text and Video**

Paragraphs under the animation are used to provide further general information about the sorting algorithm selected. This will provide more detailed information that the user can read along side the animation and will explain things like Time and Space Complexity. Information outside of the JavaScript animation is useful as the user does not have to either read the code or run through the animation to find the segmented pieces of information – it is all together where the user can read.  
The video is an embedded YouTube video that can be played by the user. It contains a verbal explanation of each algorithm and displays images and animation alongside it.

## Navigation Map

This project uses a Composition style of navigation where every page within the project is accessible from each other (for the exception of the Language page and the Information page which act as individual pages providing critical information).

All other pages are accessible through their “Sorting Algorithm” buttons below the JavaScript animations where any sort can be accessed by those 5 buttons.

Every page within the project can access the “Home Page” at any time using the banner at the top of every page with the logo and “Sorting Algorithms”.

The most likely order of navigation that the user may take is:

1. Home Page – Once the user opens the webpage, the default page is the Home Page
2. The Language page if the user is not happy with the currently selected language and wants to change it
3. Selection Sort – First button from the left. Naturally, the user is most likely to click the left-most button as we read left to right in Western societies unless they are looking for a specific sorting algorithm in which case it is unpredictable
4. From Left to right: Bubble, Insertion, Merge, Quick
5. Information page: Metaphor for Information page is not entirely clear so users may click on this to see what it is. Likely to be the last thing a user clicks on or not at all on the webpage

The project uses a Composition style to provide a simple navigation system where any other page can be reached by each other. This prevents the pages from being hierarchical which means the user would have to navigate through a series of pages before they reach their desired page. This also means that if a user is looking to find a specific sorting algorithm, the button linking it will be one of the first things their eyes lay upon when scanning the website. This is done by focussing the colour onto the animation and to the buttons either side of it so it most likely where the user will look for information.

Having all links be in the same place for every page provides a consistent and easy-to-understand design that visually shows which page you are on by greying out the sort algorithm button being displayed.

Each page within the webpage (the exception of Language and Information) are rewritten versions of the same page. Each have the same layout and the same design just with different contents of information. Therefore, it is useful to have every page link to another as they can seamlessly transition from one page to another with the click of a button. This also helps to alleviate any confusion that might occur when trying to navigate a hierarchical system by eliminating the need for breadcrumbs and having it all accessible from any page.

# Prototype Description

The prototype version of the project includes some of the features discussed but is still missing a proportion of the final product.

The prototype includes every page that would be present in the final product, but the contents and designs of the page are not accurate. Only one of the sorting algorithms has a working animation but is missing the media keys to control and manipulate the data. So instead, the algorithm is given a random set of values every time the page is refreshed to provide a wide set of testing scenarios to ensure that it works with any value given.

The prototype includes the navigation bar as it would look in the final product and works as it should. This is ready for testing and for users to give their feedback about.

Placer text, videos and pictures have been implemented instead of relevant media just to test the formatting and the loading times of the media. Videos are particularly long to test how long websites take to load with them even if the read media will have a shorter runtime.

That sorting animation is present but is missing the code view due to time restrictions. Only the animation needed to be implemented to test the functionality of the JavaScript box and to give an idea of the final design of the page.

Only one animation is provided as all other pages will work exactly the same so multiple animations were not necessary but will be implemented for the final product.

# Usability Testing

## Independent Usability Testing

Accessibility

1. Pages load within reasonable time
2. Text is easily readable with contrasting colours
3. Text/spacing is not too small
4. Flash and add-ons are not overused
5. Images have ALT Tags
6. Site does not produce error when first loaded
7. Embedded video has captions
8. Metaphors are clear to the user
9. Text is translated correctly

Identity

1. Logo is visible on every page
2. Homepage is not overwhelming
3. Easy to change language
4. Easy to access Information page

Navigation

1. Navigation bar is easy to identify
2. Buttons are identifiable
3. Number of buttons is reasonable
4. Links go to where they are supposed to
5. Buttons have accurate text

Content

1. Titles are clear and concise
2. Bite-size and important content is displayed first
3. Style and design are consistent across pages
4. The design is not over emphasised using bold
5. No ads or pop-ups
6. Information is clear and easy to understand
7. URLs are user-friendly
8. HTML titles are user-friendly

|  |  |  |
| --- | --- | --- |
| Test | Does it work? | Feedback |
| 1. Pages load within reasonable time | Yes | Pages load without any issues |
| 1. Text is easily readable with contrasting colours | Yes | Black text on white background and vice versa. Text is never unreadable |
| 1. Text/spacing is not too small | Yes | Text is a good size and is readable |
| 1. Flash and add-ons are not overused | Yes | No flash or add-ons. Only the JavaScript animation per page |
| 1. Images have ALT Tags | Yes | Filler images have ATL tags |
| 1. Site does not produce error when first loaded | Yes | No errors occur |
| 1. Embedded video has captions | Yes | Autogenerated by YouTube |
| 1. Metaphors are clear to the user | Yes | Allows for easy navigation and are commonly recognised metaphors |
| 1. Text is translated correctly | N/A | Translation function not yet implemented |
| 1. Logo is visible on every page | Yes | Visible via navigation bar |
| 1. Homepage is not overwhelming | Yes | Currently Home Page is simplistic |
| 1. Easy to change language | N/A | Not yet implemented |
| 1. Easy to access Information page | Yes | Via Navigation Bar |
| 1. Navigation bar is easy to identify | Yes | Always at top of page and distinguishable |
| 1. Buttons are identifiable | N/A | Buttons not yet implemented |
| 1. Number of buttons is reasonable | N/A | Buttons not yet implemented |
| 1. Links go to where they are supposed to | Yes | All navigation links work |
| 1. Buttons have accurate text | Yes | Self-explanatory text |
| 1. Titles are clear and concise | Yes | Self-explanatory titles |
| 1. Bite-size and important content is displayed first | Yes | Images, animations played first |
| 1. Style and design are consistent across pages | Yes | Same style used across all pages |
| 1. The design is not over emphasised using bold | Yes | Bold is only used for titles and important information only |
| 1. No ads or pop-ups | Yes |  |
| 1. Information is clear and easy to understand | N/A | Only Lorem text is displayed in the prototype |
| 1. URLs are user-friendly | N/A | URLs cannot be edited at the current time |
| 1. HTML Titles are user-friendly | Yes | They are the same as the titles |

**Conclusion:** More features need to be added to finalise the testing e.g. the use of buttons and the code view but everything currently implemented is working or in development. User feedback may dictate a change in design choice or added colours etc. All user feedback and Usability testing should be considered before the final version of the product is released. Current prototype requires a lot of work but includes the basic principles of the website and provides a basis to which the project team can evaluate and work on.

## User Usability Testing

Objectives:

* How long does it take for the user to find the ‘Code View’?
* Can the user successfully edit and then Start the animation?
* On first entering the website, how long does it take for the user to play the video?
* What percentage of users interact with the animation before reading the long text at the bottom of the page?
* General feedback from accessibility and navigability of website.

References:

1. Designing User Experience: A Guide to HCI, UX and Interaction Design – Benyon, David 2019
2. w3schools.com – HTML, CSS and JavaScript concepts and syntax
3. <https://codesandbox.io/s/zen-minsky-bwyr9?from-embed=&file=/index.html> - JavaScript code for the BubbleSort animation used in the project.