WEB701 Web Design

Max Adams

NMIT Student

Helping Families Nelson

Design, Evaluation and Development

# Executive Summary:

Contents

[Executive Summary: 1](#_Toc73446468)

[Part One - Design 3](#_Toc73446469)

[Introduction 3](#_Toc73446470)

[Designing the website 3](#_Toc73446471)

[Website Brief 3](#_Toc73446472)

[What Are the Website Requirements? (Using Information Architecture Analysis) 4](#_Toc73446473)

[User Requirements 4](#_Toc73446474)

[Site Goals 4](#_Toc73446475)

[Define the User Experience 5](#_Toc73446476)

[Site Content 13](#_Toc73446477)

[Site Structure 16](#_Toc73446478)

[USER STORIES – DESIGNING TO BUILD 29](#_Toc73446479)

[CRUD Analysis 29](#_Toc73446480)

[Part Two – Evaluate 30](#_Toc73446481)

[The Purpose of Web Frameworks 30](#_Toc73446482)

[Website Requirements: 31](#_Toc73446483)

[1. Charity members and beneficiaries can register, log in and administer their own accounts. 31](#_Toc73446484)

[2. Register Product & Acquire Tokens 35](#_Toc73446485)

[3. Interactive Elements: 37](#_Toc73446486)

[4. Token Interface 39](#_Toc73446487)

[5. Store and Retrieve data from a server-side database 41](#_Toc73446488)

[Recommendation of Framework (Conclusion): 41](#_Toc73446489)

[Angular 41](#_Toc73446490)

[Vue 41](#_Toc73446491)

[Decision 42](#_Toc73446492)

[References 43](#_Toc73446493)

# Part One - Design

# Introduction

# Designing the website

## Website Brief

The Helping Families Nelson Charity is a “clothes distribution style” charity in the Nelson region who offer their clothing distribution service to the local community.

We need a website for the Nelson region for members of the charity to provide clothes or wearable apparel.

The purpose of the new website is to help with the distribution of the service or product by members of our charity to people in need. The website lets members register and describe the service or products they are providing, the number and frequency of the offerings.

The website lets members of the community (beneficiaries) access the service, by providing tokens that can be “spent” in the service. To achieve our goal, we would like to have an online token function on our website that let a beneficiary access a limited number of tokens, for example the system needs a mechanism that verifies that a member of the community is accessing the service and that the token belongs to a particular community member. The website needs some homepage text about general information of our products and services such as: types, characteristics, quality factor, usage and benefit.

We require the following features on the website:

1. Charity members and beneficiaries can register, log in and administer their own accounts.

2. Members use the website to register their products and services, and beneficiaries use the system to acquire tokens.

3. Interactive element(s) that engages the website user.

4. The system provides an interface that the members can use to accept a token in a transaction.

## What Are the Website Requirements? (Using Information Architecture Analysis)

### User Requirements

#### General Functionality

We propose that the website will need to feel very friendly and personal to draw in potential donators to your cause. There will be images of people smiling as they receive an item that has been donated, and images of people that need to be supported as it will help users empathize with your cause. There will be bright, warm colours and personal quotes from people in your organization to add that personal touch.

#### Specific Functionality

There could be specifically pages dedicated to:

* The cause of the charity and people that are currently being helped i.e., the local people groups that are in need of or have received donations.
* An informative page of the type of products that you are requesting.
* A page dedicated to the staff involved with a quick bio including the reason they are participating.
* A login/registration page.
* A page where a user can upload their items to donate.
* An account page where they can view their donations.
* A thankyou page after donating.

### Site Goals

#### Mission or purpose of the website

The goal of the website to provide an online donation interface for the charity Helping Families Nelson. It will also promote the charity and make it easier for you to share your mission and get people involved that could otherwise be unaware of your charity. It will also highlight the need for peoples help on a more global scale and be used for promoting the good cause of donating and distributing clothes to those in need.

#### Short Term Goals

The short-term goal for the website it to create a space that allows users to upload their items to donate including pictures and to promote the cause of the charity. It will need a login functionality and the ability to upload data to a server where you are able to view the items uploaded and approve/disprove them before they are donated. Each item will be given a category so that when they drop them off, they will know which box they are to place each donated item in to save you sorting it.

#### Long Term Goals

The long-term goals for the website is to promote the need for people that have ample or old clothes to give to those that are struggling. Through this website and your charity we hope to help the locals of Richmond realize that they are able people close to them that they are able to see being affected. It will also make people aware that they don’t need to donate money to places overseas where it is difficult to see the benefit (though that is still a good thing!) but whether they are able or unable to do so, it is much easier to donate clothes that are still in good nick, but aren’t used as often as they used to be.

#### Intended Audiences

There will be many types of people that will want to visit your site. Listed below are the kinds of people that we believe will do so:

* General public
* Donators
* Similar organizations/charities
* Organizations that help the poor
* The media/press
* Local/Distant churches

#### Why Will People Visit the Website?

We believe that the cause of your mission is incentive enough for people wanting to visit your website. It is exciting, uplifting and helping many in need. There are also ways to promote the website to the immediate public:

* Adding the website URL to all advertisements that you are currently using. Church flyers, posters etc.
* SEO – Google Search Optimization
  + Implementing this means that your website will be promoted higher when people type certain key words in Google.
* Using local newspapers that will likely be more than willing to do an article on your charity.
* Approaching other churches and town halls/council buildings that may offer to promote the charity for free.

### Define the User Experience

#### 

#### The Audience

Listed below are the people that will be likely to visit the website and the reasons for each group.

##### General Public

The general public will be ordinary members of the community that are interested in getting rid of their clothes or helping charities.

##### Donators

People looking for ways in which they can help in their community.

##### Similar Organizations/Charities

Such as the Salvation Army looking for inspiration to further their business or looking for a partnership.

##### Organizations that help those in need

They may be looking for a partnership with your business.

##### The Media/Press

Looking for an uplifting article about a local charity/org.

##### Local/Distant Churches

They may look to help your charity or wish to start their own in their own area.S

#### Personas

Joshua is a middle-aged male living in Richmond, NZ that has gained a substantial amount of weight due to some new medication and does not foresee himself reverting to his original, skinnier self. Joshua has a lot of clothes that are nice, but unfortunately no longer fit him. He is looking for a good alternative to dumping his clothes.

Caitlin has lost her job in the tourism industry due to Covid-19 and no longer has an income. She is only 2 years out of finishing her course at NMIT and has a large student loan that she cannot pay off and can barely afford rent and food. She still has a computer from her course and is looking for ways to get free or cheap clothes in her area. As she is struggling to find a job and doesn’t want to wear dirty or extremely cheap nasty clothes, she is stoked to find out that the Helping Families Nelson charity exists and decides to pay them a visit to see if they will be able to offer her anything.

Open Home Foundation is an organization that looks after troubled young people. They find foster parents and through Oranga Tamariki, fund these families so that they can look after these young people. Part of this funding includes a clothing allowance which is very limited. They are looking for a place that distributes clothes locally to people that are in need and wish to make a partnership with them so that they can acquire clothes without needing to pay as much so that the money saved on clothing, can be put towards housing more kids and teens in need.

#### Scenarios

Joshua, in his search for getting rid of his clothes, has stumbled across your website! Joshua reads the charities goal and achievements on the main page and sees that it is a local charity that is making a real impact in the community. He realises that he can donate his clothes to help those in his immediate area and becomes quite excited. Joshua immediately creates an account by pressing register in the top right of the navbar and entering his details. He then navigates to the “Upload Item” page also found in the navbar and uploads the photos of the items that he feels will benefit the cause the best along with a description. He enters the category he feels is the most appropriate and feels a lot better about himself as he awaits the response. Once your charity has processed his donation requests, he is rewarded with a token for each item that he can then redeems on the poll that suggests the next causes that your charity will try to help with.

Caitlin, after discovering that the website and charity exists, begins to look to see if she is eligible to receive some of the clothes that have been donated. She begins by looking at the home page but cannot find exactly who is eligible to receive the donated items. She then navigates to the about page where she discovers that anyone that is struggling financially and in the region is a viable recipient for their clothes. She then sees the location of the store further down on the page and decides to pay them a visit as she is not far away and wants to talk to some of the people in person. She then checks the staff members profiles that are also found on the about page to see the kind of people that would volunteer for such a cause.

Karen from the Open Home Foundation is looking for places to find clothes for the number of young people that are currently under their care. She discovers your charity after asking one of her church friends and is shown that week’s newsletter that has a brief overview of the charity as well as the websites URL. She checks the website and reads through all the mission statement, goals, success stories and staff bios found between the about us page and the home page and is thoroughly impressed. She finds your charities phone number in the footer of the page along with your email address and decides to give you a call to see whether a partnership between the two organizations can be formed.

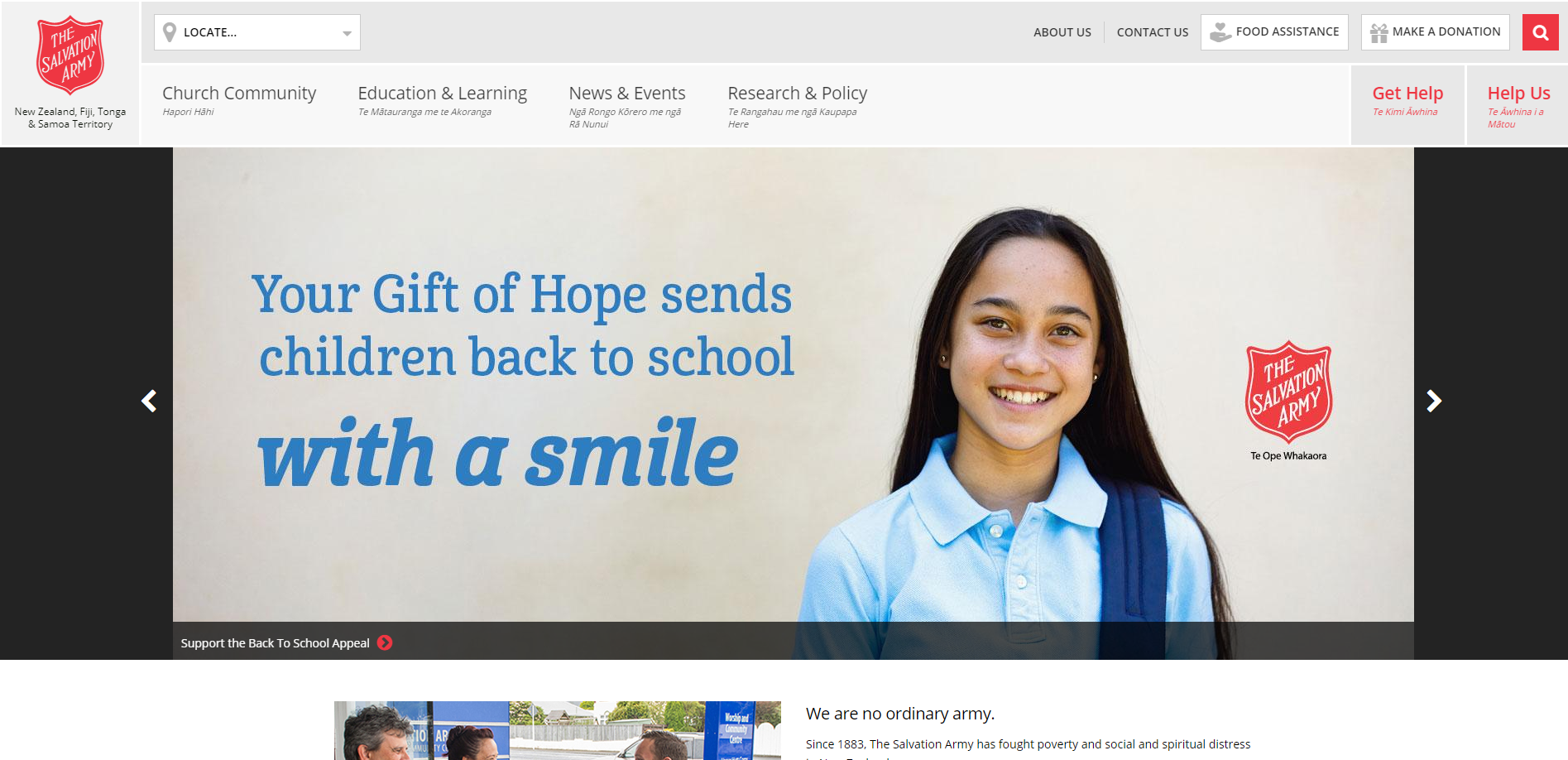
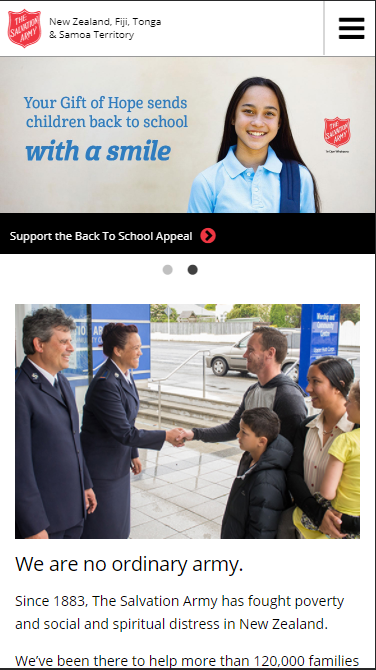
#### Competitive Analysis

Here is a list of websites that are being used for promoting similar charities around NZ:

* <https://www.salvationarmy.org.nz/>
* <https://www.redcross.org.nz/shop-with-us/red-cross-shops/how-donate-our-shops/>
* <https://www.theclothingproject.co.nz/>
* <https://thetidylady.co.nz/>

##### Salvation Army - <https://www.salvationarmy.org.nz/>

###### Homepage:



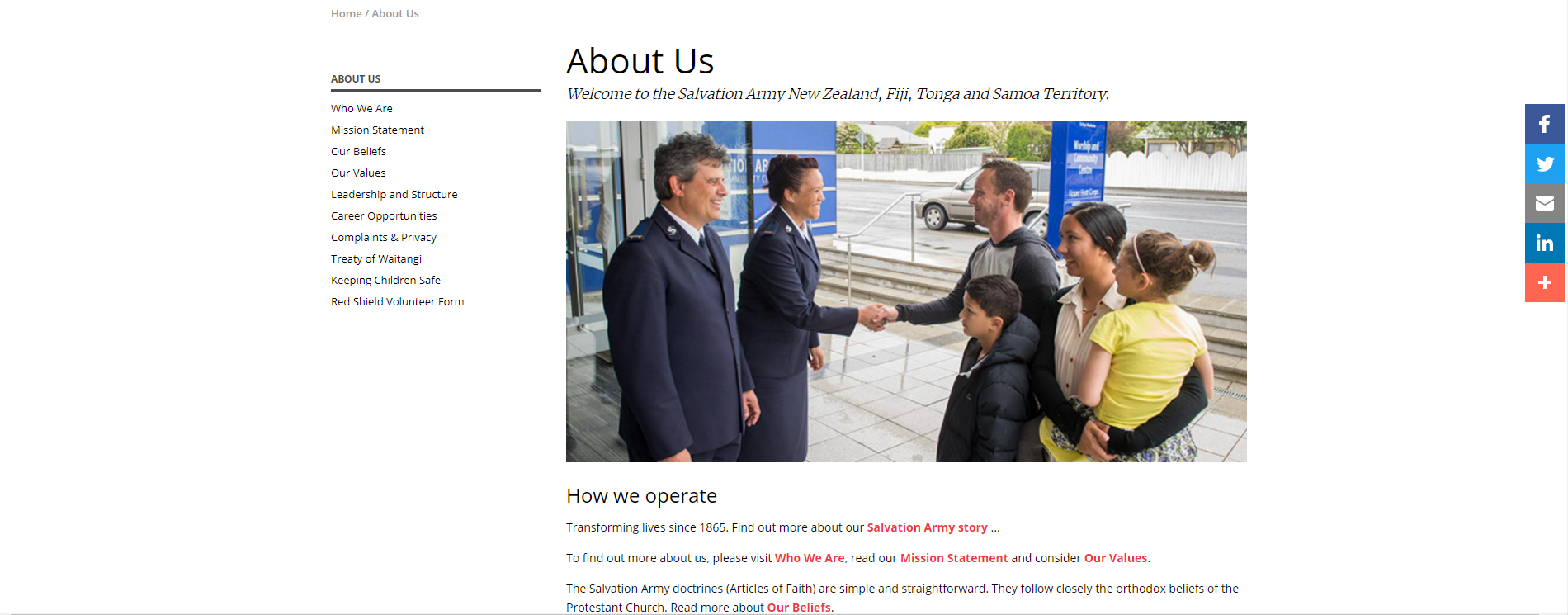
Pros:

1. I like the carousel at the top of the page that shows a happy girl and the caption “Your Gift of Hope sends children back to school with a smile”. It promotes the idea that donating to the charity is benefitting children’s education and their mental state. This is a good message to get across as soon as you view the site.
2. They have a “MAKE A DONATION” button clearly visible at the top of the page. This button is part of the static navbar and is always present on the site reminding people to donate to support the cause.
3. The blurb (just at the bottom of the screenshot) talks about the success of the salvation army as well as their mission statement and history. It is succinct and meaningful.
4. Their use of Te Reo Maori is good as it shows their support for the culture.
5. The site resizes well to fit different aspect ratios of phones and the navbar condenses to a hamburger menu.

Cons:

1. The navbar seems a little bit cluttered in certain areas despite there being lots of empty space in others. This could be easily fixed.
2. The overall site structure seems a little dated. The carousel and map being the full width with content above and below being restricted makes it seem less structured to me.

###### About Us



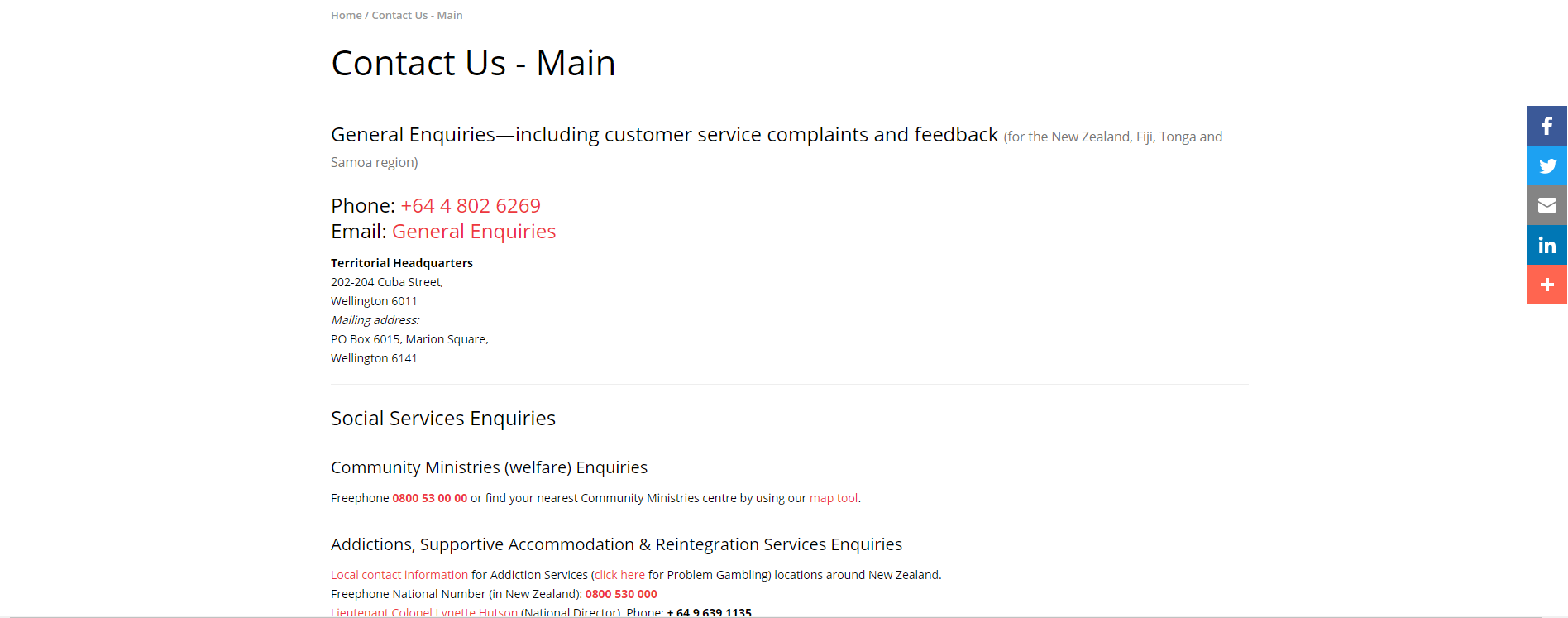
Pros:

1. As there is a lot of information to present in this page, I like their decision to only show one thing at a time with clear links on the left side of the page to navigate each section. It makes this page much more user friendly than it otherwise could have been.
2. The social media icon links on the right are clearly labelled and do not interfere with any of the other content as the rest of the content is all centred.

Cons:

1. There is not a lot interesting about this page at all other than the content. Use of more colour and images could have gone a long way as it feels quite boring.

###### Contact Us



Pros:

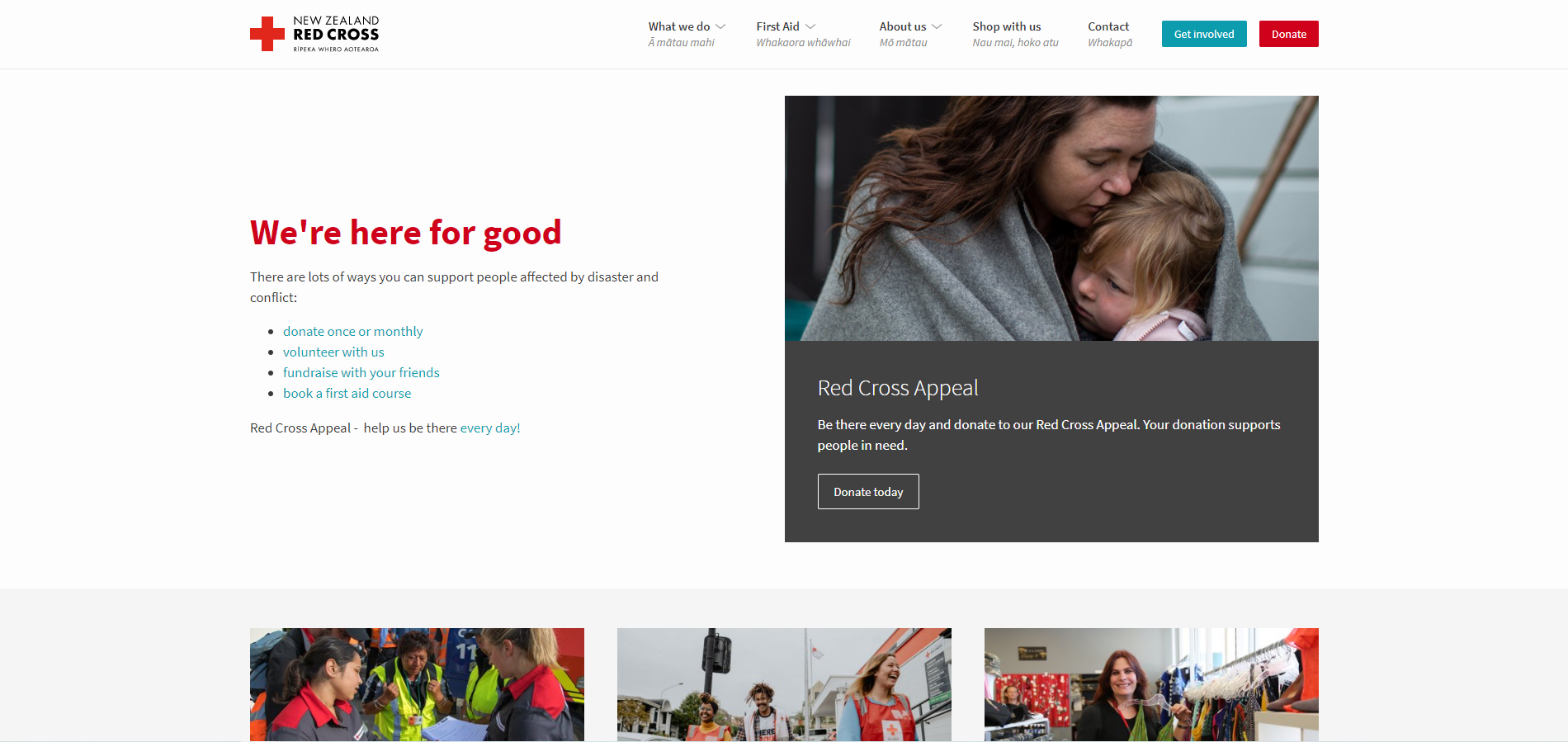
1. The red text is nice as it identifies links in an unobtrusive way.

Cons:

1. There is too much block text and not enough to break it up.

##### Red Cross - <https://www.redcross.org.nz/>

###### Homepage



Graphical user interface

Description automatically generatedPros:

1. Nice clean colour scheme that matches the brand and logo.
2. Modern approach to containers. Looks good.
3. Use of Te Reo in main areas such as navbar.
4. Very resizable/mobile friendly 🡪

###### Donate Container

Graphical user interface

Description automatically generated

Pros:

1. Big and bold, very difficult to miss.
2. White font to stand out.
3. Call to donate.
4. Offers a one off so people don’t feel trapped into paying more than they want.

##### Graphical user interface, website Description automatically generatedThe Clothing Project - <https://www.theclothingproject.co.nz/>

###### Homepage

A group of people posing for a photo

Description automatically generated with medium confidencePros:

1. I really like the aesthetic of this page. The black and white image is both sad and joyful and a stark contrast to the rest of the pages bright, almost pastel colour scheme.
2. Each piece of content takes up a good amount of space on the page.
3. Very inviting atmosphere.
4. Parallax scrolling background images are a neat touch.
5. Images pop really well or have a tint to them that enables text to be very readable on top of them.

###### Footer

Graphical user interface, website

Description automatically generated

Pros:

1. An exciting footer!
2. Content well-spaced out and structured.
3. Not overdone with links.
4. Links social media.
5. A close-up of several ties

   Description automatically generated with low confidenceI like the full-page footer as it doesn’t feel like an afterthought.

###### Overall Cons:

1. In mobile, the images can take up too much of the website. These could be condensed to a carousel or removed in smaller windows.
2. The Footer suffers a similar fate. In a smaller aspect ratio the footer takes a long time to scroll through and could have been more condensed for smaller displays.

A picture containing person, child, little

Description automatically generated

###### Blog

Graphical user interface, website

Description automatically generated

Pros:

1. The blog gives a real indication of what is happening right now at the charity group and gives a more grounded sense that they are working and active in the community.

Cons:

1. The latest blog was March 2020 and the previous blog to that was September 2018. Blogs need to be more frequent than this to have any significant reason to be on the company’s website. It has the opposite effect of showing an active community if they straight up aren’t active.

##### The Tidy Lady - <https://thetidylady.co.nz/>

###### Donations

Graphical user interface, text, application

Description automatically generated

Pros:

1. This page on The Tidy Lady lists many, many different organizations that take donations for all sorts of items. A page like this may be necessary on the proposed website in case people want to know where to donate different types of items and/or they are in a different area.

Cons:

1. It isn’t a pretty page. There are lots of links and nothing breaking them up.
2. There is too much word content for one page. Having sub pages navigable through headings or pagination could have saved this.

##### Competitive Analysis Summary

There are a lot of websites out there promoting their charities. Almost all of them seem to have a very similar layout with a few exceptions. The main page has a big photo and/or motto as the first thing people see. They largely include donation links that are static, so they are always present. A navbar along the top of the page with an about, contact and mission page. A lot had Te Reo for some of the major headings as well.

###### Summary of Pros:

1. A large image/carousel at the top of the main page with an appropriate caption is eye catching and informative and sets the mood for the rest of the site.
2. A static navbar with appropriate labels other main pages.
3. Te Reo translations of these labels.
4. A static donate button as a part of navigation.
5. Good colour choice is key, it will subconsciously make the viewer feel more, or less attracted to the charity cause.
6. Content that is not too cluttered and text rich. If content like this is required, break the page up with images and/or make them subpages.
7. A blog of current activity could promote the feeling that the charity is very active and making a difference.
8. A full-page footer with well-spaced content ties the whole page together nicely.

###### Summary of Cons:

1. Websites being cluttered.
2. Too many links.
3. Too many words not broken up.
4. Too many images on small aspect ratios.

### Site Content

#### Content and Functional Requirements

Based on the previous Competitive Analysis and Functional Requirements, multiple pages will be needing to be implemented to divide the required content and are listed below. Things to note are that each page will include a static header with navbar, and a footer at bare minimum which will be explained in more detail later. The Blog link in the navbar will simply link to the Blog which will be hosted on an external site.

##### Home:

Will feature a carousel or static image with the charities mission statement.

This will be followed by a brief blurb summarizing the goal of the charity and some of the work you have done and a button to the about page.

Below will be a donation content block that has a “Donate Now” button that takes the user to the Donate page and a “What can I donate” button which takes the user to the appropriate subpage in the About page.

Under this will be a poll that logged in users can vote in provided they have tokens.

##### About

It will have a side navigation that will act similar to a table of contents and take the user to the appropriate part on the page when clicked. The links will be labelled the same as the title they will be scrolling to.

The about us page will firstly have the static mission statement of the charity, followed by a more in-depth description of why the charity exists including the heart behind it and its purpose.

Below this will be a static brief history outlining some of the important dates and events in the charity’s history.

Then there will be a static section dedicated to the staff and volunteers.

After this will be the community section that links to other organizations in the area that do similar things or take and distribute items that Helping Families Nelson do not such as furniture.

##### Donate

This page will begin with a message of thanks with an image of a happy child or family behind the text. Below this will be the donate section which will have 4 list boxes, a dynamic picture import component, a text input and a submit and cancel button. The first list box will be “Item Type” and will contain an option for all currently supported items that can be donated as well as “Other”. The next list box will be “Item Size” and will adjust its contents to match that of the type selected. The next will be “Gender” which will have the options “Male”, “Female” and “Gender Neutral”. Then will be “Age ~” which will have the age brackets that Helping Families Nelson use to organize your stock. Then will be the section that handles uploads of pictures of the item. There will then be a text input for “Anything Else” that will allow the user to input a string of up to 255 characters. Finally will be the “Submit” button and the “Cancel” button that will either upload the item or cancel the upload and refresh the page.

##### Login

This page will be accessible through the “Sign Up / Register” button

The login page will simply have a text input for Email and Password with a “Cancel” and “Login” button that will either check the user exists and log in to their profile page or return to the home page as a default user.

##### Registration

The registration page will contain 7 text inputs, a component to upload an image and a “Cancel” and “Register” button that will refresh the page or perform a registration check respectively.

The text inputs will be:

* Name
* Email
  + This will be checked to see that it meets the conventional email format.
* Confirm Email
  + This will be checked to see that it matches that of the previous input.
* Password
  + This must be greater than or equal to 8 characters and contain one symbol.
* Confirm Password
  + This must match the previous input.
* Address
* Phone Number
  + This must match the standard landline or mobile conventions.

##### Profile

This page will be accessible once the user is logged in and the Sign Up / Registration

The profile page will be dynamic and show the users Profile Picture with a “Change” button below it, allowing the user to change their profile image. There will also be their Name, Email, Address, Phone Number and Tokens, all labelled and listed in text input fields that cannot currently be changed. There will be an “Edit” button which will make the text input fields editable and a “Cancel” button that is greyed out until “Edit” is clicked. The text inside the “Edit” button will also change to confirm when “Edit” is pressed.

#### Group and Label Your Content

The above pages cover all of the functional requirements for the website.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Home | About | Profile | Registration | Login |
| Carousel with text overlayed and buttons to navigate | Motto with underlying, emotional image | Carousel with slides linking to appropriate featured event | Motto | Username and password input |
| About section summary with link to about page | Introduction to what the purpose of the charity is. | Five events linking to appropriate event page | Usual Routine/About Detonate description | Login Button |
| Donate section with link to Donate page | History of the charity |  |  |  |
| Poll Section that requires users’ tokens and login | Staff members listed. (Potentially dynamically generated) |  |  |  |
|  | Community links and brief summary of each |  |  |  |

### Site Structure

#### Metaphors

##### Organisational Metaphor

Organizational metaphors are used to create the layout of a website. They take examples from how a real-life scenario is organized and convey that across a website. This website will use the organizational metaphor of a supermarket. There will be the main navigation headings across the top of the page, like the isle numbers and headings you would see at the supermarket. Once you have navigated to this place, you would see sections along each isle telling narrowing down the type of item stored there to be more specific, as with the website. Then there would be the items themselves, laid out so that they are easy to see and grouped together surrounded by similar items. The site will mimic this too. The same metaphor can be used for a user uploading an item as they narrow down what the item will be used for until it is specific enough to upload.

##### Functional Metaphor

A functional metaphor, in this instance, is used to relate how the website will function in relation to a real-life scenario. The act of donating is based off the system that you are currently using yourselves. It starts by you having the product, then grouping it into its type, size, gender, and age. This is the primary functional metaphor of the site.

##### Visual Metaphor

A visual metaphor is used to make an object on a website appear as an object in real life. Websites use a lot of these to make their visuality much more user friendly and intuitive.

The visual metaphors that will be used in this site are as follows:

* Icons
  + To identify what that button will do without needing to read it. For example, a House indicates that that is the home page. A person indicates their profile, and X indicates cancel and a tick indicates confirm as these are related to real life objects.
* Carousel
  + This is like a photo album. People know to click the arrows to scroll through and expect to see a new image.

#### Site Structure Listing

1. Home Page
   1. Navbar – Has these links: About, Community, Blog, Contact, Helping Families Nelson Logo (link to Home Page), Sign Up / Register Button and Donate Button. This Allows the user to navigate the website.
   2. Carousel of featured items. Gives the user a feel for the heart of the site.
      1. Image
      2. Text
   3. About – Shows a synopsis of what the site and charity is for.
      1. Headline
      2. Text
      3. (Probably an image)
   4. Donate Panel
      1. Headline
      2. Description
      3. Donate Button
      4. What Can I Donate Button – Links to the about page where the user will be informed on what they can donate.
   5. Poll – Allows logged in users to use their tokens to vote for a cause the charity is looking to support.
      1. Image x3
      2. Description x3
      3. Radio Button x3
      4. Commit Button – Commits their vote to the poll and reduces tokens by 1.
      5. Tokens – Shows the number of tokens left for the user.
   6. Footer
      1. Links for about
      2. Links for Contact Details
      3. Map showing location – So people know where to go. Includes the address in text form next to the embedded map.
      4. Location Details
2. About Page
   1. Navbar – Has these links: About, Community, Blog, Contact, Helping Families Nelson Logo (link to Home Page), Sign Up / Register Button and Donate Button. This Allows the user to navigate the website.
   2. Charity Motto
      1. Image
      2. Text
   3. About – Shows a synopsis of what the site and charity is for.
      1. Headline
      2. Text
      3. (Probably an image)
   4. History
      1. Headline
      2. Text
      3. (Probably an image)
      4. Date
   5. Staff
      1. Staff Members x3
         1. Image of staff member
         2. Name
         3. Role
         4. Quote
         5. Short Description of the reason they volunteer.
   6. Community
      1. Links and descriptions to various community groups
   7. Donate Panel
      1. Headline
      2. Description
      3. Donate Button
      4. What Can I Donate Button – Links to the about page where the user will be informed on what they can donate.
   8. Poll – Allows logged in users to use their tokens to vote for a cause the charity is looking to support.
      1. Image x3
      2. Description x3
      3. Radio Button x3
      4. Commit Button – Commits their vote to the poll and reduces tokens by 1.
      5. Tokens – Shows the number of tokens left for the user.
   9. Footer
      1. Links for about
      2. Links for Contact Details
      3. Map showing location – So people know where to go. Includes the address in text form next to the embedded map.
      4. Location Details

3 Contact

* 1. Navbar – Has these links: About, Community, Blog, Contact, Helping Families Nelson Logo (link to Home Page), Sign Up / Register Button and Donate Button. This Allows the user to navigate the website.
  2. Contact Us
     1. Contact Us Header
     2. Phone Number
     3. Email
     4. First Point of Contact – Links to appropriate staff member on about page.
     5. Location Header
     6. Map plugin
     7. Location Details
  3. Footer
     1. Links for about
     2. Links for Contact Details
     3. Map showing location – So people know where to go. Includes the address in text form next to the embedded map.
     4. Location Details

1. Login Page
   1. Navbar – Has these links: About, Community, Blog, Contact, Helping Families Nelson Logo (link to Home Page), Sign Up / Register Button and Donate Button. This Allows the user to navigate the website.
   2. Username/Password input fields – Input fields where a user can input their details to log in.
   3. Login Button – Button user can press to login.
   4. Cancel Button – Button that takes user back to home page.
   5. Footer
      1. Links for about
      2. Links for Contact Details
      3. Map showing location – So people know where to go. Includes the address in text form next to the embedded map.
      4. Location Details
2. Registration Page
   1. Navbar – Has these links: About, Community, Blog, Contact, Helping Families Nelson Logo (link to Home Page), Sign Up / Register Button and Donate Button. This Allows the user to navigate the website.
   2. Register Content Block
      1. Heading
      2. Profile Image
      3. Name Input
      4. Email Input
      5. Confirm Email Input
      6. Password Input
      7. Confirm Password Input
      8. Address input
      9. Phone Number Input
      10. Cancel Button – Refreshes page data.
      11. Register Button – Performs email, password and various other checks and if passes, the user is registered and taken to the home page but logged in.
   3. Footer
      1. Links for about
      2. Links for Contact Details
      3. Map showing location – So people know where to go. Includes the address in text form next to the embedded map.
      4. Location Details
3. Profile Page
   1. Navbar – Has these links: About, Community, Blog, Contact, Helping Families Nelson Logo (link to Home Page), Sign Up / Register Button and Donate Button. This Allows the user to navigate the website.
   2. User Profile Content Block
      1. Heading
      2. Profile Image
      3. Change Button – Allows user to change profile image.
      4. Name
      5. Email
      6. Address
      7. Phone Number
      8. Tokens
      9. Edit Button – Allows Name, Email, Address and Phone Number to be altered.
      10. Confirm Button – Checks changes and commits or rejects them.
      11. Cancel Button – Refreshes page to previous state.
      12. Location Details
   3. Poll – Allows logged in users to use their tokens to vote for a cause the charity is looking to support.
      1. Image x3
      2. Description x3
      3. Radio Button x3
      4. Commit Button – Commits their vote to the poll and reduces tokens by 1.
      5. Tokens – Shows the number of tokens left for the user.
   4. Donate Panel
      1. Headline
      2. Description
      3. Donate Button
      4. What Can I Donate Button – Links to the about page where the user will be informed on what they can donate.
   5. Footer
      1. Links for about
      2. Links for Contact Details
      3. Map showing location – So people know where to go. Includes the address in text form next to the embedded map.
4. Donate Page
   1. Navbar – Has these links: About, Community, Blog, Contact, Helping Families Nelson Logo (link to Home Page), Sign Up / Register Button and Donate Button. This Allows the user to navigate the website.
   2. Thankyou Message
      1. Image
      2. Text
   3. Donate Component
      1. Headline
      2. Item Type List Box
      3. Item Size List Box
      4. Gender List Box
      5. Age ~ List Box
      6. Upload Button – allows user to upload an image.
      7. Remove Button – Allows a user to remove an uploaded image.
      8. Anything Else Input
      9. Submit Button – Submits item to donate.
      10. Cancel Button – Resets page.
   4. Footer
      1. Links for about
      2. Links for Contact Details
      3. Map showing location – So people know where to go. Includes the address in text form next to the embedded map.
      4. Location Details

#### Architectural “Blue-Print”

Graphical user interface

Description automatically generated

#### Define Navigation

##### Global Navigation

The global navbar will be primarily responsible for the navigation of the website as it will always be present at the top of the website. It will have links to the following:

* About
* Community
* Contact
* Home
* Login
* Donate

The secondary global navigation method will be the footer. It will have links to all the above pages as well as specific links to the Staff section in About, and the Location section in Contact.

##### Local Navigation

The About page will have local navigational functionality in the form of the side bar. It will scroll to the appropriate section of the page that corresponds with the label. These are:

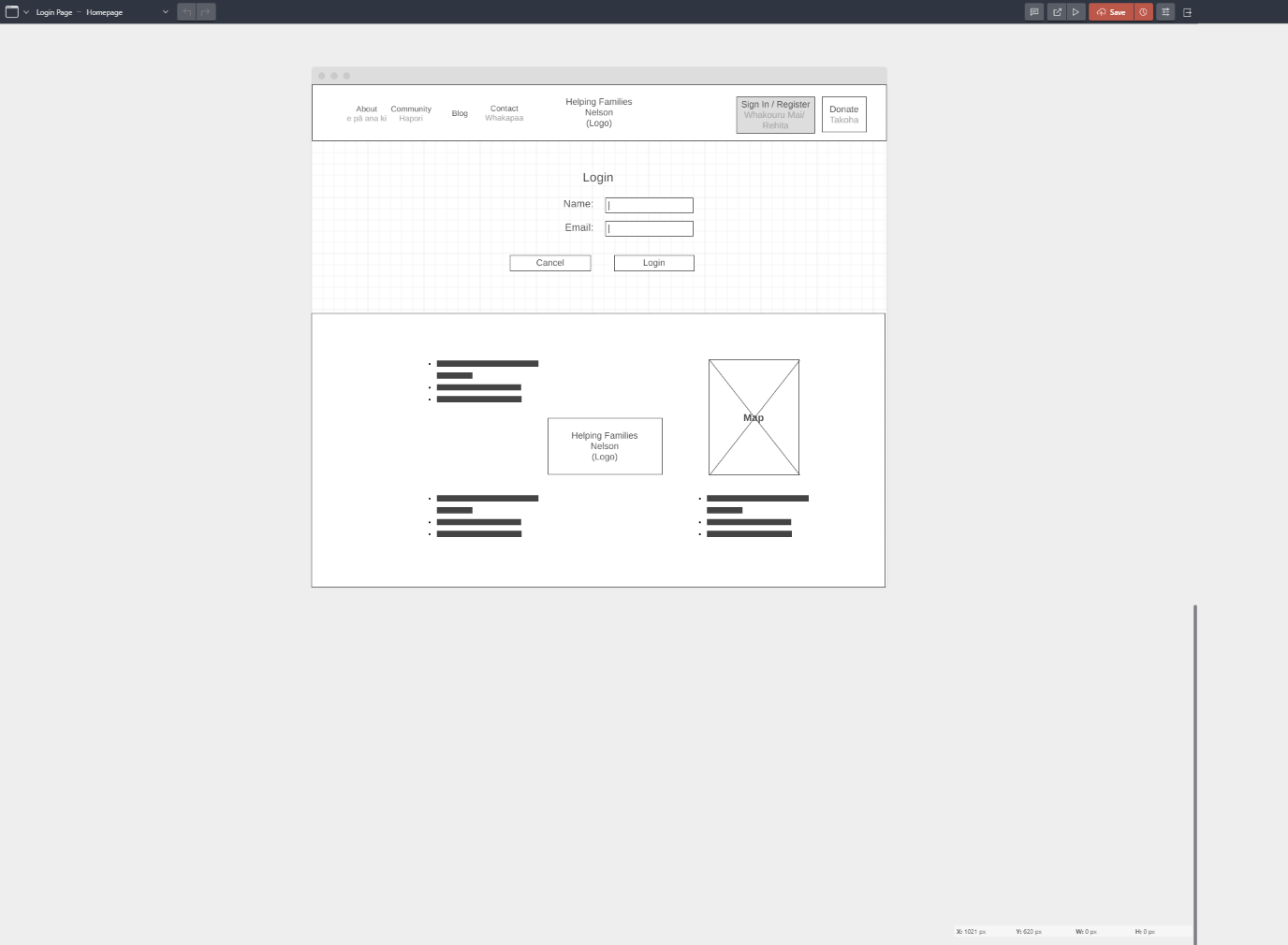
* About
* History
* Staff
* Community

#### Visual Design

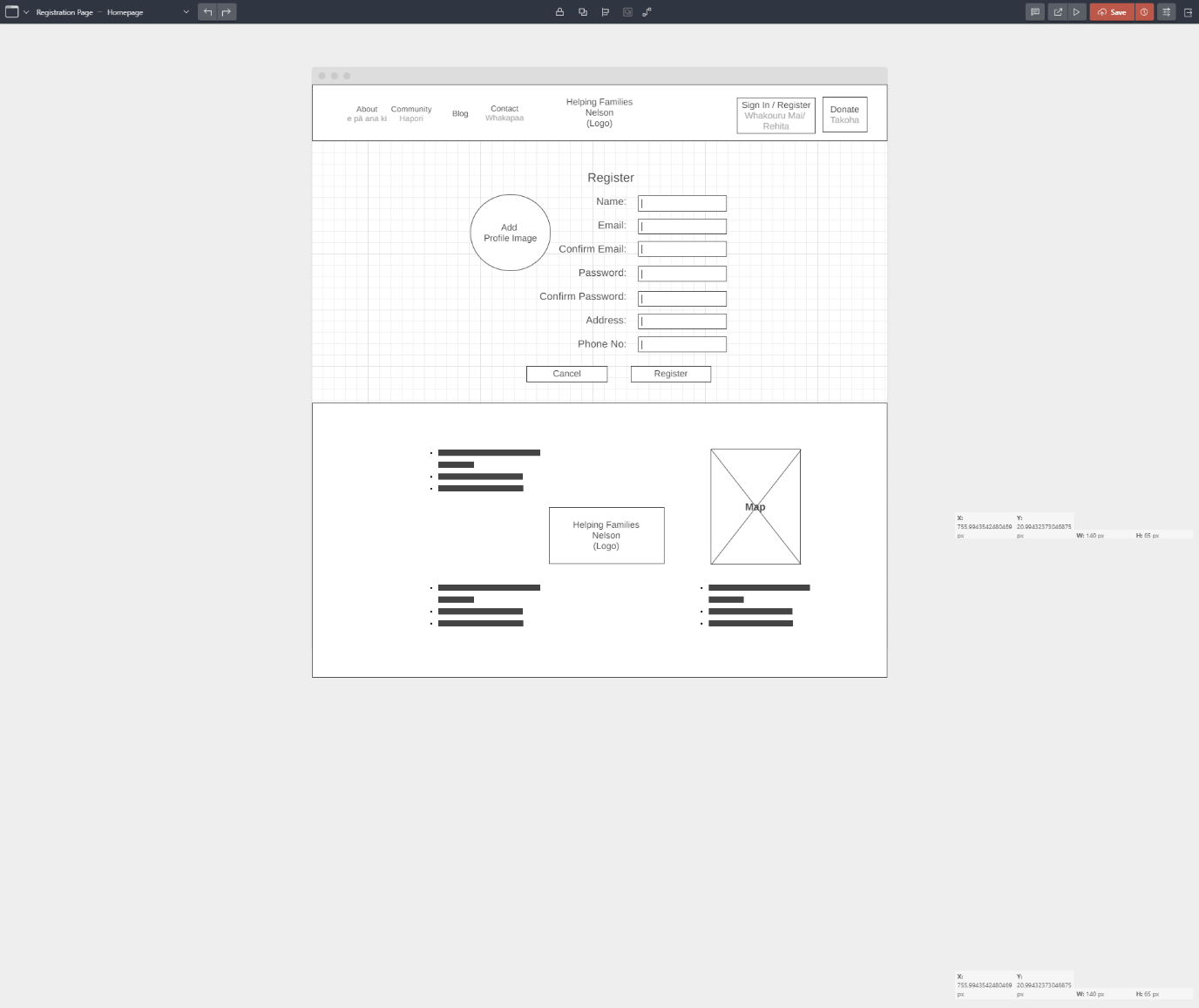
##### Home Page



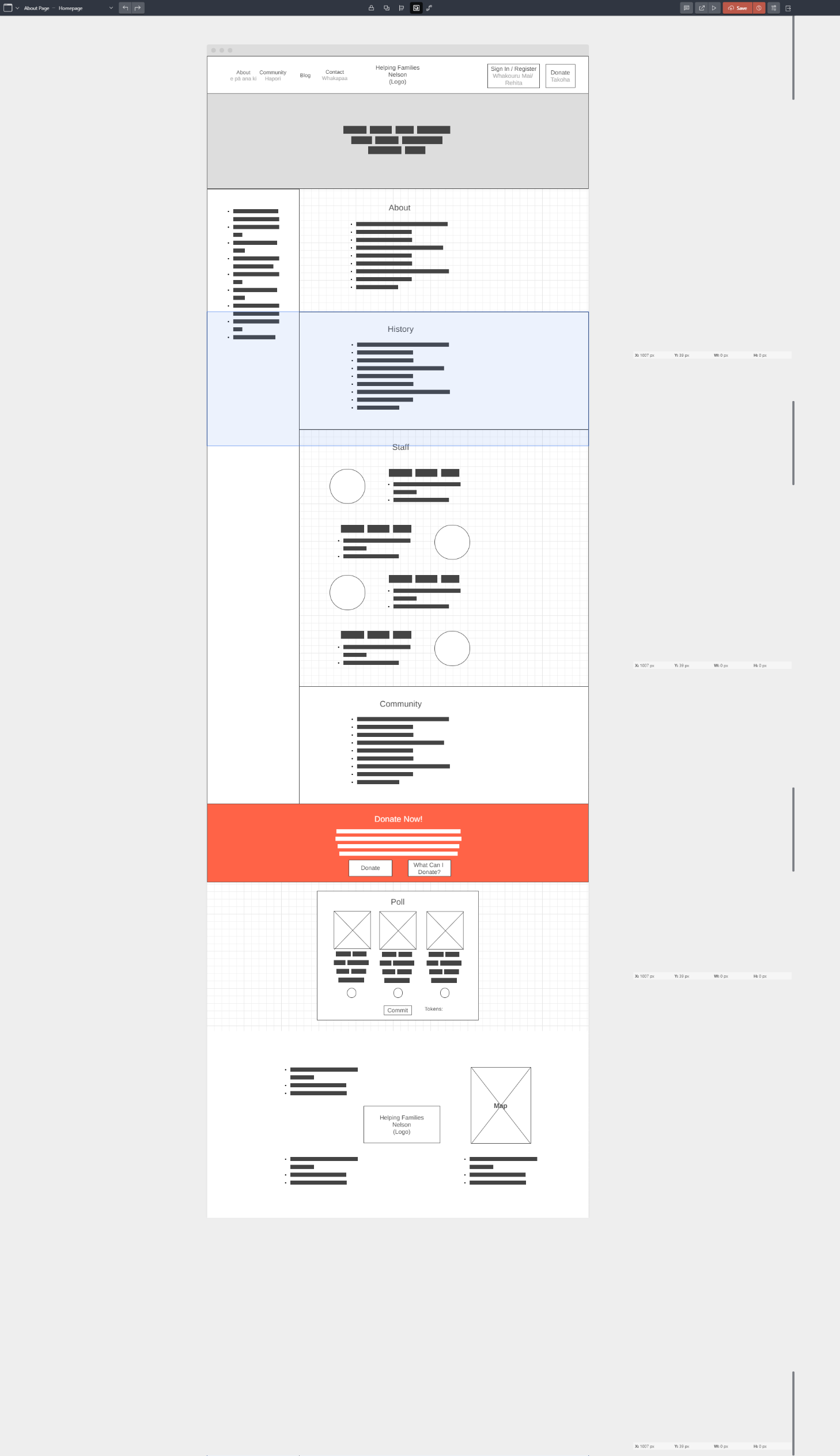
##### Login Page



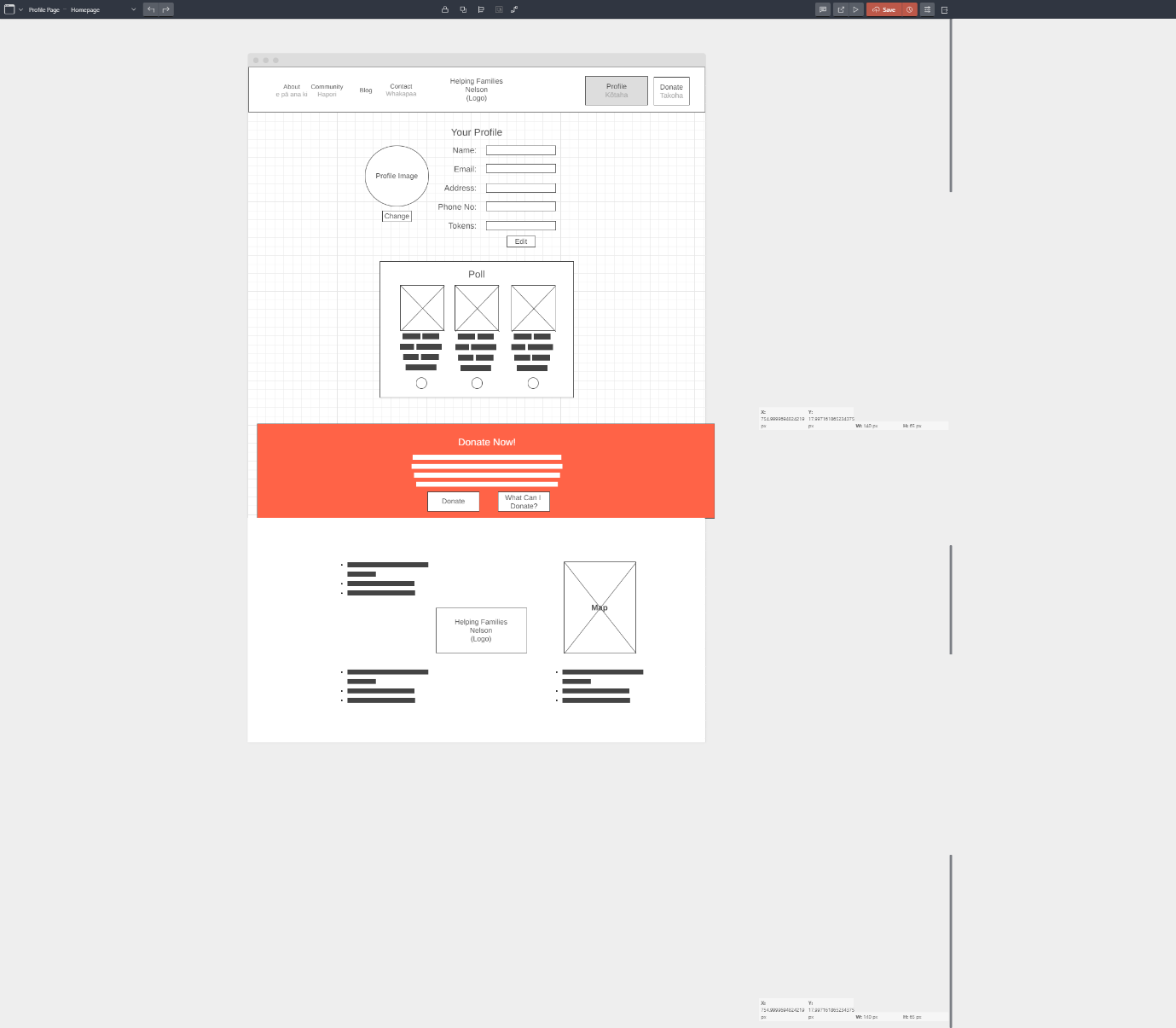
##### Registration Page



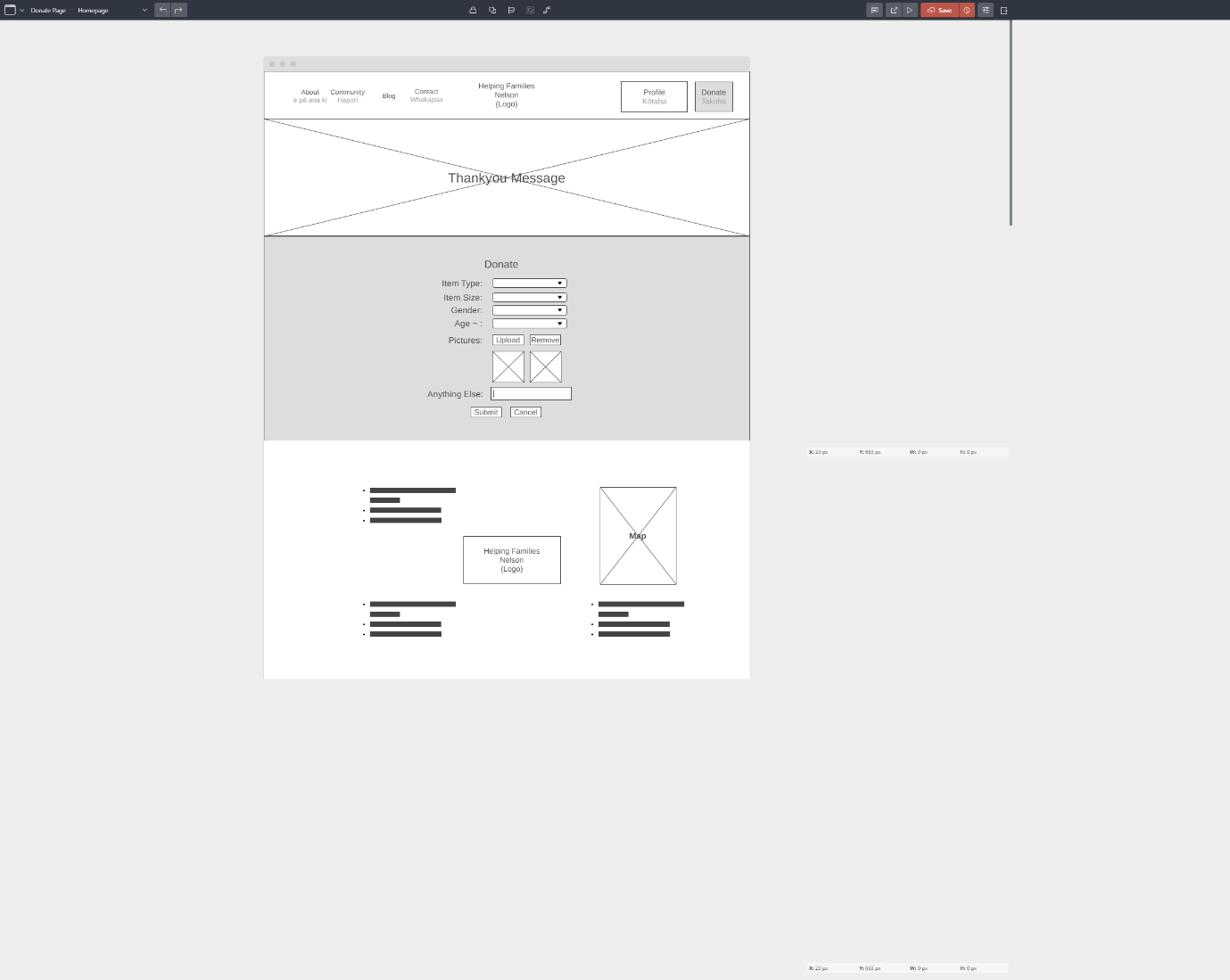
##### About Page



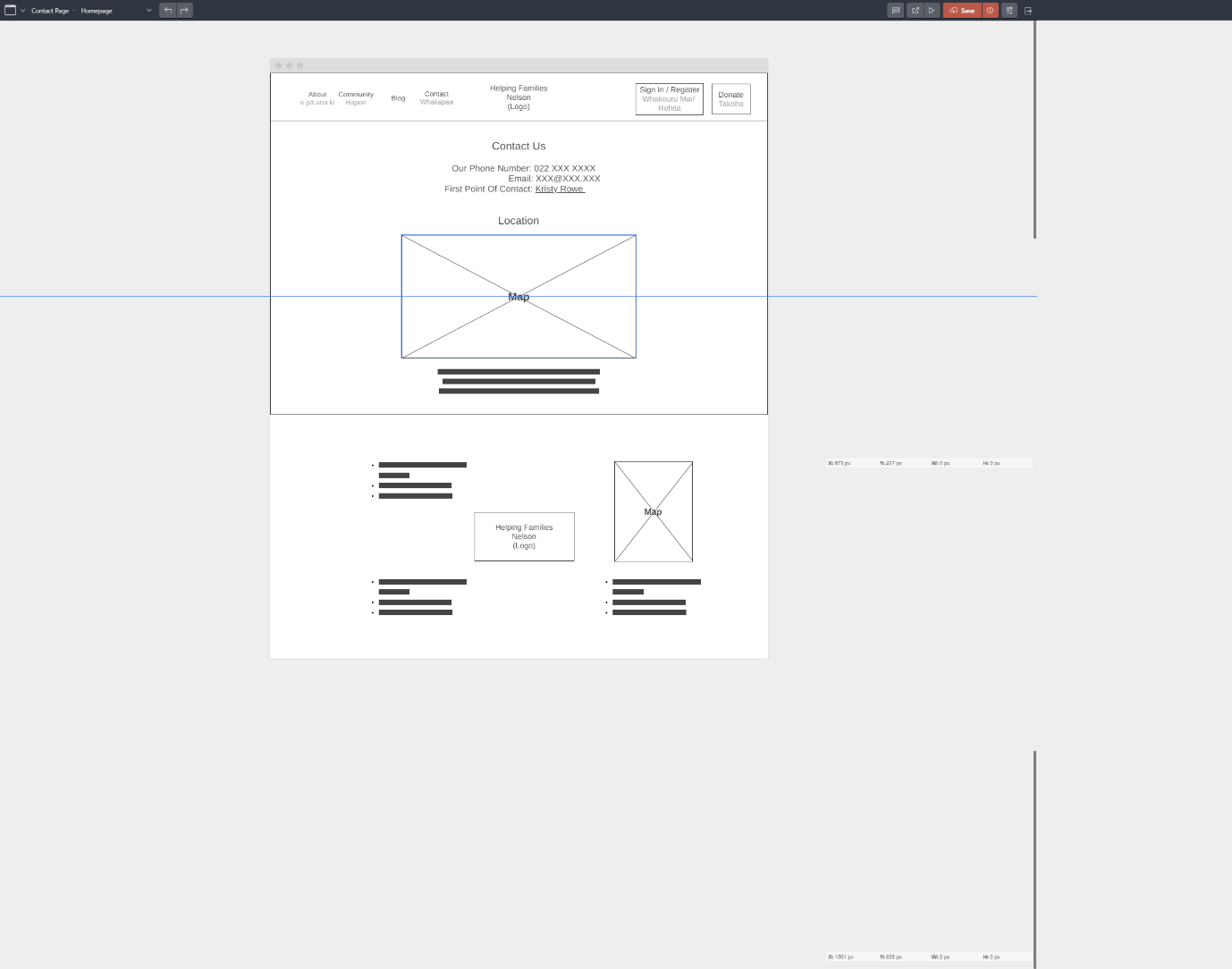
##### Profile Page



##### Donate Page



##### Contact Page



## USER STORIES – DESIGNING TO BUILD

### CRUD Analysis

# Part Two – Evaluate

## The Purpose of Web Frameworks

Coding as website strictly using PHP (the default way a website can be created) is possible but can cause many different problems. The first of these is that all of the code from the website resides on one single page which makes the coding very messy and disorganized and hard to find what you are looking for. You can also end up with chunks of code that produce the same or a very similar result that another chunk of code is already doing making a lot of the code redundant. There is little reusability in this as well. When the web app gets larger, making changes can be very difficult as in changing it will likely result in another piece of code no longer operating successfully. Testing code and implementing a means of authentication also become extremely difficult as the system is unorganized, and the use of built-in plugins is not supported as it might be for a web framework. Working on the website with others becomes increasingly difficult as all the code is on the same document so you will likely end up altering someone else’s code that can cause issues for them.

When using web frameworks, the entire project becomes far more organized and allows for reuse of code and a far better structure for teams to work on than simply using PHP. Most frameworks come with plenty of built-in features that can take some unnecessary strain off the developer from no longer needing to also code these as well. Changing anything becomes far easier and altering a module will replicate that change across all instances of it removing the need to change every module individually.

There are, however, situations where using a web framework simply adds unnecessary bloat to a web app. This is only on very small and basic web apps however where they do not need repeating components and functionality.

- (Why Use a Framework?)

For the purpose of this web application being developed for Helping Families Nelson, a web framework will be extremely beneficial as there are several components that will be repeated across several pages and a system to link to a database. For this project Angular and Vue will be two frameworks that will be assessed for their suitability for the creation of the Helping Families Nelson web app and the most suitable will be the chosen framework to fully develop this web app.

The components that will be needed to be accessible to multiple views are:

* Polls.
* Carousels.
* About section.
* Donate section.

Data that will be needed to be accessible to many parts of the app:

* User.
* Roles.
* Items.

## Website Requirements:

\*\*\*Note\*\*\*

= Indicates most important files used to perform action

### 1. Charity members and beneficiaries can register, log in and administer their own accounts.

#### Angular:

##### Register:

A screenshot of a computer

Description automatically generated with medium confidenceGraphical user interface, text, application

Description automatically generatedCode: View:

A picture containing diagram

Description automatically generated

###### MongoDB created new User:

Graphical user interface

Description automatically generated with medium confidence

###### Description:

This panel can be accessed through the navbar from any page as long as there is not currently a user logged in by pressing Signup. The user inputs their details into the form and clicks the “Sign Up” button.

Provided that the data entered into the form (register.component.html 8) is valid, e.g. the email entered matches the format of an email, this data is passed to the authService (register.component.ts 29). This data is then sent to the backend using the http link for auth (auth.service.ts 26) backend server component which is linked through the store. If there is no username or email currently in the database equal to these two values, this data is entered into the database (auth.controller.js 22).

##### Login:

A picture containing diagram

Description automatically generatedChart

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidenceCode: View:

###### Description:

This panel can be accessed through the navbar from any page as long as there is not currently a user logged in by pressing Login. The user inputs their details into the form and clicks the “Login” button.

Provided that the data entered into the form (login.component.html 8) is valid, this data is passed to the authService (login.component.ts 34). This data is then sent to the backend using the http link for auth (auth.service.ts 18) backend server component which is linked through the store. If the username and password both match a single entry in the db, the user data of this entry is returned minus the password (auth.controller.js 75).

##### Graphical user interface, text, application Description automatically generatedProfile:

Code: View:  
Text

Description automatically generated

###### Description:

This page shows the users information. It has received this information from the getUser command found in services/token-storage-service (profile.component.ts 15).

#### Vue:

##### Register:

###### Text Description automatically generatedCode: View:

A picture containing chart

Description automatically generatedGraphical user interface, application

Description automatically generated

###### MongoDB created new User:

###### Graphical user interface, text, application Description automatically generated

Description:  
This panel can be accessed through the navbar from any page as long as there is not currently a user logged in by pressing Signup. The user inputs their details into the form and clicks the “Sign Up” button.

Provided that the data entered into the form (Register.vue 9) is valid, as checked by the schema at (Register.vue 86) this data is passed to the store (Register.vue 139). This data is then sent to the store auth.module.js at line 28 where the data is then passed to auth.service.js at line 31. The username, password, email, phone, address and user type are then sent to the backend to auth.controller.js at line 11. This creates a new const based on the User mongoose schema and sets the values to that of the req payload (line 13). The user is then saved and given its role based on the information given in role of user.

##### Login:

Graphical user interface, application

Description automatically generatedA picture containing graphical user interface

Description automatically generatedChart

Description automatically generated with low confidence Code: View:

###### Description:

This panel can be accessed through the navbar from any page as long as there is not currently a user logged in by pressing Login. The user inputs their details into the form and clicks the “Login” button.

Provided that the data entered into the form (Login.vue 9) is valid, as checked by the schema at (Login.vue 53) this data is passed to the store (Login.vue 82). This data is then sent to the store auth.module.js at line 12 where the data is then passed to auth.service.js at line 8. The username and password are then sent to the backend to auth.controller.js at line 77. The username and password are then checked to see if that user/password combination exists in the database. If so, that user is retrieved, and its attributes sent back to the frontend. If not, an error is returned. Once the user is logged in, they are pushed to the profile page.

##### Profile:

###### Code: View:

A picture containing graphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generated

Description:

This page shows the users information. It has received this information from the currentUser compute (profile.vue 52).

#### Comparison:

This comparison is hard to make as for the most part I followed tutorials by bezkoder (bezkoder, 2019 & 2020) for implementing these steps. I added a phone, address and tokens atttribute and each was implemented and functions very similarly, and I had little issues getting either to operate successfully. I did however add two custom roles to the Vue framework: Charity and Beneficiary. These I was able to fully implement in under five minutes as the whole process was extremely straight forward. For this alone I will say that Vue was the better choice. While both seem to function very similarly, the ease at which I was able to achieve adding these roles I was not able to replicate on the Angular framework.

### 2. Register Product & Acquire Tokens

#### Angular:

##### Register Product:

Text

Description automatically generatedGraphical user interface, text, application

Description automatically generatedCode: View:  
Graphical user interface, application

Description automatically generated

###### MongoDb added item:

Graphical user interface, text, application, email

Description automatically generated

###### Description:

This component is found on the Donate page on the navbar if there is a user logged in (donate.component.html 1). It takes the data entered in the form (upload-item.component.html 3) provided that there is data entered for the category name and size (upload-item.component.ts 22) and this data is then sent to the authService under the method name “upload” (upload-item.component.ts 114). This data is then sent to the backend using the http link for auth (auth.service.ts 37) backend server component which is linked through the store. If a user is found in the db matching the username of the req body, the rest of the data is pushed into the table where the username matches in the form of a const newItem matching the Item schema. A token is added to this user, it is saved in the db and then the users tokens are returned as well as the message “Token Earned” (auth.controller.js 128-152).

#### Vue

##### Register Product:

###### Text Description automatically generatedCode: View:

Graphical user interface, text, application, chat or text message

Description automatically generatedGraphical user interface, text, application

Description automatically generated

###### MongoDB created new item:

###### Graphical user interface, text, application, email Description automatically generated

Description:  
This panel can be accessed through the navbar from any page as long as user is logged in and is also of the type “Charity” (Donate.Vue 3, 143) and is accessed by pressing Donate. The user inputs their items details into the form and clicks the “Upload” button.

Provided that the data entered into the form (Donate.vue 4) is valid, as checked by the schema at (Donate.vue 102) this data is passed to the store (Donate.vue 161). This data is then sent to the store auth.module.js at line 40 where the data is then passed to auth.service.js at line 44. The user, categoryName, size, gender, age and other are then sent to the backend to auth.controller.js at line 133. This creates a new const based on the Item mongoose schema and sets to the values to that of the req payload minus the user (line 135). The user is then searched using the submitted user’s username. The user’s tokens are increased by one and the item is pushed into the user and it is saved. The updated user is returned with the message that then changes the state to be successful and the “Token Earned” view shows. Upon clicking the “Yes” button the view reverts to its original.

#### Comparison:

For this comparison I find hard to remain just and objective to only the frontend development. As Angular was the first framework I implemented, it meant that I had to get the backend functioning simultaneously where for Vue the backend was already mostly complete. This implementation took several days to complete in Angular and a matter of hours in Vue. While this is the case, I found that through “yup” schemas, Form handling was exceptionally easy in Vue when compared to Angular. I will award it as being the least complex and easiest way of implementing these modules out of the two though as I cannot ignore that it took at least 10 hours less to complete.

### 3. Interactive Elements:

#### Angular

##### Carousel: (Angular Powered Bootstrap, n.d.)

###### Code: View:

A picture containing text, tree, outdoor, bridge

Description automatically generatedText

Description automatically generated

Description:  
Currently this carousel is more of a proof of concept. It finds images from <https://picsum.photos/id/> based on the number in the array (carousel.component.ts 10). Using the bootstrap built in functions it then generates the carousel that can be controlled by pressing the buttons on the side and/or the item select at the bottom middle of the carousel (carousel.component.html 1-30).

Gallery: (Sousli, 2016/2021)

###### A picture containing plant, flower, close, arranged Description automatically generatedA child walking on a sidewalk Description automatically generated with low confidenceCode: View:

Text

Description automatically generated

Description:  
This gallery is also a proof of concept. When the gallery is clicked it opens a window that overlays the rest of the page and allows the user to cycle through the images in this new panel by using the thumbnails at the top or the navigational buttons either side. It can be closed by pressing the X in the top right.

#### Vue

Unfortunately, I ran out of time for the Vue additions. The Poll is reactive and when hovered over changes slightly.

#### Comparison:

As I have no comparison between Vue and Angular for this I will have to give it to the Angular framework. All in all, I found it very easy to import modules that had been created by someone else and are available to edit and upload yourself. There is a vast library of these making it quite user friendly.

### 4. Token Interface

#### Angular

Timeline

Description automatically generated with low confidenceA picture containing timeline

Description automatically generatedText

Description automatically generatedRedeem Token:  
Code: View:

Graphical user interface, text, application, email

Description automatically generated MongoDb token is redeemed:

###### Description:

This component will be available on many pages but is currently only being used by the home page. The poll handles requests to vote for a charity to support and subtracting a token from the user which acts like a payment method to vote. When a “Vote” button is clicked (poll.component.html 27, 48, 67), the value assigned to it will be passed and the vote method called (poll.component.ts 22)

This the authService function “redeem” is called and passed this information (poll.component.ts 24). This function (auth.service.ts 49) will send an http post request and include in the payload the username and value submitted. The auth controller will find the user matching the username submitted and if found reduce their tokens by 1 as payment for the vote and return the users token amount and success message (auth.controller.ts 163).

#### Vue

##### Redeem Token:

###### Timeline Description automatically generatedTimeline Description automatically generatedCode: View:

A picture containing diagram

Description automatically generated

Description:  
This panel can be accessed through the navbar from any and is accessed by pressing Home. The user can choose to support a local charity by voting which redeems one of their tokens when they press any “Vote” button which also passes that charity value.

Provided that the user has tokens spare and is logged in, the users data is passed to the store (Poll.vue 103). This data is then sent to the store auth.module.js at line 52 where the data is then passed to auth.service.js at line 62. The username is then sent to the backend to auth.controller.js at line 176. This checks the db for a user matching that with the username supplied (auth.controller.js 178). Once found, if the users tokens are greater than or equal to 1, their tokens are reduced by one and the user is saved and returned along with a success message. If the user does not have enough items a failure message will be returned. The users’ tokens at the bottom will then change to match this new value.

#### Comparison:

The two frameworks both had features that were more beneficial than the other for this example. In Vue, the Poll module was substantially easier to create and import into the home page as a module, however I had some difficulty in getting the screen to refresh even using this.$forceUpdate() and setting a key that changed which is meant to reload that element. In Angular, the users’ tokens were able to be managed more fluidly and update the screen, however the implementation of the Poll module I found more aggravating than in Vue. Purely based on my levels of sanity I prefer the Vue framework for this as I spent many hours trying to implement this module in Angular and only one in Vue.

### 5. Store and Retrieve data from a server-side database

Besides [3. Interactive Elements](#_3._Interactive_Elements:), each of the previous steps in the rubric contain examples of data being passed to, or from the database.

Here are the links if you need to be refreshed:

* [Register](#_Register:)
* [Login](#_Login:)
* [Upload](#_Register_Product:)
* [Redeem](#_Angular)

## Recommendation of Framework (Conclusion):

### Angular

Pros:

* An older, more fleshed out and supported framework with many built in features.
* Has more support and forums online.
* Overall, more popular.
* In my experience the built-in features work all the time.
* I found it very logical once I wrapped my head around it.

#### Cons:

* Heavy on the system which may not pay off for smaller apps.
* Very steep learning curve.
* Comparatively slow when running.
* Did not “Hot-Load” constantly meaning a manual reboot of the frontend was required.

### Vue

#### Pros:

* Lightweight framework, less unnecessary bloat.
* Far easier to learn.
* Far quicker implementation.
* Preferred file/folder structure.
* Template, script, and style in the same document made passing data easy.
* Preferred the store in Vue to service in Angular.
* Forms were much easier to error check.
* Never had to reboot due to not being up to date with code.

#### Cons:

* Newer, less online support
* Less fleshed out with fewer built-in features
* reRenderKey did not function as intended to update the currentUser (Poll.vue 123)

### Decision

I have decided that, after weighing up the pros and cons I have come across of each, Vue will be the framework that I choose going forward. It is newer and more exciting to learn, the file/folder structure I find easier to use and the ease of which forms were implemented and how much sense certain aspects made when compared to Angular leaves me no choice but to go with Vue. I did not dislike Angular, it had its benefits and was a good framework, but Vue being much easier to learn and understand is a very big deal as it saves a lot of time that would otherwise be spent trying to figure out how certain things operate.

## References

The Salvation Army. (n.d.). Retrieved March 23, 2021, from <https://www.salvationarmy.org.nz/>

The Clothing Project. (n.d.). The Clothing Project. Retrieved March 23, 2021, from <https://www.theclothingproject.co.nz>

Donations | The Tidy Lady. (n.d.). Retrieved March 23, 2021, from <https://thetidylady.co.nz/useful-links/donations/>

Donate goods to our shops | New Zealand Red Cross. (n.d.). Retrieved March 23, 2021, from <https://www.redcross.org.nz/shop-with-us/red-cross-shops/how-donate-our-shops/>

Wireframe.cc | The go-to wireframing tool. (n.d.). Wireframe.Cc. Retrieved March 27, 2021, from <https://wireframe.cc/LeFa8l>

Why use a framework? (n.d.). Lynda.Com - from LinkedIn. Retrieved May 31, 2021, from <https://www.lynda.com/PHP-tutorials/Why-use-framework/540346/579816-4.html?srchtrk=index%3a5%0alinktypeid%3a2%0aq%3aLaravel%0apage%3a1%0as%3arelevance%0asa%3atrue%0aproducttypeid%3a2>

Sousli, M. (2021). MurhafSousli/ngx-gallery [TypeScript]. <https://github.com/MurhafSousli/ngx-gallery> (Original work published 2016)

Angular powered Bootstrap. (n.d.). Ng-Bootstrap. Retrieved June 1, 2021, from <https://ng-bootstrap.github.io>

bezkoder. (2019, October 18). Vue.js JWT Authentication with Vuex and Vue Router. BezKoder. <https://bezkoder.com/jwt-vue-vuex-authentication/>

bezkoder. (2019, October 19). Node.js + MongoDB: User Authentication & Authorization with JWT. BezKoder. <https://bezkoder.com/node-js-mongodb-auth-jwt/>

bezkoder. (2020, August 2). Angular 11 JWT Authentication example with Web Api. BezKoder. <https://bezkoder.com/angular-11-jwt-auth/>