ASSESSMENT 1

WEB701

Ezekiel P. Brown
NMIT TePukenga

NMIT SN - 13491119

23/06/23

Site Goals

Volunteer Work: Volunteer your time and skills to local businesses and organizations. People can post listings that they need help with. People can volunteer to help using their special trade and skills.

Charity Name: Hands for Hope

Mission or purpose

Our mission at Hands for Hope is to bring a community closer than ever before. By connecting local individuals with local businesses through their trade and skills. This is made possible with the Hands for Hope website.

Short and long-term goals

Short-term goals:

Have a minimum of 20 users volunteering over the next 3 months
Establish partnerships with local businesses and organizations within the next 6 months to provide volunteer support.
Develop a website and social media presence to promote our mission and connect with potential volunteers and partners within the next 2 months.

Long-term goals:

At least 200 active community members within the next 2 years.
Establish a relationship with 20 local businesses.
Increase community awareness and support for our mission.
Expand to other regions, then countries.

Intended Audiences

Volunteers: Individuals who are interested in volunteering their time and skills to support local businesses and other members in need.

Local Businesses and Organizations: Businesses and organizations within the local community who need support and are interested in partnering with volunteers to enhance their operations.

Community Members: Residents within the local community who are interested in supporting local businesses, organisations, and other members of the community.

Donors and Sponsors: Individuals or businesses who are interested in supporting the mission of Hands for Hope.

Why will people come to the website?

To learn more about the mission and work of "Hands for Hope": Visitors may be interested in learning more about “Hand of Hope”, its goals, and the impact it's making in the community.

To find volunteer opportunities. Individuals who are interested in volunteering may visit the website to explore the various volunteer opportunities available, and to learn how they can get involved.

To find local businesses and organizations in need of support. Local businesses and organisations who are seeking volunteer support may visit your website to learn more about how they can connect with volunteers through "Hands for Hope."

To donate or sponsor. Individuals or businesses who are interested in supporting "Hands for Hope" financially may visit your website to make donations or learn more about sponsorship opportunities.

To connect with the community. The “Hands for Hope” website may feature somewhere where they can communicate with each other.

Define the user experience

Define the audience

The audience for our charity range from all sorts of people from local communities. From start-ups to well-established businesses, from builders to doctors. Anyone with a particular set of skills is more than welcome.

Create scenarios

John wants to join Hands for Hope and start helping others.

John is new in town and is looking for ways to get involved in his community. He has heard about Hands for Hope and thinks it might be a great way to meet people and make a difference.

1. John goes to the Hands for Hope website and clicks on the "Help now" button.
2. He's taken to a registration page where he enters his name, and email address, and creates a password for his account.
3. John sees that there are several requests for help posted on the website, such as a request to help clean a local park.
4. He decides to volunteer and clicks on the "Volunteer" button.
5. John sends a message to the person who posted the request.
6. The person who posted the request responds to John's message and they arrange a time and place to meet and start the task.
7. John goes to the park at the scheduled time and meets the person who posted the request. Together, they work to clean up the park.
8. Once the job is finished, John feels great about having made a positive impact in his community and looks forward to meeting other people who are in need of help.

Sam needs help fixing his fence.

Sam has a fence in his backyard that fixing. Sam is unsure on how to fix it, he posts a job on Hands For Hope website

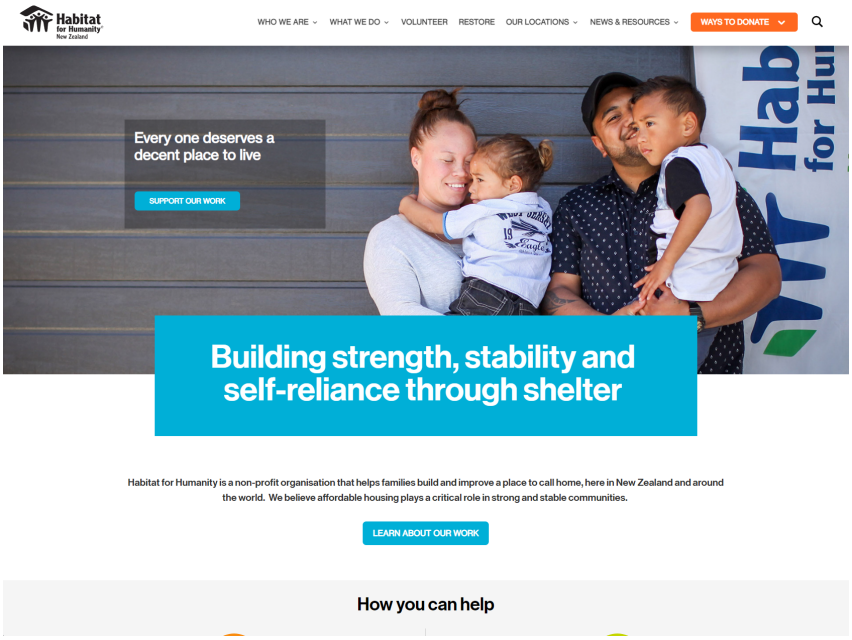
1. Sam goes to the Hands for Hope website and logs in to his account.
2. He clicks on the "Create Request" button and fills out the form, describing the problem with his fence and the help he needs.
3. Sam details what needs to be done on his fence, including photos.
4. Sam chooses a volunteer who has experience with fence repair and seems like a good fit for the job.
5. Sam and the volunteer arrange a time and date to meet at his home and work on the fence together.
6. The volunteer arrives at Sam's home at the scheduled time.
7. Together, they work on the fence and chat about their shared interests and experiences.
8. After they're finished, Sam thanks the volunteer for their help and feels grateful for the support he received from the Hands for Hope community.
9. Sam can then give recommendations on the volunteer that helped.

Competitive Analysis

I have chosen 3 websites to analyse in depth. These websites are chosen because they are well-established charity websites that have been around for ages and have proven that they are something that people should look into. I have chosen Habitat for Humanity New Zealand, Hands for Hope, and Mates. These are all local charities in New Zealand and are similar to mine.

In all these websites, I will be analyzing the navigation, layout, colours, content, functionality, and call to action of each website to produce some of the design elements that will influence the designs that I will create.

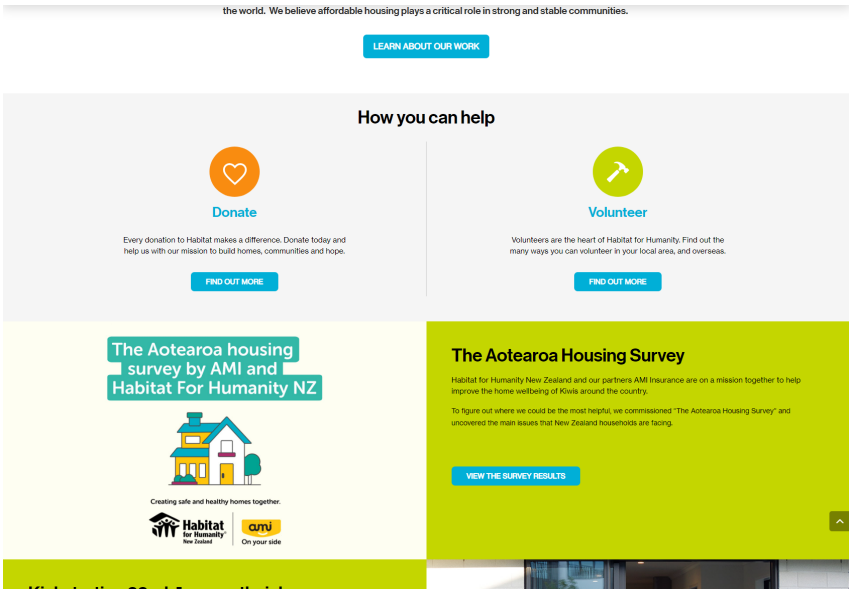
https://habitat.org.nz/



Navigation
The navigation menu on the Habitat for Humanity New Zealand website is located at the top of the page and includes links to various sections of the site. The menu is easy to read and understand, and it provides clear pathways for visitors to find the information they need. The navigation menu features a clear logo on the left-hand side with the necessary links on the right-hand side. This is your standard navigation menu with the text “ways to donate” highlighted in orange as it expresses that they want you to donate in some form.

Layout
The layout of the Habitat for Humanity New Zealand website is clean and simple, with a white background and minimal use of images and graphics. The website uses a grid-based layout, with information organized into clear sections that are easy to read and navigate. The text is well-spaced, making it easy to read, and the use of headings and subheadings helps to break up large blocks of text.

Colours
Habitat for Humanity New Zealand uses white as its background with orange, blue, and green as its accent colours.

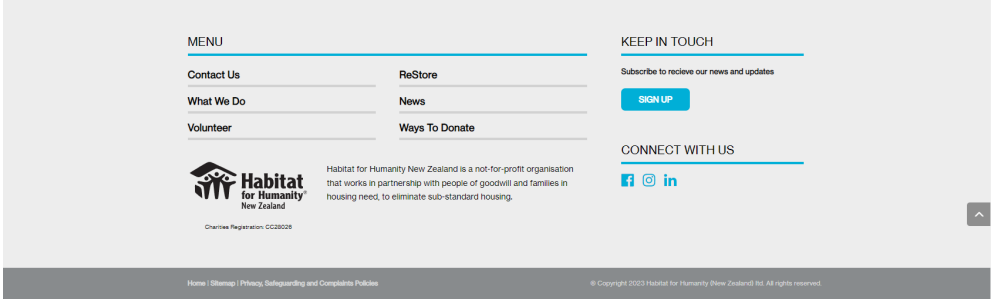


Content

The content on the Habitat for Humanity New Zealand website is well-written, informative, and easy to understand. The site provides a good overview of the organization's mission, values, and history, as well as information about its projects, volunteer opportunities, and ways to donate.

Functionality

The Habitat for Humanity New Zealand website is easy to navigate and user-friendly. The site is well-optimized for different screen sizes, and the pages load quickly. The website includes social media links and an email newsletter signup form, making it easy for visitors to stay connected with the organization. The site also includes a search function, which helps find specific information.



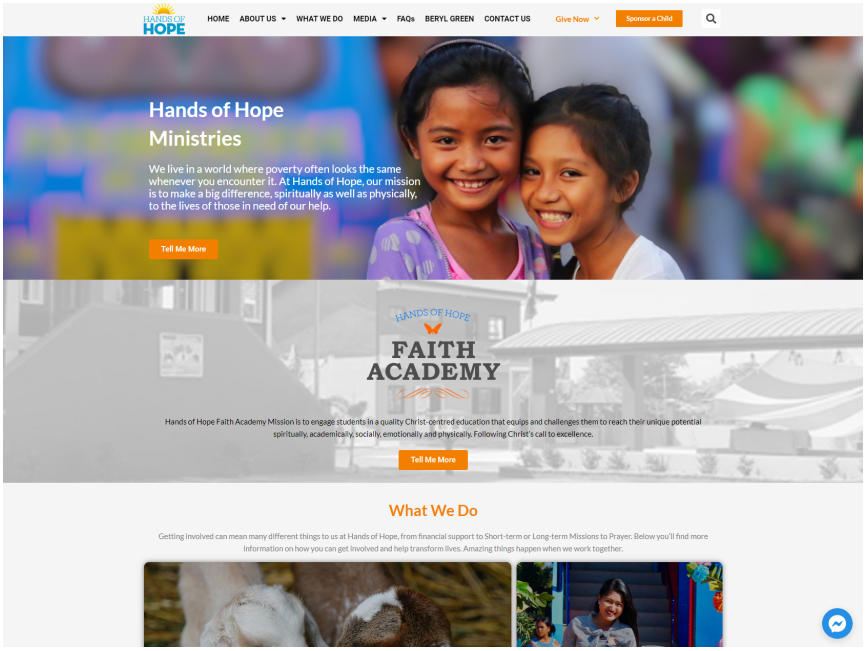
Call-to-action

The Habitat for Humanity New Zealand website includes several calls to action, such as "Ways to Donate" and “Donate”. These calls to action are prominently displayed on the homepage and throughout the site in orange, making it easy for visitors to take action and support the organization's mission.

Overall

The overall website design of Habitat for Humanity New Zealand is simple, clean, and effective. The site is easy to navigate, with a clear hierarchy of information, and the content is well-written and informative. The site is also well-optimized for different screen sizes, and the functionality is user-friendly. The calls to action are prominently displayed, making it easy for visitors to take action and support the organization's mission. Overall, the website is well-designed and effective in achieving its goals.

https://handsofhope.org.nz/



Navigation

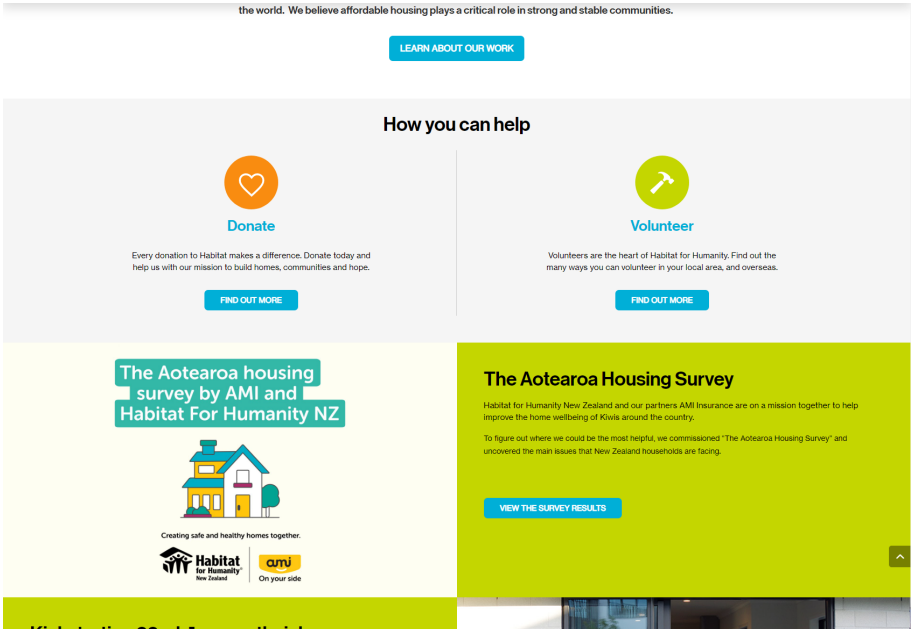
The navigation on this website is clear and intuitive, with a simple menu located at the top of the page. This navigation menu does feature a few too many links. Cluttering the menu up. Like most charities, they also feature “Give Now” and “Sponsor a Child” buttons along the menu. This is intended by the designer to draw the user's attention.

Layout

The layout of the site is straightforward, with a responsive design that adapts to different screen sizes. The homepage features a hero image that fills the screen, followed by a brief introduction to the organization and its mission. Followed by “What we do” and then their social media feeds. The social media feeds do make the website look unprofessional and messy.

Colours

There is mainly a white and orange colour scheme used on the site, which helps to convey a sense of warmth, optimism, and hope. The white background provides a clean and uncluttered feel to the site, allowing the content and images to stand out. The use of orange as an accent colour throughout the site adds vibrancy and energy, drawing the eye to important elements such as calls to action and headings. Additionally, blue is used sparingly on some links, the logo, and a few buttons at the bottom of the page, creating a sense of balance within the overall colour palette.

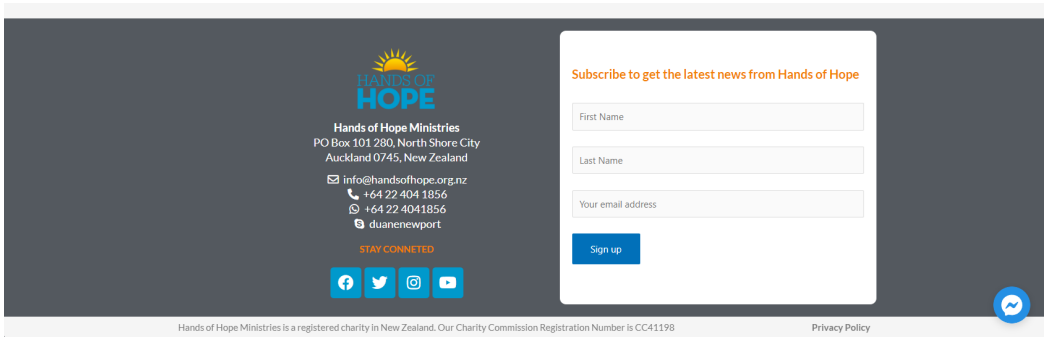


Content

The website's content is both informative and well-written, with a primary focus on the organization's mission and the positive impact it has had on the community. The language used is clear and accessible, ensuring that users can easily understand the organization's goals and the work it does. In addition, the website features social media feeds, including a Facebook feed that displays user comments. While this feature allows for user engagement and feedback, it also opens the door to all sorts of text and images.

Functionality

The site is well-designed from a functional perspective, with fast load times and clear calls to action throughout. The site also features social media links, allowing users to share content on different platforms.



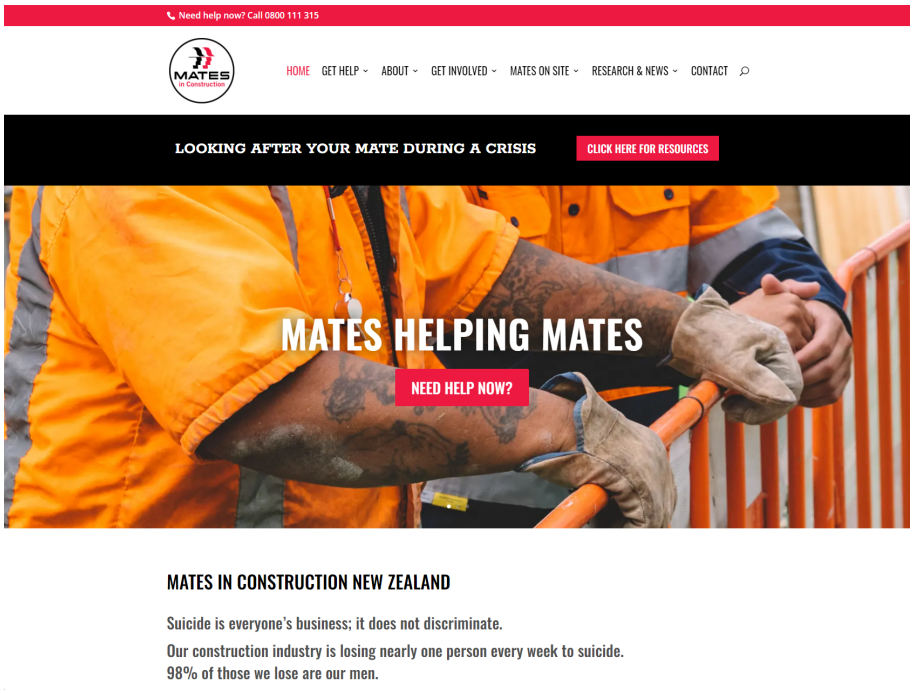
Call-to-action

The site features clear calls to action throughout, encouraging users to get involved with the organization in different ways. For example, there are links to donate, volunteer, and sign up for the organization's newsletter. These calls to action are prominent without being overly intrusive, making it easy for users to take action if they are interested in doing so.

Overall

Overall, the design of the site is okay. The site does a good job of highlighting the organization's mission and the impact that it has had in the community, while also providing clear calls to action for users who are interested in getting involved. This website could do with an overhaul, fixing up the messiness and some of the outdated designs. Overall, this is a website that effectively communicates the goals and activities but falls short when it comes to modern design.

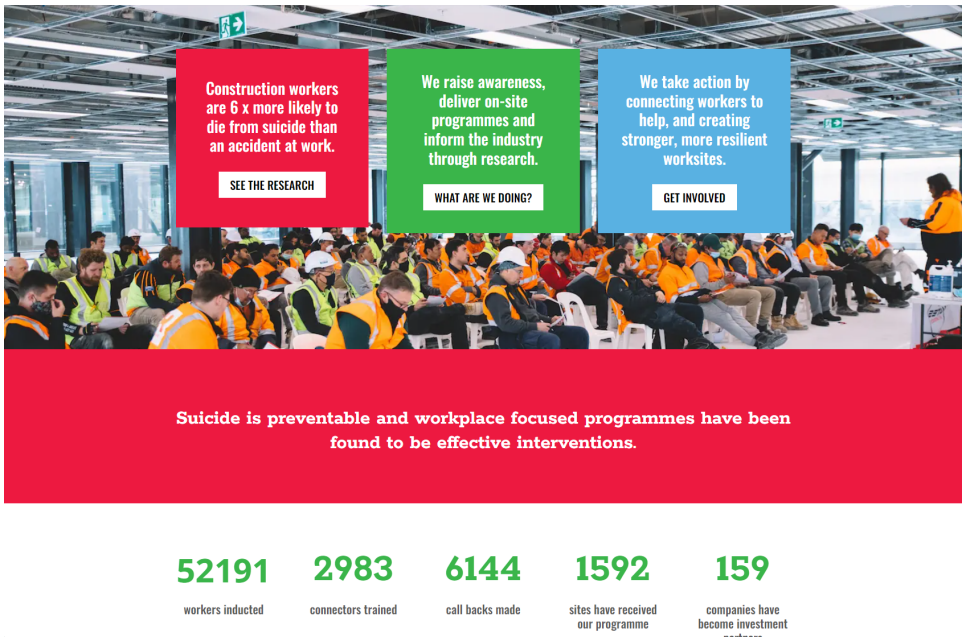
https://mates.net.nz/



Navigation
The website's navigation is straightforward and user-friendly, with the navigation menu located at the top of the page, as is typical of most modern websites. Notably, the site does not include a "Donate" button or similar features. Instead, a "Call Now" button is prominently displayed above the navigation menu, emphasizing the site's primary objective of encouraging users to seek immediate help.

Layout
The website layout appears modern and well-organized, with content divided into blocks for ease of navigation. The design may have drawn inspiration from construction websites, as the site's target audience is males within the construction industry. Each page follows a consistent layout that is easy to follow, regardless of the user's technical proficiency.

Colours
Mates' website utilises a colour scheme of white, red, and black, which remains consistent throughout the site. The use of red serves to emphasize the urgency of seeking help, conveying the message that Mates wants users to act quickly and not delay in getting assistance.

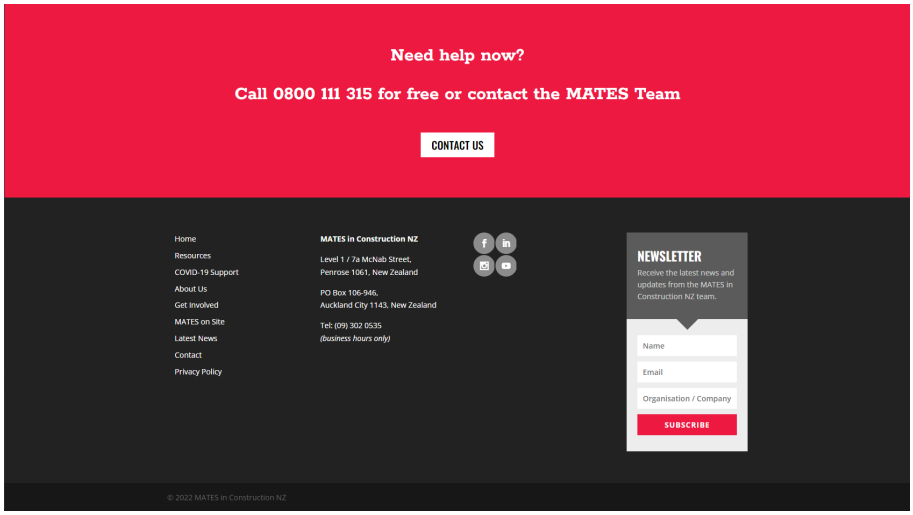


Content

The content on the website is well-written and informative, providing users with a clear understanding of the company's services and values. The language used is straightforward and easy to understand, with a focus on explaining complex technical concepts in simple terms.

Functionality

The website is easy to navigate and use, with clear calls to action throughout the site. The site is responsive, meaning it adapts to different screen sizes and devices, which is important for users who access the site on mobile devices. The site also includes a contact form that allows users to get in touch with the company directly.



Call-to-action

The website's primary objective is to encourage users to seek immediate assistance if they are struggling with mental health issues. This message is conveyed prominently throughout the site, with the first content on the page urging users to call for help. The site also includes numerous buttons and sections dedicated to connecting users with the help they need.

Overall

Overall, the website design is clean, professional, and user-friendly. The site effectively communicates the company's services and values, while providing users with a clear path to take action. The layout is very easy to follow and not too intrusive by having ample space between content, making it easy to navigate.

My Thoughts

Navigation
Keeping the website design simple and easy to navigate is an excellent approach, particularly if the target audience is not particularly computer-savvy. The design should be familiar and intuitive, allowing users to find what they're looking for quickly and easily. Including a logo can help establish the project's identity and build brand recognition, while a few prominent links can direct users to essential sections of the site. A clear and prominent donate button is crucial to encourage users to support the project financially, while a search bar can help users find the specific information they may be looking for quickly. Overall, the goal should be to make the website as user-friendly and accessible as possible, with a focus on providing users with a clear and concise path to take action and support the project.

Layout
Having a hero image as the first thing a user sees can be an effective way to capture their attention and make a strong first impression. It's essential to ensure that the hero image reflects the project's purpose and message and is visually appealing to the target audience. After the hero image, it's a good idea to include a clear and concise statement of the project's mission to help users understand its purpose. Providing some information about the charity can also help establish credibility and trust with potential donors or volunteers. It's crucial to balance between providing enough information to entice users and overwhelming them with too much information. Keeping the home page free from excessive information and displaying the right information in a visually appealing and easy-to-digest manner can make a significant difference in engaging users and encouraging them to take action.

Colours
Using a colour palette website to select complementary colours is an excellent idea to ensure that the colours used on the website contrast and complement each other. Pastel colours are often associated with calmness, serenity, and tranquillity, making them a good choice if you're aiming for a soothing, relaxing atmosphere. Harsh colours, on the other hand, can be overwhelming and detract from the user experience. It's crucial to select colours that enhance the website's overall design and message, rather than detract from it. Ultimately, the colour scheme should reflect the tone and purpose of the project, while also being visually appealing and easy on the eyes.

Content
Content is indeed crucial, and it's essential to ensure that the content is relevant and related to the project. Including irrelevant or unnecessary information only adds clutter to the website and detracts from its message. To reduce the amount of text on a page and make the content more digestible, it may be helpful to combine sections when possible. This can help streamline the user's experience and make the site more visually appealing. Ultimately, the goal should be to provide the necessary information clearly and concisely that is easy for users to understand and engage with.

Functionality
Load times are crucial, and it's essential to ensure that the webpage is optimized for quick loading times. While it may be tempting to fill the website with images, it's important to limit them to prevent slow load times, especially for users with slower internet connections. Moreover, optimizing the site for search engines is crucial to increase its visibility and reach a larger audience. This includes implementing best practices for search engine optimization (SEO) and making sure that the site is accessible on a wide range of browsers and devices.

Call to Action
Clear calls to action are essential throughout the webpage to encourage user engagement and increase interaction. These should be prominently displayed on the site without being too invasive, striking a balance between being eye-catching and not disrupting the user's browsing experience.

Site Content

Identify Content

Homepage:

- Hero Image
- A brief introduction to the charity and its mission
- Call-to-action buttons for visitors to log in or register
- Featured volunteer jobs/events hosted by Hands for Hope

About Us:

- A more detailed description of the charity and its history
- Vision and goals for the future

Volunteer Work:

- A search function for volunteers to find jobs
- Categories of jobs
- Job listings with descriptions, requirements, and contact information
- Information for new volunteers

Events:

- Calendar of upcoming events hosted by Hands for Hope
- Opportunities for volunteers to sign up for events or RSVP
- Information about past events and their impact

Donate:

- Ways to donate to the charity
- Information about how donations are used to support the charity's mission

Contact Us:

- Contact information for the charity
- Frequently asked questions and answers

Functional Requirements

User Registration and Login:

- Users should be able to create an account with their personal details and login credentials.
- Users should be able to log in to their accounts with their email and password.

Volunteer Job Search and Application:

- Volunteers should be able to browse and search for volunteer job opportunities.
- Volunteer job listings should include a detailed description of the job and its requirements.
- Volunteers should be able to apply for the job directly through the website.

Business/Nonprofit Job Posting:

- Businesses and nonprofit organizations should be able to create an account and post job opportunities that require volunteers.
- The job posting form should include information about the job, its requirements, and the organisation's contact information.
- Businesses and nonprofits should be able to edit or delete their job postings.

Events:

- The website should provide a calendar of upcoming events hosted by the charity.
- Volunteers should be able to sign up for events directly through the website.

Donation:

- The website should provide a secure online donation form for users to donate.
- The donation form should include options for one-time and recurring donations.

Contact Us:

- The website should provide a contact form for users to submit inquiries or feedback.
- The website should provide a frequently asked questions section with answers to common questions.
- The website should provide contact information for the charity, including email and phone.

Group and Label Content

Home

- Hero
- The charity's mission statement
- A section about the charity
- A calendar or cards to show upcoming events
- A call to action button that leads to the donation page
- A section that provides charity information to get in touch

Volunteer Jobs

- Job Posting System
- Job Search
- Accepting Jobs

Events

- Calendar showing upcoming & past events

Donate

- A section about the importance of donations to support the charity's work
- Donation form

Contact

- Full charity information
- Contact form
- FAQ

Login

- Login form
- Registration form

Site Structure

Metaphor Exploration

Organizational Metaphors

Imagine "Hands for Hope" as a community garden, where businesses are like plants that need nurturing. Volunteers collaborate with businesses to cultivate a sustainable environment. To reflect this on the website, I would incorporate earthy colour tones and highlight the significance of community and progress.

Functional Metaphors

Think of "Hands for Hope" as a platform similar to TradeMe or Craigslist, but instead of purchasing items, you can apply to assist local businesses or individuals in need. The focus of “Hand for Hope” is not about the money but about helping others within your community. "Hands for Hope" is all about building a stronger, more connected community.

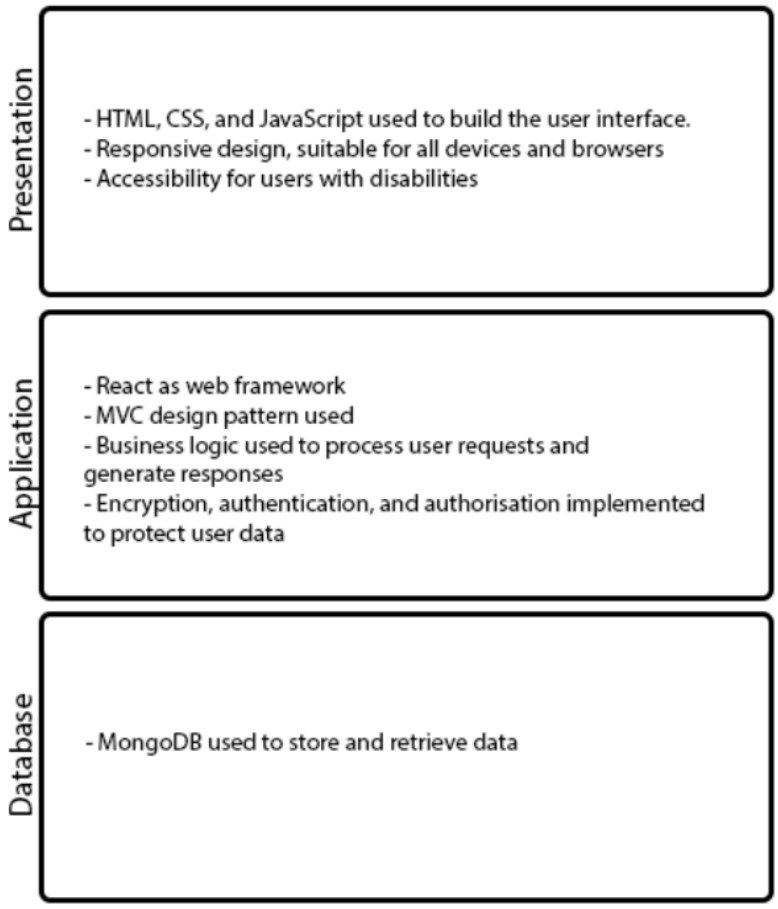
Visual Metaphors

Helping others can come in many forms, but one that often comes to mind is physically helping someone build a fence or tend to their garden. A real hands-on task. I believe that using plant imagery and an earthy colour palette on the website would symbolize the idea that creating a thriving ecosystem requires the collaboration of many individuals.

Site Structures Listing

- Home
    - Volunteer Jobs
        - Create Job
        - View existing Jobs
    - Events
        - Upcoming
        - Past
    - Contact
        - Contact form
        - Charity information
        - FAQ section
    - Donate
        - Donate form
    - Login
        - Registration form
        - Login form
    - Account
        - My information
        - Volunteer History

Architectural Blueprint



Define Navigation

Global Navigation

Global navigation refers to the menu that is visible across all the pages. This allows users to navigate back and forward throughout the website. This menu is usually located at the top of the page but it is depended on the website's specific requirements and needs. Other locations of the navigation could be on the side or on the bottom.

Local Navigation

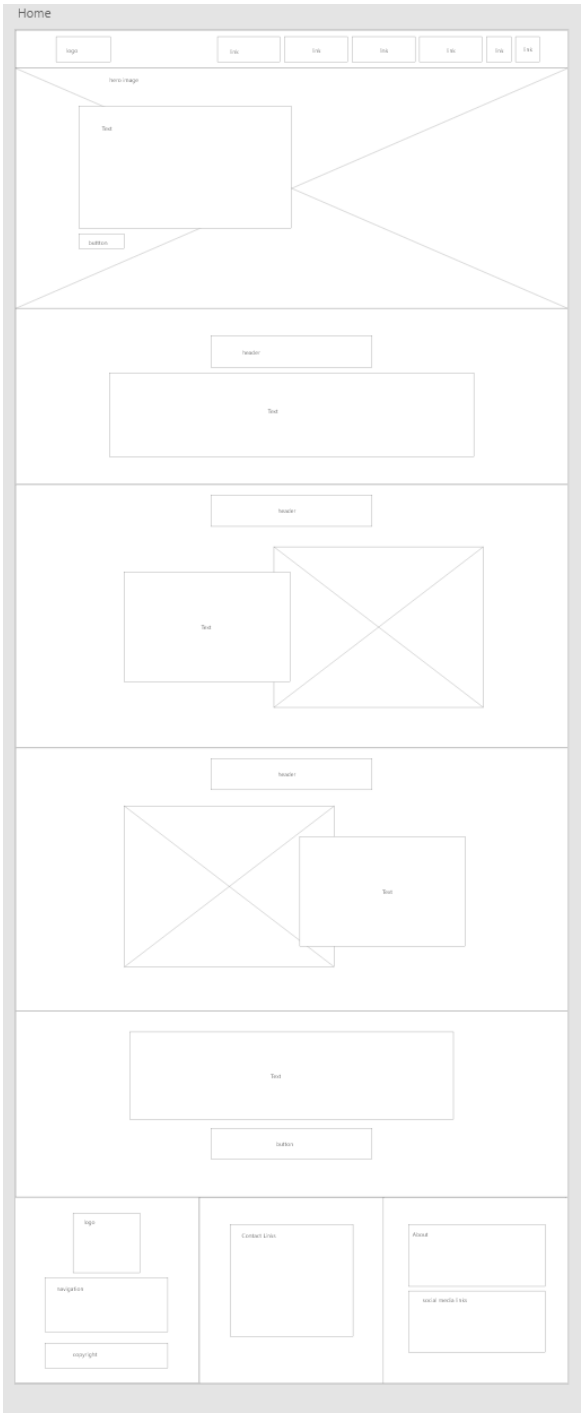
Local navigation refers to a set of links specific to that region or section of a website. Such as sub categories, breadcrumbs or related articles. Local navigation helps to send users down the rabbit hole of a particular sections of a website.

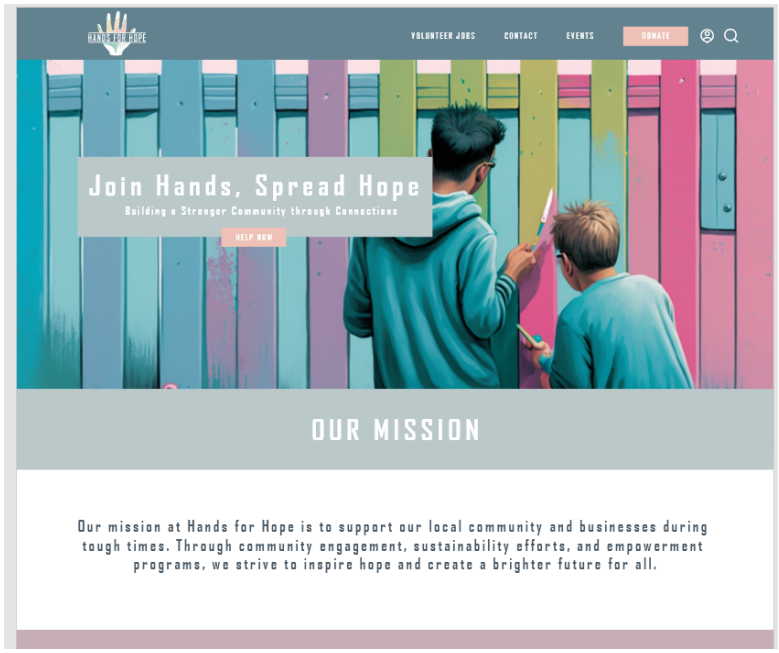
Visual Design

#62828F	#C7AEB4
#BAC8C9	#62828F
#4A5A67	

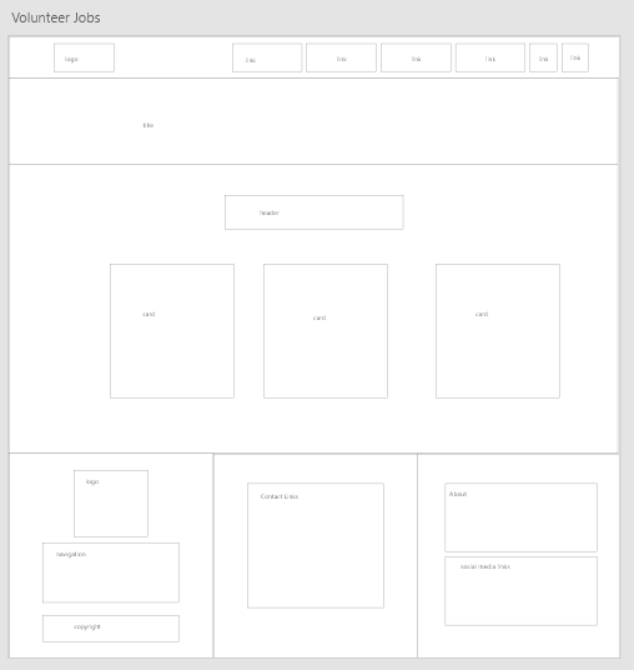
Design Sketches and Page Mockups

Home

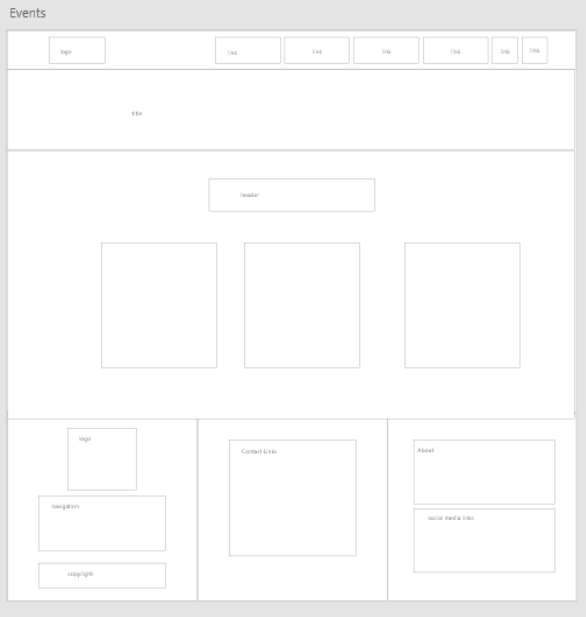




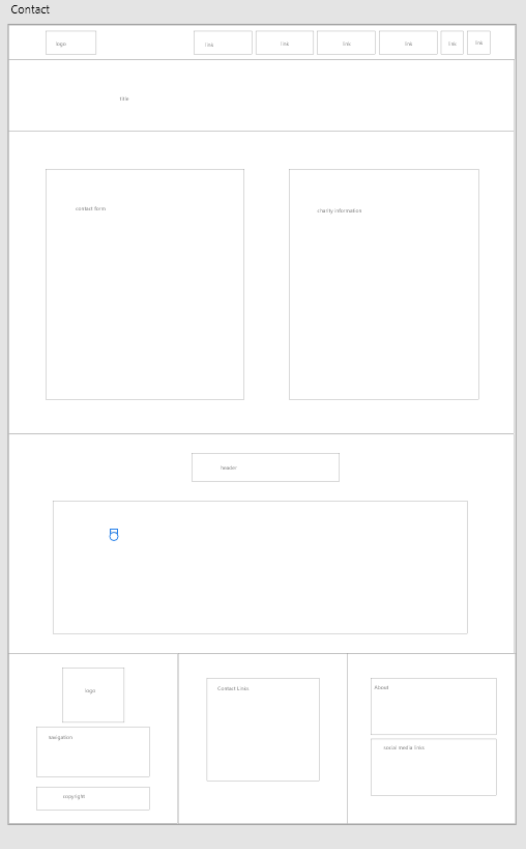
Volunteer Jobs


Volunteer Jobs
logo
link
link
link
link
link
link
title
header
card
card
card
logo
navigation
copyright
Contact Links
About
social media links

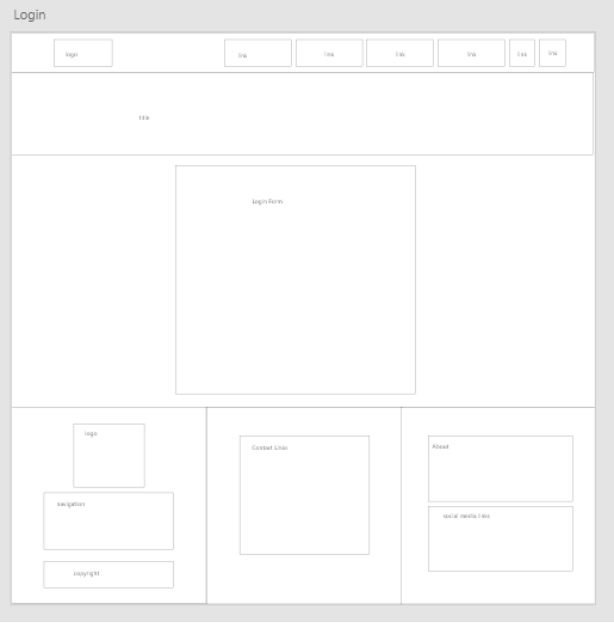
Events


Events
logo
link
link
link
link
link
link
title
header
logo
navigation
copyright
Contact Links
About
social media links

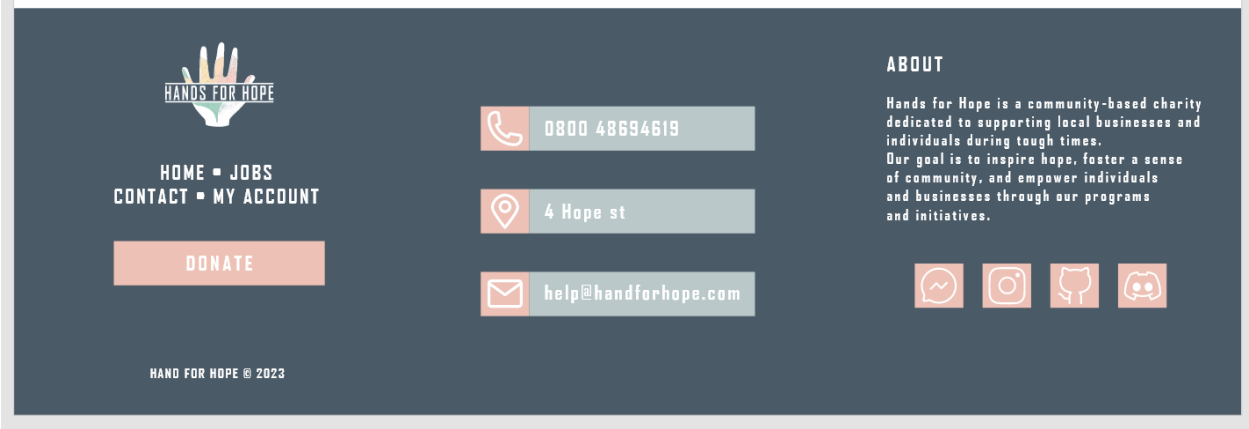
Contact



Login



Footer



Navigation

Purpose of Web Frameworks

Web frameworks provides developers with libraries and tools that are standardised within that framework. Web frameworks encourage the best practices for building web pages and applications. This process has simplified web development by abstracting tasks and jobs by putting them into reusable components. Web frameworks handle the lower-level code so that developers can focus more on the application features.

Most Web frameworks share similar features. The common features include routing, middleware, templating, database integration and other security features.

In web development, routing, also known as a router, is a mechanism that directs HTTP requests to the appropriate code (Routing, 2019). This means that the developer determines what happens when a user visits certain pages. As an example, a user visits “/users” and the system returns a list of all the users. Routing is helpful to create a clean and organised application structure.

"Middleware is software that different applications use to communicate with each other. It provides functionality to connect applications intelligently and efficiently so that you can innovate faster.” (What Is Middleware? - Middleware Software Explained - AWS, n.d.). Middleware can be used for tasks such as authentication, input validation and logging. It also provides a way to execute code before and after a request has been handled.

Templating is the production of web pages with dynamically based data from the database. This is used in conjunction with creating mass web pages. Such as if a user logs in and the web page wants to display the user's name then the page would fetch the user's name and display it. Instead of creating a million different pages covering everyone's name, you only create one page and have it fetch data from the database.

Web frameworks make it easy to integrate databases such as MySQL and MongoDB. This allows developers to store and retrieve data easily from their database.

Each framework may include security features to protect against harmful attacks such as XSS and SQL injection. Frameworks can also utilise libraries to help with web application vulnerabilities. Cookie-session, helmet and mongoose just to name a few.

For the Hands For Hope website, I will be using the MERN stack which includes MongoDB, Express, React and Node. This stack will help streamline the development process for the website. This website will be using MongoDB for the database, React for the front end and Express for the back end. Jsonwebtoken, mongoose, bcrypt and helmet will be some of the dependencies that I will use with my web application.

Emerging Web Technologies

Some of the web technologies used for the Hands For Hope project will be React, Express, Node, MongoDB and Redux. React is a popular web framework used to build interactive UI and applications quickly and efficiently. Express is a web framework for Node. Express makes it easy for developers to build a back end in Node. MongoDB is a database system that stores data in JSON-like documents meaning that documents and data structures can vary and be changed over time. (MongoDB, 2023). Redux is a Javascript library that helps manage the state of the application. Redux creates a centralised store that makes it easy to track and access the state from anywhere in the application.

As web applications and frameworks continue to grow in size and complexity, managing one's state has become increasingly more challenging. Redux provides a solution by managing states in a centralised store. This makes it easier for developers to write clean and maintainable code. One user has found a tradeoff stating “One might feel it adds up boilerplate code, making simple things a little overwhelming; but that depends upon the architecture decisions.” Ighodaro (2022). Redux was one of the early adopters of the state management system and has been stated by multiple sources that it will not be going anywhere soon. Redux is one of the best options for large projects and for actual production applications. There are other state management systems such as Recoil which have an easier user and developer experience but it is still in its early days.

I chose to use Redux because it is a well establish state management system. I could have gone with some of the lesser-known ones but they are still in early development and are currently going through changes as they develop. Using Toolkit with Redux reduces that “boilerplate” code that some others have noted and makes it easier to write applications. As of May 30th 2023, Redux has 6.5 million NPM weekly downloads. That is more than any other combined. The next highest library is Zustand with 1.6 million NPM weekly downloads (fe-tool, 2023).

To implement and install Redux, first, you would use “*npm install react-redux*” to install redux into React. Followed with “*npm install @reduxjs/toolkit*” to install Redux Toolkit. Then within my client folder, I would create a state folder with a js file name index. This is where I will state my states. In this file, I would store the login and logout state of a user. These can be called from other files if needed.

Since Redux, other developers have expanded and improved on Redux. Libraries such as Zustand, Xstate and Mobx. Zustand is built on top of React’s Context API and hooks to manage states so you do not need to wrap your application in context providers. Recoil is another one to keep an eye on. Fe-tools (2023) talks about Recoil as it “ allows developers to manage and share states across different components without relying on complex data flow patterns. It uses a simple and intuitive API to define and access state, making it easier to understand and maintain complex codebases.”

Redux and other state management systems have had big importance on how web development within the last few years. Ighodaro (2022) states that “Without Redux, you would have to make data dependent on the components and pass it through different components to where it’s needed.”

Comparing Nextjs & React

I am comparing Nextjs to React. React is a javascript library used to build user interfaces with components. Whereas Nextjs is a framework built on React.

Register, log in and administer their account.

Members can register, log in and edit their accounts With React, I have utilised redux and its createSlice function. “A function that accepts an initial state, an object of reducer functions, and a "slice name", and automatically generates action creators and action types that correspond to the reducers and state.” (*createSlice | Redux Toolkit*, 2022). I also used yup for some data validation.

React

This allows to set the state if that user is logged in or not. If the user is logged in, then they have access to parts that they were previously allowed.

```
export const authSlice = createSlice({
  name: "auth",
  initialState,
  reducers: {
    // Set the user and token information when the user logs in
    setLogin: (state, action) => {
      state.user = action.payload.user;
      state.token = action.payload.token;
    },
    // Clear the user and token information when the user logs out
    setLogout: (state) => {
      state.user = null;
      state.token = null;
    },
  },
});
```

This is the schema for registration. Using yup for data validation.

```
const registerSchema = yup.object().shape({
  firstName: yup.string().required("required"),
  lastName: yup.string().required("required"),
  email: yup.string().email("invalid email").required("required"),
  password: yup.string().required("required"),
  location: yup.string().required("required"),
  occupation: yup.string().required("required"),
  picture: yup.string().required("required"),
});
```

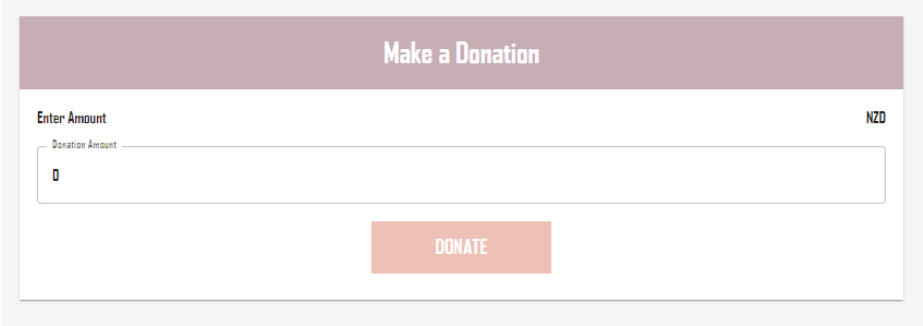
Nextjs

Because Nextjs is built on react, they have a few similarities. Instead of using the same login system, I have implemented Google Authenticator. This redirects the user to a Google login. This uses Google Cloud Apis so the user does not need to sign up. Eliminating a login page.

```
import NextAuth from "next-auth";
import GoogleProvider from "next-auth/providers/google";
import { connectToDB } from '@utils/database';
import User from '@models/user';

const handler = NextAuth({
    providers: [
        GoogleProvider({
            clientId: 'process.env.GOOGLE_ID',
            clientSecret: 'process.env.GOOGLE_CLIENT_SECRET',
        }),
    ],
    callbacks: {
    async session({ session }) {
        const sessionUser = await User.findOne({ email: session.user.email });
        session.user.id = sessionUser._id.toString();

        return session;
    },
    async signIn ({ profile }) {
        try {
            await connectToDB();

            const userExists = await User.findOne({ email: profile.email });
            if (!userExists) {
                await User.create({
                  email: profile.email,
                  username: profile.name.replace(" ", "").toLowerCase(),
                  image: profile.picture,
                });
            }
            return true;

        } catch (error) {
            console.log(error);
            return false;
        }
    },
    }
})
```

Acquire tokens.

With my project, the way businesses and members can acquire tokens is through donating. These donations help fund other projects around the community. Donating such as $100 would grant the user 1 token. These donations are stored separately and the database will keep a total. Below is the interface where users can donate. Ideally, the user would be redirected to a safer donation page. This could be PayPal, stripe, or anything as such. Businesses and clients could use those tokens to have their prompt pushed to the main page.



Authentication using JWT to transfer state.

React

This middleware function is used to verify that a valid JSON web token is provided with the request. It is used to protect certain routes. That secret code is stored within the .env file.

```
export const verifyToken = async (req, res, next) => {
  try {
    let token = req.header("Authorization");

    if (!token) {
      return res.status(403).send("Access Denied");
    }

    if (token.startsWith("Bearer ")) {
      token = token.slice(7, token.length).trimLeft();
    }

    const verified = jwt.verify(token, process.env.JWT_SECRET);
    req.user = verified;
    next();
  } catch (err) {
    res.status(500).json({ error: err.message });
  }
};
```

Nextjs

Nextjs does JWT in the same fashion as react. We can explore NextAuth instead of using the same for React. This function handles the login. If the user already exists in the database, then the system will log in them in. If the user does not exist then the system will create the data for them in the database.

```
const handler = NextAuth({
    providers: [
        GoogleProvider({
            clientId: 'process.env.GOOGLE_ID',
            clientSecret: 'process.env.GOOGLE_CLIENT_SECRET',
        }),
    ],
    callbacks: {
        // Call this function when the session is accessed, find the user by email and add the user id
to the session object
    async session({ session }) {
        const sessionUser = await User.findOne({ email: session.user.email });
        session.user.id = sessionUser._id.toString();

        return session;
```

```
    },
    // Call this function when the user signs in, create a new user if one doesn't exist already
    async signIn ({ profile }) {
        try {
            await connectToDB();

            const userExists = await User.findOne({ email: profile.email });
            if (!userExists) {
                await User.create({
                  email: profile.email,
                  username: profile.name.replace(" ", "").toLowerCase(),
                  image: profile.picture,
                });
            }
            return true;

        } catch (error) {
            console.log(error);
            return false;
        }
    },
    }
})
```

System Stores and retrieves user data.

For both React and Nextjs, I used MongoDB for my database. This can be used by using `import mongoose from "mongoose";` within your main server folder. Since they are both using Mongoose, both are very similar. Within your backend, you would create a schema that MongoDB can read.

React

This is part of the user schema. This is the information that is stored within the backend.

```
const UserSchema = new mongoose.Schema(
  {
    firstName: {
      type: String,
      required: true,
      min: 2,
      max: 50,
    },
    lastName: {
      type: String,
      required: true,
      min: 2,
      max: 50,
    },
});
```

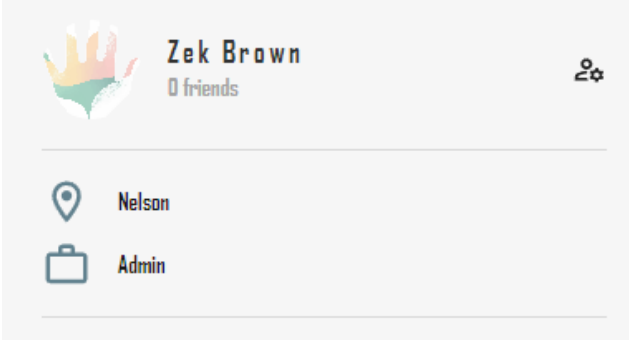
Below is what is actually stored on MongoDB.

```
  _id: ObjectId('64367139d1ae50f1eb8acbe9')
  firstName: "Zek"
  lastName: "Brown"
  email: "zek.bro@hotmail.com"
  password: "$2b$10$6bQURjpVqO8KmhemP3kOJ.62UuKnRSOH9m1xyp808oSG3cTBhPiZS"
  picturePath: "Logo2.png"
▸ friends: Array
  location: "Nelson"
  occupation: "Admin"
  viewedProfile: 8729
  impressions: 1878
  createdAt: 2023-04-12T08:52:09.489+00:00
  updatedAt: 2023-04-12T08:52:09.489+00:00
  __v: 0
```

This can be called using a function. This piece of code is on the user's profile page to fetch data about the user. It uses the userID to figure out whose profile it belongs to.

```
  const getUser = async () => {
    const response = await fetch(`http://localhost:3001/users/${userId}`, {
      method: "GET",
```

```
    headers: { Authorization: `Bearer ${token}` },
  });
  const data = await response.json();
  setUser(data);
};
```



Nextjs

The Schemas are the same as React as it uses MongoDB.

```
const UserSchema = new Schema({
  email: {
    type: String,
    unique: [true, 'Email already exists!'],
    required: [true, 'Email is required!'],
  },
  username: {
    type: String,
    required: [true, 'Username is required!'],
    match: [/^(?=.{8,20}$)(?![_.])(?!.*[_.]{2})[a-zA-Z0-9._]+(?<![_.])$/, "Username invalid, it should
contain 8-20 alphanumeric letters and be unique!"]
  },
}),
```

Below is the data that mongoDB receives when a user is logging in for the first time. It takes the username, email and image from Google's database.

```
_id: ObjectId('646737154d988dbff2aaaa93')
email: "zek.[redacted]"
username: "zek"
image: "https://lh3.googleusercontent.com/a/AGNmyxbVsb99lXNLDtMpDbk0b[redacted]"
__v: 0
```

And this is the output rendered on the website. This includes name, email and image. Any of this information could be censored but I am showing the email for testing purposes.

Z

zek

zek[redacted]

Hey, I am in need of help putting up a fence

#Building

Routes

React does not have an inbuilt solution when it comes to routing. It has to make use of the react-router-dom dependency. Nextjs has an in-built routing system meaning developers will not have to set up routing the same as react. With Nextjs, you would just create a folder within the app folder and then create a page.jsx. That is your route. Nextjs has simplified routing.

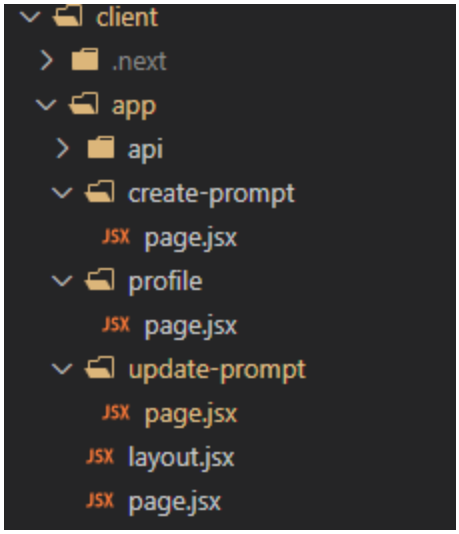
React

This is some of the code in my app.js. Specifically around the routing. You can see that you use the route tag to direct the user. If the path is localhost:3000/donate then the user will be taken to the donation page.

```
import { BrowserRouter, Navigate, Routes, Route } from "react-router-dom";

  return (
    <div className="app">
      <BrowserRouter>
        <ThemeProvider theme={theme}>
          <CssBaseline />
          <Routes>
            <Route path="/" element={<HomePage />} />
            <Route path="/login" element={<LoginPage />} />
            <Route
              path="/profile/:userId"
              element={isAuth ? <ProfilePage /> : <Navigate to="/" />}
            />
            <Route path="/contact" element={<ContactPage />} />
            <Route path="/donate" element={<DonatePage />} />
            <Route path="/events" element={<EventsPage />} />
            <Route path="/volunteer" element={<VolunteerPage />} />
          </Routes>
        </ThemeProvider>
      </BrowserRouter>
    </div>
  );
}
```

Nextjs

With nextjs, for your routing you would create a new folder within your app file structure. Then add the file page.jsx. Below you can see routing for /create-prompt, /profile, /update-prompt as an example.


client
.next
app
api
create-prompt
page.jsx
profile
page.jsx
update-prompt
page.jsx
layout.jsx
page.jsx

I have made significant progress with both of these frameworks. React is bare bones and requires a few more dependencies than something built on top of it such as Nextjs. With React I have 19 dependencies installed to handle the routing, state management, icons, and data validation just on the front end. Comparing that to Nextjs, I have 11 with Nextjs. Nextjs makes the setup a lot easier. With Nextjs, you do not need to set up your server. It comes with server-side rendering right out of the box. Nextjs provides simple routing through the file structure.

In conclusion, React comes with a lot out of the box but you have to know and find other dependencies to truly take advantage. Nextjs have done some of that work for you already. As I continue to develop my website I may look into switching over to Nextjs. It seems easier to use and more developer friendly. If I was not so far through with React then I would have used Nextjs.

The GitHub Repositories can be found here.
Nextjs
React

References

createSlice | Redux Toolkit. (2022, August 17). https://redux-toolkit.js.org/api/createSlice#:~:text=A%20function%20that%20accepts%20an,approach%20for%20writing%20Redux%20logic.

fe-tool. (2023, May 30). *24 Best React State Management Libraries in 2023*. Fe-tool. https://fe-tool.com/awesome-react-state-management

Ighodaro, N. (2022, October 21). *Understanding Redux: A tutorial with examples - LogRocket Blog*. LogRocket Blog. https://blog.logrocket.com/understanding-redux-tutorial-examples/#:~:text=Redux%20allows%20you%20to%20manage,come%20with%20tradeoffs%20and%20constraints.

MongoDB. (2023). *What Is MongoDB?* https://www.mongodb.com/what-is-mongodb

Routing. (2019, May 3). https://divpusher.com/glossary/routing/#:~:text=Routing%20or%20router%20in%20web,simple%20routing%20example%20from%20Laravel.

What is Middleware? - Middleware Software Explained - AWS. (n.d.). Amazon Web Services, Inc.

https://aws.amazon.com/what-is/middleware/#:~:text=Middleware%20is%20software%20that%20different,that%20you%20can%20innovate%20faster.