A picture containing text

Description automatically generated

Causeway Coast & Glens Interactive Web Mapping (GIS)

A dissertation submitted in partial fulfilment of

The requirement for the degree of

MASTER OF SCIENCE in Software Development

in

The Queen’s University of Belfast

By

Markus Condren

9th September 2022

**Declaration of Academic Integrity**

Before signing the declaration below please check that the submission:

1. Has a full bibliography attached laid out according to the guidelines specified in the Student Project Handbook
2. Contains full acknowledgement of all secondary sources used (paper-based and electronic)
3. Does not exceed the specified page limit
4. Is clearly presented and proof-read
5. Is submitted on, or before, the specified or agreed due date. Late submissions will only be accepted in exceptional circumstances or where a deferment has been granted in advance.
6. Software and files are submitted via Canvas.

**I declare that I have read both the University and the School of Electronics, Electrical Engineering and Computer Science guidelines on plagiarism - http://www.qub.ac.uk/schools/eeecs/Education/StudentStudyInformation/Plagiarism/ - and that the attached submission is my own original work. No part of it has been submitted for any other assignment and I have acknowledged in my notes and bibliography all written and electronic sources used.**

**I am aware of the disciplinary consequences of failing to abide and follow the School and Queen’s University Regulations on Plagiarism.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name: (BLOCK CAPITALS)** |  | | | | |
| **Student Number:** |  | | | | |
|  |  |  | |  | |
| *Student’s signature* |  | | *Date of submission* | | **09/09/22** |

Acknowledgements

Abstract

Introduction

* Come back to once rest is done, will be easy to write by then.

Chapter 1: Understanding the Problem

* MoSCoW analysis of requirements
* Potential languages, frameworks and libraries to be used for the project
* Development strategy – must be clearly explained and justified in terms of the problem and the project

In 2019, the Royal National Lifeboat Institution (RNLI) attended 17,000 incidents, giving aid to nearly 30,000 people. Of these, 200 alone were along the coast of Northern Ireland (CommunityAd, 2020). As a result, following the easing of lockdown restrictions in May 2020, fearing an influx of visitors to the coast, the RNLI and HM Coastguard launched a “beach smart” campaign (NIWorld, 2020) aiming to inform residents and visitors on how to enjoy the beaches and coastline of Northern Ireland, England, Wales and Scotland safely and with consideration of the natural environment. This was done in the hopes of reducing the load on their, already strained, resources. Comprising 12 district councils (Kelly and Whyte, 2019), Causeway Coast & Glens Council (CCG) oversees 11 beaches stretching from Benone to Cushendall (Causeway Coast & Glens Borough Council, 2022). There is limited in-depth information, that is easily accessible in one place online, about the coast in general but the beaches. This makes it difficult, particularly for elderly and people with disabilities, to access to required information to enjoy the coastal features safely.

RNLI crews gave aid to 53,665 people in 2021, up 60% from 33,546 in 2020 (RNLI, 2021). While 35 people required assistance from volunteer lifeboat crews *daily* in 2021, a 52% increase from 23 in 2020 (RNLI, 2021). It must be said that these figures relate to the UK in its entirety, with much of the higher lifeboat activity being seen in the south of England (Fig. 1). Despite this fact, as mentioned previously, there is a not insignificant number of incidents taking place along the coast of Northern Ireland, e.g., nearly 4,000 people saved via water rescue since 2011 (RNLI, 2021). Coupled with the wild and ragged nature of much of the North Coast, which remains largely unwatched by safety crews, it is worth trying to minimise incidents by whichever means.

Map

Description automatically generated

Figure 1 - RNLI Lifeboat Activity Map 2021 (RNLI, 2021)

It is hoped that by making this information easily accessible to more people, there will be fewer incidents requiring the attention of the RNLI or coastguard. This should, in turn, allow for resources to be further invested in improving beach safety for all, improving equipment and training, and potentially free up funds for investment in the personnel that carry out the rescue work. This may be particularly prudent, given that 95% of RNLI staff are volunteers (RNLI, 2021) and 92% of the money raised by the RNLI comes from charitable donations (RNLI, 2021), so there is never a guarantee of further funding.

The beach smart web app aims to provide a single portal for all things CCG. Users will be able to use the app to explore CCG beaches, coastline and towns, including local landmarks, entertainment, eateries and shops.

This, primarily, is aimed at improved beach safety education among locals, with an estimated 144,943 people currently living in the CCG area (NINIS, 2020) as of June 2020. The same technology may then be used for the rest of the United Kingdom, given that the beach smart campaign encompasses all the coastal territory under the remit of the RNLI.

Tourists, however, represent different challenges when it comes to beach safety as they may be unfamiliar with their surroundings, may be inexperienced with beach and sea safety concepts or may simply take more risks than usual due to their being on holiday.

Tourism spending accounts for 5.7% (£131.5 billion) of the entire UK’s GDP, in the same year (World Travel & Tourism Council, 2022). In Northern Ireland alone, 56% of visitor trips are made up of external visitors (DfE, 2020). The COVID-19 pandemic did have a significant impact on tourism levels with **x** amount fewer tourists coming to Northern Ireland due to pandemic and lockdown restrictions ( ). Having said that, pre-pandemic there were 5.3m tourists visiting Northern Ireland, this figure (<https://www.tourismni.com/industry-insights/tourism-performance-statistics/>). The COVID-19 pandemic saw significantly fewer tourists visit. Pre-pandemic tourism contributed £1bn to the NI economy, of which 70% was from external visitors (DfE, 2020). Therefore, it is important for the beach smart app to be rich in functionality that applies to tourists as well as locals, to take advantage of recovering tourism numbers. This could have a significant impact on the economic prosperity of the area in the coming years and help improve education around beach safety for all.

**ADD DEVELOPMENT METHODOLOGY**

Following the analysis of the problem at hand and having decided on the development methodology to be used, requirement prioritisation is the process of looking at the proposed features a client or customer wants to include in their product and deciding on how best to prioritise them during the development cycle. This is done to optimise the allocation of time and resources to the project so that a high-quality product may be brought to market in the quickest time possible, with as little waste as possible. Requirement analysis is oft considered one of the more important aspects of the development lifecycle (Ngo-The and Ruhe, 2005), with poor analysis having the potential to de-rail projects at any given time.

There are a few well used methods of requirement prioritisation widely used in industry. For the project, three were compared, Ranking, Kano Analysis and MoSCoW prioritisation.

Ranking involves creating a big list of potential requirements and ranking them from most to least important. Although simple in theory, ranking requirements in this way may become unmanageable as requirements are added and the project increases in complexity. It has been suggested to reserve this for very small projects as to process many requirements in one large list becomes inefficient once they grow beyond 7 (Miller, 1994). It was decided that Ranking would be inappropriate as the project would likely have more requirements and it did not feel appropriate when following an Agile methodology as requirements are listed by preference, but no indication is given as to the amount of preference between requirements; they are all weighted equally (Hatton, 2008).

The Kano Model was developed by Professor Noriaki Kano in 1984 (Kano, 1984; Wikipedia, 2022). Kano analysis looks at requirements for a project in terms of their effect on customer satisfaction. This is done by assigning each product requirement an attribute of one of the following. Threshold: the most basic requirements expected of the product, Performance: requirements that may not be wholly necessary but would improve the customer satisfaction of the product and Excitement: these are requirements which the client may not have even thought of but ones which would add a high level of satisfaction to the final product (MindTools, 2016). Requirement analysis using Kano analysis can then be ratified through standardised questionnaires, given to the development team, client group or prospective users, in order to statistically measure opinion on the matter (Kano, 2022). The fact that the Kano Model uses real customer data to analyse requirements is intriguing, this would offer a wide range of opinions from real-world users of various backgrounds and persuasions. In theory, this would lead to a more robust, customer-centric development cycle with more highly tuned requirements. In the case of this project, it may have been somewhat excessive. Unfortunately, the proposed contact at the council is off on long-term leave, this lead to Dr David Cutting acting as the client (who really was excellent in doing so) and meant there would be no access to the prospective users in the numbers required for accurate and valid conclusions to be drawn from any questionnaire-based survey – a minimum of 100 participants has been suggested as this number (Graglia, 2022; Bullen, 2022). Were this project to be carried out on an industrial scale, this may well have been the most appropriate form of requirement prioritisation due to the user-focussed nature of the project.

Finally, MoSCoW prioritisation. Developed in 1994 by Dai Clegg (Agile Business, 2022), the acronym MoSCoW denotes the four categories in which requirements can be placed, these describe the hierarchical importance of each requirement in terms of *must* have, *should* have, *could* have and *won’t* have (yet), with the letter ‘o’ added in-between to aid pronunciation. Must have requirements are non-negotiable, they must be included for there even to be a minimum viable product. Should have requirements are features that would make for a nice addition to the project if they can be developed and integrated but won’t cause it to fail if not. Could have requirements are ones which would add to the product, although less than should haves, but are also not essential to the completion of the project. Won’t have requirements will not be included, this must be caveated with the fact that they are not bad or inconsequential requirements, they are simply ones which will not be included within the given timeframe of the project. Though these may be revisited as the project in maintained and further developed upon.

**WHY DID I PICK MOSCOW OVER THE OTHER TWO?**

|  |  |
| --- | --- |
|  | Requirement |
| Must have | * Accessibility features for users with keyboard-only or visual/auditory impairments – to include light/dark themes and support for text-to-speech services. * An interactive map with pins displaying the beaches that the council oversees. * Users must be able to explore the map, to include zooming and panning. * A link for each beach to the relevant CCG website page. * Environment, historical and practical (accessibility, closure times etc.) information for a beach that is selected. To include beach status such as flag and water category. * Users must be able to easily contact the council for any extra information or to report anything. |
| Should have | * Local landmarks, businesses and sports facilities, highlighted with icons/pins – to include opening hours, contact details and links to external websites, where applicable. * A map filter allowing users to decide what type of amenity or attraction type they wish to browse. * Live weather and tidal data allowing users to safely explore the area. * Bus and train routes around the area with timetables (better UX) or links to external Translink website. * Should have external links to other websites for information on the area, beaches, lifeguards etc., this should open externally so as not to confuse any less technically minded users. * Points of interest stored in a database and accessed via URL requests. * There should be the ability for an admin to access all points of interest as a list and be able to add, update or delete points of interest on the map. |
| Could have | * Users could pick a list of beaches, landmarks etc., they wish to visit and give their available time, start and end point. An algorithm would be used to calculate their most efficient route. This is perhaps a more tourist-oriented requirement but could be of great use to locals too. * Users could be able to create an account – allowing them to add map items to a watchlist, get notifications about events and save favourite routes – could sign up with email or Google account. * Users could report any rubbish, damaged property or anti-social behaviour to the council, perhaps uploading a photo from the app. This would be reported along with time, date and geolocation data, allowing the council to take quick and efficient action. * Users could report sightings of wildlife e.g., birds, whales, dolphins or seals, which occasionally show up on the North Coast. These could be pinned to a location to let others know where they may catch a glimpse. This could be used also to highlight areas where people have, for example, been stung by jellyfish. This could allow others to bathe and enjoy the beach safely and with greater peace of mind. * An API, likely built using PHP or NodeJS, allowing other developers to make use of the aggregated data used in the app. * Social media aggregators which display tweets or Facebook posts using pre-determined hashtags or the CCG social feeds, as per “plan your trip” page on the CCG website (RNLI, 2022). |
| Won’t have yet | * Integration with Fitbit, Strava or similar, to save routes or walks and share progress with friends and family. * Social media integration allowing posts to be made direct from the web app. * Photos uploaded by viewers aggregated together and shown in the information section of each pin location as a slideshow with username and date below. * Augmented reality functionality, allowing users to preview the area and decide as to where they wish to visit. |

Once requirements have been decided upon and prioritised, the technology best suited to the project must be chosen.

**Technology considerations**

The brief for this project was to create a web-based, interactive mapping application. Thus, technology was considered for front and back-end as well as the database element.

*Front-End*

Given the web-based nature of the project, the main structure of the website was developed in HyperText Markup language (HTML) 5, with styling coming from Cascading Style Sheets (CSS).

When considering a general plan for CSS, global variables to be used where possible for colours, fonts and any other constant values that are to be reused throughout the codebase. Where possible, height, width, margins, etc., to be controlled vw and vh, keeping them relative to the width and height of the given viewport. This will be important for maintaining responsiveness across the entire array of modern device screen dimensions. Likewise, font sizing to be controlled using relative units – rem or em for the same reason. The decision remained whether to use plain CSS or a library/framework. There are advantages and disadvantages to both; plain CSS allows for ultimate customisation and control over every aspect of the app’s CSS and can avoid the added project bulk, given that some external frameworks and libraries often are massive files (Agrawal, 2021), which on a project of this scale may or may not affect performance. On the other hand, writing all the app’s CSS from scratch would take a huge effort, with the added potential for bugs. The advantages of using a framework tend to revolve around the ease with which one can implement advanced features e.g., navbars, dropdown menus or nested elements which maintain their properties throughout (something which can be tricky when building CSS from scratch). Frameworks also allow for cross-browser support of the project’s features and should minimise the occurrence of bugs as new versions of browsers are developed, and older versions defunct (Bose, 2022). The main advantage of using a framework in the case of this project is that they allow for the development of a modern, stylish user interface (U.I) that requires minimal external styling, with high-quality responsiveness across devices, with minimal effort and, quite often, well-written documentation (Coyier, 2008).

When considering the use of a CSS framework, there are many popular ones in the market. There were three considered for this project due to having previous experience working with each – Bulma, Tailwind CSS and Bootstrap.

Bulma is an open-source, class-based, mobile-first CSS framework (Bulma, 2022). Built on flexbox, it is responsive across various devices. It comprises CSS classes which are easy to implement in a project, including many components such as forms, navbars, menus etc., which are simple and sleek in appearance. Bulma also includes many optional Sass (an extension language of CSS) files which can be imported for additional functionality (Juviler, 2022).

Tailwind CSS is a lightweight, fast framework, described as ‘utility-first’. Unlike Bulma and Bootstrap, Tailwind is not a U.I. kit but rather than have a default U.I. theme, tailwind uses ‘widgets’ and looks at the HTML and JavaScript code one has written to generate a static CSS file using the relevant class names (TailwindCSS, 2022). This in theory allows the quick creation of website U.I.s, with one commentator describing it as “a cool way to write inline styling” (Ekwuno, 2021). Tailwind has fast loading times but is not as robust as the likes of Bootstrap when it comes to deploying U.I. components which can be utilised site-wide (Ozanich, 2022).

Bootstrap is a mobile-first build which scales well on desktop. As is the case with Bulma, Bootstrap allows for a lightweight, responsive design with components that are easily customisable and offer high levels of functionality. Bootstrap also allows access to open-source icons for customising the look of the site. As these come with Bootstrap, they are free to use without additional accreditation, reducing the likelihood of legal complications arising from the finished product. Although external icons may be used, with accreditation. It is robust in its use of HTML, CSS and JavaScript component templates (Ozanich, 2022), and has a good track record in industry being, according to W3Techs, used by over 25% of websites (W3Techs, 2022).

As this project is primarily designed to be used as a web-app on a mobile phone, it was important to use a framework that is designed ‘mobile-first’, with high-quality and reusable U.I. components with functionality that reflects this. Tailwind CSS has many advantages, not least with its smaller loading times. It was decided though, that a U.I. kit framework such as Bulma or Bootstrap would be more appropriate. Bulma or Bootstrap would be perfectly adequate for this project but ultimately it was decided that Bootstrap would be used, primarily out of personal preference, having worked with both throughout the year.

Having decided on a CSS framework, the next technology to consider was JavaScript. JavaScript is a client-side scripting programming language - which can also be used in server-side applications, in non-browser environments e.g., Node.js - currently used in 98% of websites (W3Techs, 2022). It is a lightweight language which supports imperative, declarative and object-oriented programming (MDNWebDocs, 2022) and is compiled at run-time as opposed to pre-execution (Aycock, 2003). JavaScript is responsible for all the proposed interactive elements of the project and is the go-to in any modern, web-based application (Duraj, 2019). The question remains as to which ‘flavour’ of JavaScript would be most appropriate.

Vanilla JS

Typescript

JS front-end frameworks

React

* React or React Native for Front End UI
  + React is a Java Script library rather than framework, allows for the creation of reusable components for the app front end.
  + High speed and good performance due to the virtual DOM (<https://massivepixel.io/blog/react-advantages-disadvantages/>).
  + Allows for quick iterations of product and should make responding to client requests for edits more straight forward.
  + Search Engine Optimisation friendly (<https://www.javatpoint.com/pros-and-cons-of-react>), comes in handy for making people aware the product exists.
  + Updates are carried out regularly, keeping things up to date with modern standards.

Angular

Mapping APIs - a couple of interactive mapping APIs have been considered, with the two main options being Google Maps and Open Street Maps

* Google Maps
  + Widely adopted by all browsers, small learning curve for most users.
  + Highly customisable with good map controls.
  + Expensive to use at larger scale - $300 free trial mode, then $200 monthly credit for Google maps platform - <https://developers.google.com/maps/billing-and-pricing/billing#monthly-credit>.
* Open Street Maps
  + Open-source, free to use mapping system. Community driven and maintained which gives the option of maintaining accuracy for more obscure localities. This gives the developer more control as the map can be edited to remove any inaccuracies.
  + Good API documentation available.

Mapping frameworks

* Leaflet JS to be used for map integration.
  + Open-source framework that is designed to be mobile-friendly. As this project will be developed “mobile-first” this seems to be appropriate.
  + Works seamlessly with Open Street Maps, with an easy-to-follow guide for implementation.
  + Core features are designed to be extremely lightweight but there are several plug-ins available to add extra functionality.

*Back-End*

* Server-side
  + Node JS
    - Runtime environment for Java Script, allowing it to be used on the server-side.
    - Developing full stack in Java Script gives certain benefits such as high speed, good performance and good efficiency (<https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-node-js-web-app-development/>).
    - Can use Express JS framework alongside, although many other options available too.

*Database*

* + MySQL
    - Widely used legacy system with syntax I am somewhat familiar with already.
    - Secure Socket Layer makes transmitted data well protected.
  + Mongo DB
    - No SQL database – that is rather than a relational data system as with MySQL, Mongo DB is an object-based system that employs the use of JSON objects.
    - More flexible when it comes to searching for data and building a dataset.
  + Both are support Java Script server-side technology. MySQL seems more appropriate for high traffic sites that require high levels of security. Mongo DB is said to be more applicable for sites with an analytical focus.
  + As for which is better, MongoDB being part of the MERN stack containing React may be more appropriate, despite the steeper learning curve having never used it before.

Chapter 2: User Interface Design

It was important to consider the user-interface design in terms of user experience, design principles and accessibility.

The font family: Arial, Helvetica, sans-serif was gathered from CCG website using Font Finder chrome extension (Fig. **X**), a simple, clean and crisp font-family that is easily recognisable, which scales well as device size changes.

Graphical user interface, application

Description automatically generated with medium confidence

Figure **X** - Font family gathered from CCG website using Font Finder (https://chrome.google.com/webstore/detail/font-finder/bhiichidigehdgphoambhjbekalahgha?hl=en)

* + Logo – there doesn’t seem to be a beach smart logo as such. As this is a joint venture between CCG and the RNLI, these could be used. If allowed, having the opportunity to design a logo would be an exciting prospect.
  + At this point there are no definitive requirements for design aspects so initial thoughts are with the current colour palette for CCG site in mind. Personally, I think the CCG colour scheme isn’t particularly modern but will be used if the client wishes to keep consistency across products. If given some freedom in this decision a different colour palette will be developed.
    - Site palette - Jet: #333333ff; English-violet: #503250ff; English-violet-2: #6a4c6dff; Quinacridone-magenta: #85425cff; White: #ffffffff; Dark-cornflower-blue: #163982ff; Blue-sapphire: #195d70ff
    - Extra colours which are complementary – palette created using <https://coolors.co>. Inclusion of blue colours will bring some contrast to the current, darker, CCG colour scheme. They also lend themselves
      * + Opal: #9bc4bcff; Aero-blue: #d3ffe9ff; Middle-blue: #8ddbe0ff

Chart, treemap chart

Description automatically generatedChart, treemap chart

Description automatically generated

Chapter 3: Architecture design and algorithm explanation

* High level overview of architecture to allow someone who is going to be maintaining and/or adapting the project.
  + App
    - Single page/multipage
  + API
    - Design ideas to allow for easy integration with the project.

Chapter 4: Experimentation (if relevant)

* Could I do the statistical analysis quickly of a survey for things that people look out for in a mapping application?
* Not much else experimentation is relevant?

Chapter 5: Testing

Chapter 6: Evaluation and Conclusion

It seems most appropriate to begin the evaluation of this project by looking at how well it has fulfilled the initial requirements at this stage in development.

Bibliography

Causewaycoastandglens.gov.uk. 2022. *Beaches - Causeway Coast & Glens Borough Council*. [online] Available at: <https://www.causewaycoastandglens.gov.uk/see-do/beaches> [Accessed 30 August 2022].

CommunityAd, 2020. *RNLI and HM Coastguard launch beach safety campaign*. [online] CommunityAd.co.uk. Available at: <https://www.communityad.co.uk/rnli-hm-coastguard-launch-beach-safety-campaign-urging-parents-protect-families-save-lives-coast-summer/> [Accessed 09 August 2022].

DfE, 2020. *Tourism | Department for the Economy*. [online] Economy. Available at: <https://www.economy-ni.gov.uk/topics/tourism> [Accessed 20 August 2022].

Kelly, C. and Whyte, N., 2019. *Causeway Coast and Glens District Council*. [online] Ark.ac.uk. Available at: <https://www.ark.ac.uk/elections/nlgccg.htm> [Accessed 9 August 2022].

NINIS, 2020. *NINIS: Northern Ireland Neighbourhood Information Service*. [online] Ninis2.nisra.gov.uk. Available at: <https://www.ninis2.nisra.gov.uk/public/AreaProfileReportViewer.aspx?FromAPAddressMulipleRecords=Causeway%20Coast%20And%20Glens@@Causeway%20Coast%20And%20Glens@22?#:~:text=The%20estimated%20population%20of%20Causeway,73%2C104%20(50.4%25)%20were%20female.> [Accessed 30 August 2022].

NIWorld, 2020. *New RNLI and HM Coastguard Campaign*. [online] northernirelandworld.com. Available at: <https://www.northernirelandworld.com/news/new-rnli-and-hm-coastguard-campaign-2860751)> [Accessed 6 August 2022].

RNLI, 2021. *RNLI lifeguards in Northern Ireland rescue 3,701 people in a decade of patrols*. [online] rnli.org. Available at: <https://rnli.org/news-and-media/2021/june/23/rnli-lifeguards-in-northern-ireland-rescue-3701-people-in-a-decade-of-patrols> [Accessed 18 August 2022].

RNLI, 2021. *RNLI: Annual Report and Accounts 2021*. [ebook] Royal National Lifeboat Institution (RNLI), pp.5,6,9. Available at: <https://rnli.org/about-us/how-the-rnli-is-run/annual-report-and-accounts> [Accessed 22 August 2022].

RNLI, 2021. *Running Costs*. [online] rnli.org. Available at: <https://rnli.org/about-us/how-the-rnli-is-run/running-costs#:~:text=The%20RNLI%20is%20funded%20primarily,RNLI%20receives%20no%20government%20funding.> [Accessed 20 August 2022].

VisitBritain, 2014. *Britain's visitor economy facts*. [online] VisitBritain.org. Available at: <https://www.visitbritain.org/visitor-economy-facts> [Accessed 18 August 2022].

World Travel & Tourism Council, 2022. *2022 Annual Research: Key Highlights*. [ebook] World Travel & Tourism Council. Available at: <https://www.statista.com/statistics/598093/travel-and-tourism-gdp-total-contribution-united-kingdom-uk/> [Accessed 20 August 2022].

Hatton, S., 2008. *Choosing the "Right" Prioritisation Method*. [ebook] Crawley, Western Australia: School of Computer Science and Software Engineering, The University of Western Australia. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4483241> [Accessed 20 August 2022].

Ngo-The, A., Ruhe, G. 2005. Decision Support in Requirements Engineering. In: Aurum, A., Wohlin, C. (eds) Engineering and Managing Software Requirements. Springer, Berlin, Heidelberg. Available at: < https://link.springer.com/chapter/10.1007/3-540-28244-0\_12> [Accessed 20 August 2022].

Miller, G., 1994. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 101(2), pp.343-352.

Wikipedia, 2022. *Kano model - Wikipedia*. [online] En.wikipedia.org. Available at: <https://en.wikipedia.org/wiki/Kano\_model#:~:text=The%20Kano%20model%20is%20a,customer%20preferences%20into%20five%20categories.> [Accessed 18 August 2022].

Noriaki Kano, Nobuhiko Seraku, Fumio Takahashi, and Shin-ichi TSUJI 1984. 'Attractive Quality and Must-Be Quality,' *Journal of the Japanese Society for Quality Control*, Vol. 14, No. 2, pp. 147-156.

MindTools, 2016. *Kano Model Analysis: Delivering Products That Will Delight*. [online] mindtools.com. Available at: <https://www.mindtools.com/pages/article/newCT\_97.htm#:~:text=What%20Is%20Kano%20Model%20Analysis,it's%20also%20about%20customers'%20emotions.> [Accessed 19 August 2022].

Kano, 2022. *Kano+ | Kano studies online*. [online] Kano.plus. Available at: <https://www.kano.plus/#kano> [Accessed 31 August 2022].

Bullen, P., 2022. *How to choose a sample size (for the statistically challenged) - tools4dev*. [online] tools4dev. Available at: <https://tools4dev.org/resources/how-to-choose-a-sample-size/#:~:text=The%20minimum%20sample%20size%20is,to%20survey%20all%20of%20them.> [Accessed 20 August 2022].

Graglia, D., 2022. *How many survey responses do I need to be statistically valid? | SurveyMonkey*. [online] SurveyMonkey. Available at: <https://www.surveymonkey.co.uk/curiosity/how-many-people-do-i-need-to-take-my-survey/> [Accessed 20 August 2022].

Agile Business, 2022. *Chapter 10 - MOSCOW Prioritisation*. [online] agilebusiness.org. Available at: <https://www.agilebusiness.org/page/ProjectFramework\_10\_MoSCoWPrioritisation> [Accessed 21 August 2022].

Agrawal, S., 2021. *The World Of CSS. Should you write your own CSS or use any Library?* [online] DEV Community 👩‍💻👨‍💻. Available at: <https://dev.to/theshubhagrwl/the-world-of-css-should-you-write-your-own-css-or-use-any-library-4i57> [Accessed 22 August 2022].

Bose, S., 2022. *Top 5 CSS Frameworks for Developers and Designers | BrowserStack*. [online] BrowserStack. Available at: <https://www.browserstack.com/guide/top-css-frameworks> [Accessed 22 August 2022].

Coyier, C., 2022. *What Are the Benefits of Using a CSS Framework? | CSS-Tricks*. [online] CSS-Tricks. Available at: <https://css-tricks.com/what-are-the-benefits-of-using-a-css-framework/> [Accessed 22 August 2022].

Bulma, 2022. *Overview*. [online] Bulma.io. Available at: <https://bulma.io/documentation/overview/> [Accessed 21 August 2022].

Juviler, J., 2022. *The Bulma CSS Framework: What It Is and How To Get Started*. [online] Blog.hubspot.com. Available at: <https://blog.hubspot.com/website/bulma-css> [Accessed 23 August 2022].

Ozanich, A., 2022. *TailWind CSS vs Bootstrap: What is The Difference & Which One is Best?*. [online] Blog.hubspot.com. Available at: <https://blog.hubspot.com/website/tailwind-css-vs-bootstrap> [Accessed 21 August 2022].

W3techs, 2022. *Usage Statistics and Market Share of Bootstrap for Websites, September 2022*. [online] W3techs.com. Available at: <https://w3techs.com/technologies/details/js-bootstrap> [Accessed 1 September 2022].

W3Techs, 2022. *Usage Statistics of JavaScript as Client-side Programming Language on Websites, September 2022*. [online] W3techs.com. Available at: <https://w3techs.com/technologies/details/cp-javascript/> [Accessed 1 September 2022].

Aycock, J., 2003. A brief history of just-in-time. *ACM Computing Surveys*, 35(2).

MDNWebDocs, 2022. *JavaScript | MDN*. [online] Developer.mozilla.org. Available at: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> [Accessed 1 September 2022].

Duraj, M., 2019. *How JavaScript Implementation Adds Dynamic Interactivity*. [online] pluralsight.com. Available at: <https://www.pluralsight.com/guides/how-javascript-implementation-adds-dynamic-interactivity> [Accessed 1 September 2022].

Appendices