Prapti Mane

Assignment - 3

Table of Contents

Executive Summary 4
Introduction 5
User Requirements 6
General Functionality 6
Section – Specific Functionality 6
Site Goals 7
Mission or Purpose of the Organisation 7
Short Term Goals 7
Long Term Goals 7
Intended Audience 8
Attraction to the Website 8
Backlog list & Features 9
Section Two – User Experience 9
Audience Definition 9
Scenarios 11
Competitive Analysis Summary 12
Competitor No – 1 (Breast Cancer Foundation NZ) 12
Competitor 2 (Breast Cancer Cure) 17
https://www.breastcancercure.org.nz/ 17
Competitor 3 (BCAC – Breast Cancer Aotearoa Coalition) 20
Section Three Site Content 22
Content Grouping and Labelling 22
Functional Requirements 23
Metaphors 23
Metaphor Exploration – Functional Metaphor 23
Metaphor Exploration – Visual Metaphor 23
Section 4 Site Structure 24
Site Structure Listing 24
Sitemap 25
Section 5 – Visual Design 26
Layout Grids 26
Home Page Layout 27
Breast Cancer Page Layout 29
Breast Awareness Page Layout 31
About - Us Page Layout 33

Book an Appointment Page Layout 35
Sign-Up Page Layout 37
Section 5 CRUD Analysis 39
CRUD Table 39
API Prototype 39
Purpose of web frameworks 42
Common Framework Features 43
Features used in Website 45
Proposed Web Frameworks 46
Why Angular: 46
Why VueJs: 46
Register, Login, Administer Accounts 46
Angular 46
VueJs 48
Code Snippets 54
Comparing Frameworks – Registration and Login 55
Members Use the Website to register their products. 56
Angular 56
VueJs 58
Code Snippet 58
Comparison 60
Interactive Engaging Elements 61
Angular 61
VueJS 62
Comparison 64
Store and Retrieve Data from Server-Side Database 65
Angular 65
VueJS 68
Comparisons 72
Recommendation of a Framework 72
Implementation of the Web Technology 76
Introduction to Angular 76
Implementation of Angular 76
Problems faced during Development. 78
Impact that you foresee this technology having on internet users. 78
Links to Journal 78

Bibliography 79

Executive Summary

This report can be used as a client document which will show how the Breast Cancer Charity website will be created to showcase the client's idea of promoting breast cancer awareness amongst the youth. This document presents the recommended steps that are documented for building an effective functional website as per the industry standard. The goal of this document is to show you a prototype of the website for your intended audience.

We are recommending on building this website using Express for the API and Angular or Vue for the front end. Since the client requirements are customized, we recommend building this website from scratch instead of using any CMS to build it. This website will be built by us as developers and it will be easier if we maintain it too as any customization can only be done in the code, which we are unsure if the client is comfortable changing. But that decision is for the client to make if they would like to maintain the website on their own or if they would like us to maintain it for them.

Below is the link to the GitHub Link to the website: -

https://github.com/praptimane26/BreastCancerCharity

So far, we've built the API for the Task Manager of the website, which will allow a user to Create, Update, Retrieve and Delete a list. The next phase would be to build the API for the Login System and the Booking System, and accordingly build the Front-end using Angular or Vue.

Introduction

After receiving the brief from client and what their expectation is towards the website, we're providing this report as a document to understand how important it is to have a functional website, using Angular for the front-end of the website which the users see on the website, for the backend we will be using Express and Node.js and to maintain the users and their database we will be using Mongo, the reason we cannot use a CMS for this website is because the client wants a customised website with very specific needs, hence we thought building this website from scratch would be the way to go.

Below is the brief that we received from the client:

The Breast Cancer Charity is about promoting breast cancer awareness and encouraging women and men to get regular check-ups and not avoid their health. This would be a charity website, helping communities spread awareness and provide free check-ups to anyone who registers as a user on the website. The website also provides a task management system just to keep the user on track with their check-ups, they can create tasks about their routine check-ups and can also be as a To-Do list. There will also be a booking system to redeem the token for free breast cancer check-up that will be emailed to the users when they register on the website.

In this report we have focused on the specific areas of the website, we have identified the potential visitors for the website, and how they would use the website, the functionality of the website and the goals too.

User Requirements

General Functionality

We think a functional website for you would contain images which would be related to breast cancer as images is what usually catches the user's attention, we will also have a video and text that will help promote the website. We will also have a task manager on the website which will help the user to keep a list of their health for example they can make a note of when their next home breast cancer check-up should be done etc. The website also provides one free breast cancer check-up to all the users that register on the website. A user can create a task edit it and even delete it depending on their requirement. Having **Home**, **About Us**, **Breast Awareness**, **Breast Cancer**, **Support** as the primary navigation will help the users direct to the areas, they are interested in. Also having a **Login** button on the top right of the website will help users to login or register on the website. And a **booking system** where the users can redeem their coupon to book the free breast cancer check-up.

Section – Specific Functionality

The specific sites on the pages could be :-

1) **About Us** – This page will contain all the information about the Breast Cancer Charity. What it is about and what their goals are. It will also display information of the founders of the website and their reason behind starting the Breast Cancer Charity.
2) **Breast Awareness** – This page will contain information about how to take care of your breasts. A video of how to check your breasts properly. This page will also have information about mammograms, risk factors and inherited risks.
3) **Breast Cancer** - This page will hold information about types of breast cancer, tests and diagnosis and the treatments available.

4) **Book an appointment** - This page is where the users will redeem their token to book a free Breast cancer check-up on the Breast Cancer Charity website.

Site Goals

Mission or Purpose of the Organisation

The purpose of the website is to spread breast cancer awareness amongst the masses. It will be used to educate people on how to take care of their health before it is too late. Since ACC does not cover most breast cancer drugs and treatments which are very expensive as Chemo alone costs thousands of dollars, it is better to take precaution, this is the message that the website wants to put out. The website will focus on creating awareness, support, treatment options, task manager which will be helpful for the user, as detected at an early stage it is easy to cure. Breast Cancer charity wants people over 20 years to focus and understand their bodies and notice any changes as quickly as possible.

Short Term Goals

The short-term goals of the website is to produce a mock-up of the Breast Cancer Charity website. This website will be functional, and it will help gather attention to the charity.

Long Term Goals

The long-term goals of the website are to have a functional and fully completed website which will be used to register new users and promote the charity as much as possible by providing free check-ups to the users who register on the website. It will also be used as a task manager by the users to keep track of their health and check-ups.

The registration on the website will be helpful to promote the Charity and its benefits of spreading breast cancer awareness, as well as be beneficial for the users to get free check-ups, and when they users register, they get a token emailed to them which they can redeem on the booking system of the website.

Intended Audience

The people that we think will be visiting the website in no particular order are: -

1) Breast Cancer Patients.
2) Doctors.
3) Charity members and anyone operating within the Charity.
4) Admin Access members
5) Women from all ages.
6) Men from all ages.
7) Other Supporting Charities.
8) Support Clinics.
9) Breast Cancer Patients Families.
10) The media/press to promote Breast Cancer Awareness.
11) Local Government to promote Breast Cancer Awareness.
12) Central Government to promote Breast Cancer Awareness.

Attraction to the Website

Some ways in which the new website can be promoted is by: -

1) Creating a Google campaign to promote the website, hence we need to make sure the content on the landing pages is in the format that will help promote the website and use the right keywords. Google business is the best tool to do this, since this is not an e-commerce website and a charity website, doing online campaigns will be more beneficial and price efficient.

2) Creating a Facebook Page and running a Facebook can be helpful as well, to promote the website and gain more users.
3) The charity members can also create an event where they will give free breast cancer check-ups to people who show up for the event, all they need to do is at the venue register on the Breast Cancer Charity website and get their free check-up on the spot, this event can be promoted using Facebook.

Backlog list & Features

Task ID	Story	Estimated Time	Priority
1	Set up an angular application, both front end and backend and set up a database connect it to the MongoDb installing Mongoose	2hrs	1
3	Login and Registration based on the Roles	10hrs	2
4	Introduce a booking system for the users to fill up the contact form	5hrs	3
5	Make a task manager for the user to create edit and delete lists	5hrs	4
6	Add creating, editing, and deleting tasks to the app	5hrs	5
2	Create the UI of the application	3hrs	6
7	Create the Admin dashboard	2hrs	7

Section Two – User Experience

Audience Definition

Breast Cancer Charity want the users to be able to register on the website and receive a token to book a free appointment for the breast cancer check-up and also use the free task manager which is available to use on the website.

The Breast Cancer Charity website will have many different types of users which were listed previously in the Intended Audience heading.

1) Breast Cancer Patients – Patients who suffer from Breast Cancer and are looking for support and more information about Breast Cancer.
2) Doctors – Doctors all over New Zealand can recommend the charity to their patients who are in the age between 20yrs old to 69yrs old. They can use the website in front of the patients to show its benefits.
3) Charity members and anyone operating within the Charity – The charity members will use the website to talk to sponsors as well as if they have the admin access, they can use it to manage the content and to check how many users the website has and how is it gaining popularity etc.
4) Admin Access members – Admin access members will use this website to maintain content on the website and post events etc.
5) Women from all ages – These are the users that we want to register on our website, as it will help create awareness, these women will come to the website to register and book their free check-up.
6) Men from all ages – These can register on the website to book a free appointment for themselves or their partners, wives, mothers etc.
7) Other Supporting Charities – Hospital charities like St. Johns etc can also promote the Breast Cancer Charity to their patients, to spread awareness. Show them the website to help them realize that they can get free check-ups and use the task manager to create tasks depending on their health etc.
8) Support Clinics – Support clinics can show their patients how beneficial the Breast Cancer Charity website and its workings and how they can claim a free check-up.
9) Breast Cancer Patients Families – The families of patients who are suffering from Breast Cancer will visit to website, to get themselves checked, as well as to gain more information about Breast Cancer as it will be helpful for them.
10) The media/press to promote Breast Cancer Awareness – The media/press will visit the website to understand what it is and how it works and promote the website to get locals and people are New Zealand to get free check-ups done.
11) Local Government to promote Breast Cancer Awareness – Local government will visit the website, to probably connect with the charity members to come

up with certain schemes or to help promote the website to spread Breast Cancer awareness especially around breast cancer month which is from 1st October – 31st October.

12) Central Government to promote Breast Cancer Awareness - Central government will visit the website, to probably connect with the charity members to come up with certain schemes or to help promote the website to spread Breast Cancer awareness especially around breast cancer month which is from 1st October – 31st October.

Let us look at the scenarios which will help describe their common reasons to visit the Breast Cancer Charity website.

Scenarios

Scenario 1 – Breast Cancer Patients

Kaitlyn is a 45-year-old breast cancer patient (woman). She wants to keep a track of her progress and the tests she needs to do, she needs to use a task manager, but the task manager also needs to make her relate to it, what is better than a website about Breast Cancer awareness which also helps maintain a task list and acts as a task manager.

Scenario 2 – Breast Cancer Charity, Member

Daniel is one of the founding members of the Breast Cancer Charity and wants to update the content on the Breast Cancer Charity website. Daniel wants to list on the website that apart from the free check-up that the users will get when they register on the website, they will also at the venue get some free goodies from the new Sponsor. To do this Daniel needs to go into the back end of the website and add this new information. Just like Daniel any other charity member who has the admin

access can make these changes as well. They can control what offers are available for users of the website.

Scenario 3 – 30-year-Old woman

Misa is a 30-year-old woman, she has never gotten a breast cancer check up done she hears that on the Breast Cancer Charity website if you register as a user you get a token for free breast cancer check-up. So, she gets on the website, hits the login button types in her email and password and registers as a new user, after doing that she gets a token send to her email which she can use to redeem a free breast cancer check-up on the website to book her appointment using the booking system.

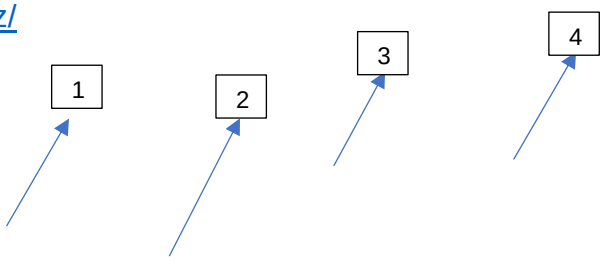
Scenario 4 – 20-year-old man

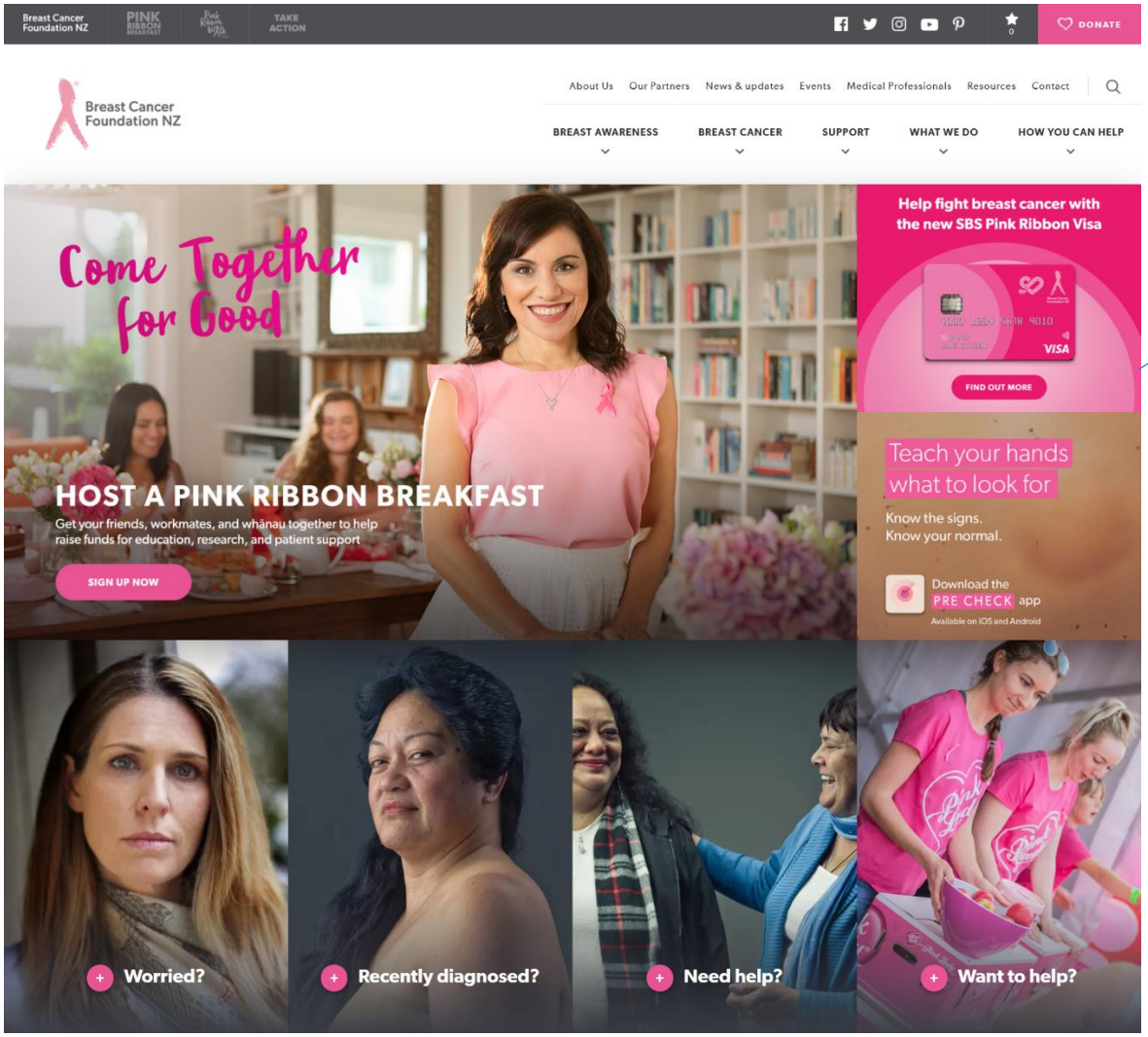
Kevin saw on a Facebook group page that Breast Cancer Charity is giving out free breast cancer check-ups for anybody who registers as a user on the website, he wants to book a free breast cancer check-up for his mother, hence he opens up the Breast Cancer Charity website and registers is mother as a user and then he tell his mother that she can use the booking system on the website to book her free check-up using the token that has been emailed to her.

Competitive Analysis Summary

Competitor No – 1 (Breast Cancer Foundation NZ)

https://www.breastcancerfoundation.org.nz/





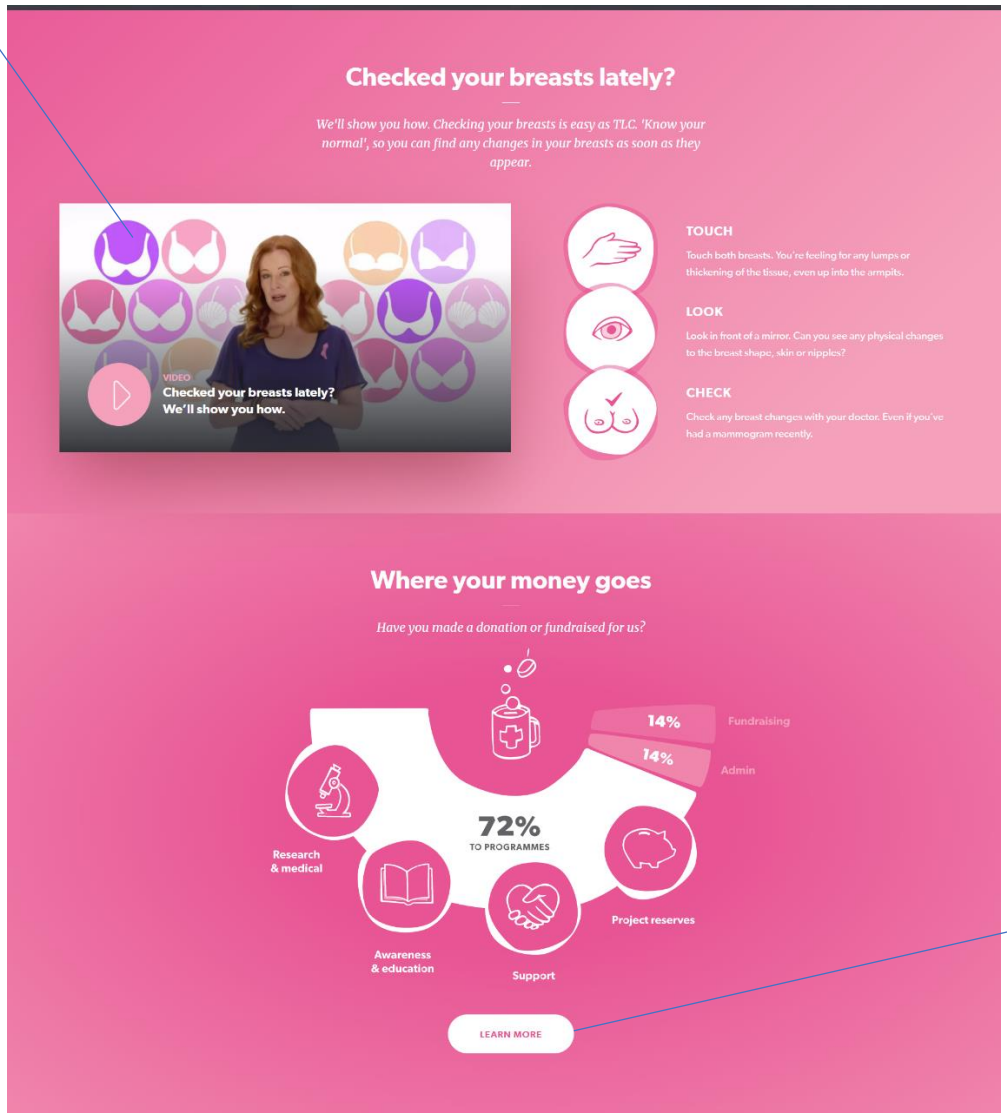
5

Breast Cancer Foundation NZ is one of the biggest Breast Cancer charities in New Zealand. They are based out of Auckland, New Zealand. Their website is a very good one, it has all the information a user would need in fact much more, but I do find their banner a little too busy. Other than the banner the website is very well made, and the colours are very nice too. Their website does look like it has been maintained professionally also it was built professionally. Their website is build in React and few other libraries using php.

1) I cannot tell if this is their primary navigation or their secondary navigation.
2) This feels like a primary navigation and contains the information that the user will be after. I do not like the idea of having both the primary and secondary navigation on the homepage displayed separately.
3) I do like the idea of having social media plugins in the header it makes it much easier for users to follow the social media pages.
4) The Donate button is located perfectly in the header as it's easy to spot and it will be very helpful for users who come to the website to support the cause.

5) The banner is very busy according to me it has a lot going on, there is app information, visa card information and even a sign up for the breakfast club, this could easily confuse users.

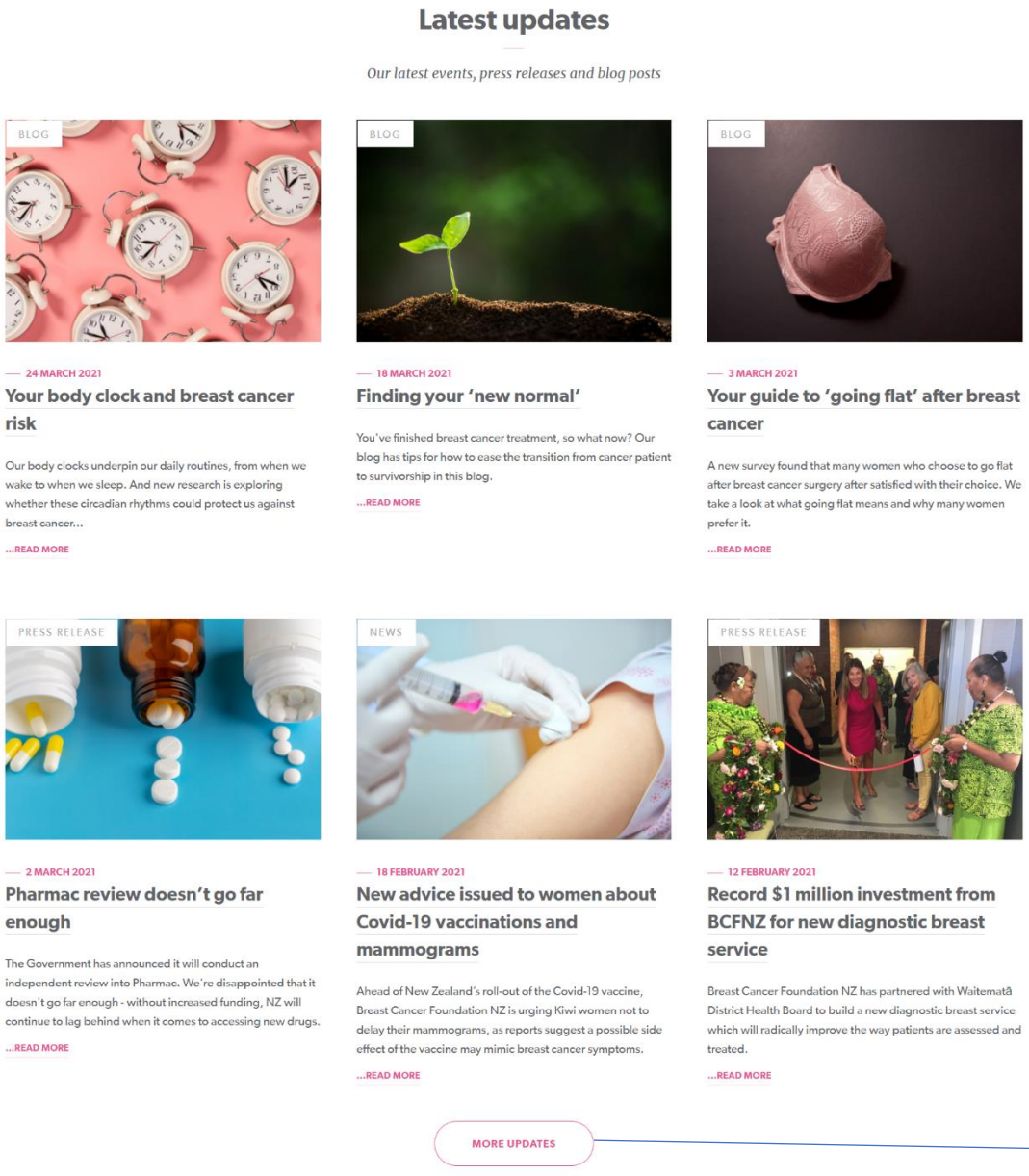
6



7

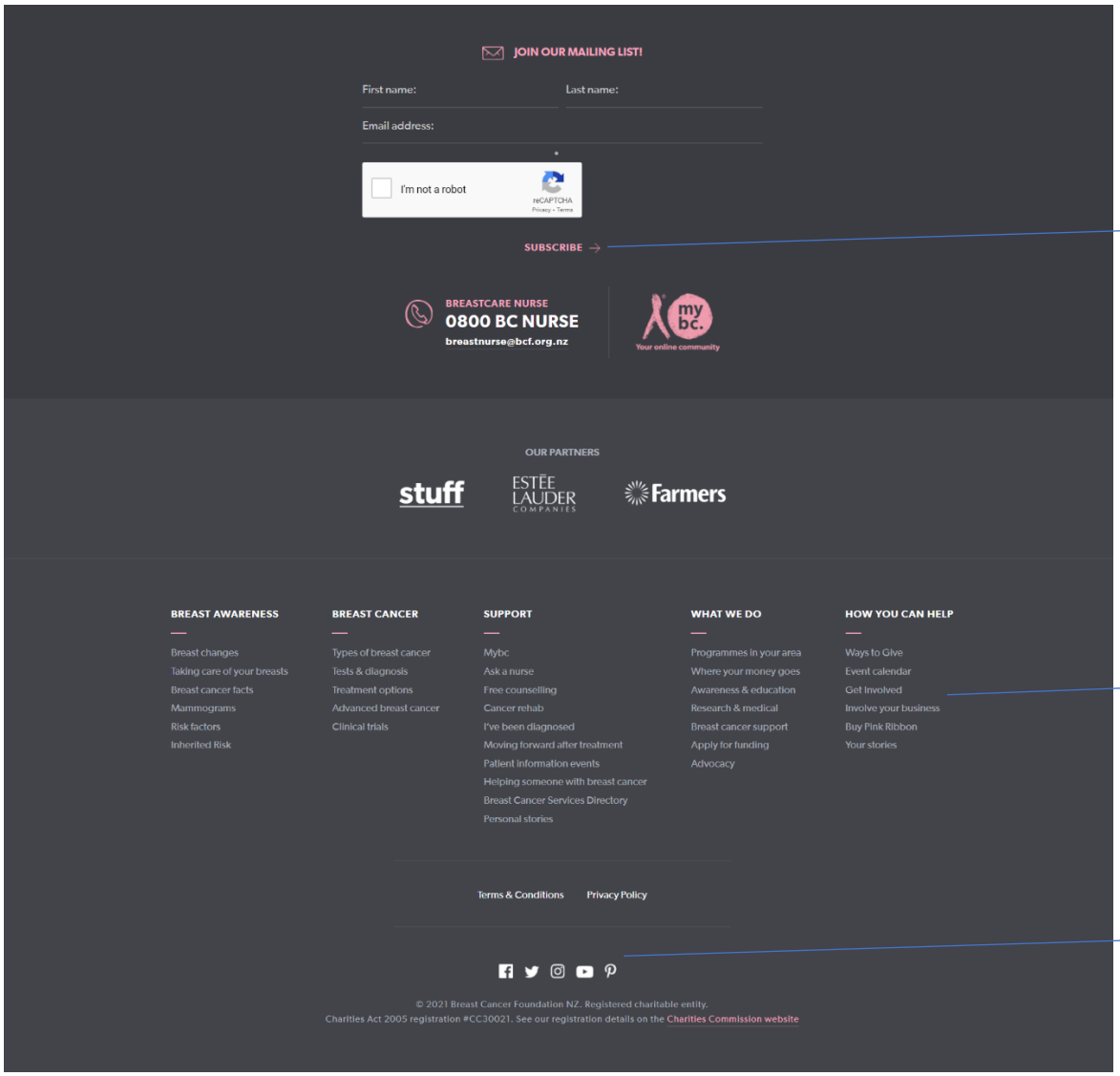
5) This is a very good spot to place the video as it is on the homepage and something that will be helpful for the users.

6) The learn more button takes you to the page which gives a detailed report of how the money that has been donated is being used, this is a good feature if the site generates big chuck of donations.



7) The website does write blogs regularly and has regular press releases hence having them place on the homepage is a good idea, but having just 3 blogs/press release articles would've been a better option than having 6 blogs. But the white space does make the website look nicer.

8) This more updates button directs the user to the blog page where all these blogs live.



9) The subscribe button needs a border because it just seems like simple text or if they even increase the font it will make it obvious as a button.

10) They have repeated the links in the navigation bar, which is a good practice for footers but also, they have linked all the hidden pages, it makes it easier for the user to just get directed to the exact link they are looking for, if they don't want to scroll through the entire website.

11) Since the website does have their social media icons in the header, I do not see the point of having in the footer as well. They should sit either in the header or the footer not both.

Competitor 2 (Breast Cancer Cure)

https://www.breastcancercure.org.nz/

Breast Cancer Cure is also an Auckland based charity. The website is clean and a bit old fashioned. The functionality of the website is good, but the layout and UI are old. The colours are good, and it is a clean website.

1) The primary navigation looks good, but the secondary navigation is a bit complicated, I have never seen contact in “about” tab before, it took a couple of minutes to find their contact info.
2) The secondary dropdowns in the “what you can do” tab is very helpful, as the user knows that is where they will find the information, they are looking of especially if they want to donate or volunteer etc.
3) The Video is placed well on the homepage with some background story about the person who is in the video telling their story, this is a good idea, as instead of just having a written story it is better to have a video where they are talking about their experiences.
4) This is where the sponsors have been listed which I thought was a very good idea, but since our charity has no sponsors now, we won’t be using this feature, but it can be used at a later stage.
5) This is where the website has placed their social media icons, which is usually how the old websites did it.
6) This is where the users can sign up which could be in the header instead of having it in the footer, as it is not convenient for users to find it.

4

BREAST CANCER CURE

PLEASE DONATE NOW.

5

Stay up to date

Sign up to receive the latest research developments and fundraising events in your area.

Email Address

SIGN UP

6

We respect your privacy.

Breast Cancer Cure is registered with the Charities Commission and can be found on the following website: www.charities.govt.nz. Our charities commission number is CC11205. A donation receipt will be issued upon receipt of payment and all donations of $5 or more are tax deductible.

Privacy Policy Terms of Use

Managed by webchores

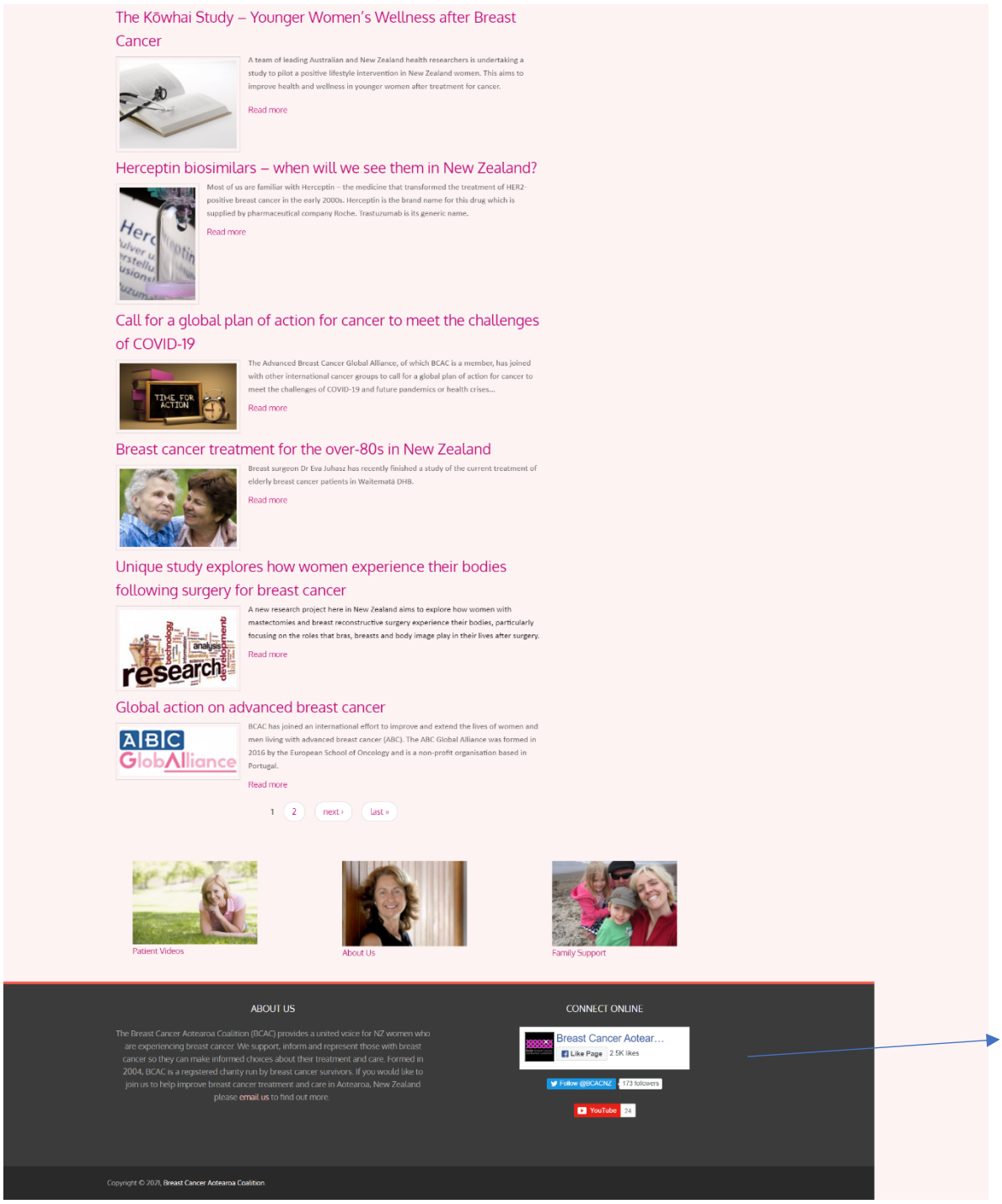
Competitor 3 (BCAC – Breast Cancer Aotearoa Coalition)

https://www.breastcancer.org.nz/

This website is badly designed, this is also a New Zealand based breast cancer charity.

1) The primary navigation is overcrowded.
2) The support pack is very confusing, it is not very clear, looks out of design.
3) The logo is not designed properly, it looks out of place does not go with the brand or it's colours.
4) The donate button looks out of place, it is just placed where they could find space.
5) The home page does not have much but just the articles and the blogs.

6) The footer has the entire Facebook page widget in it. It makes the footer look bad.



6

Section Three Site Content

Content Grouping and Labelling

Taking into consideration the Breast Cancer Charity and the way the client wants the charity to work, and their requirements, we have decided to divide the content of the pages as stated below:

1) Home Page
    - Banner Image
    - Video of a Breast Cancer Patient
    - Book an Appointment.
    - Task Manager.
2) Breast Cancer Page
    - Introduction about Breast Cancer and Image
    - Treatment Options, Tests & Diagnosis
    - Types of Breast Cancer
    - Breast Cancer Story
3) Breast Awareness Page
    - Banner Image
    - How to take care of your Breasts video and text
    - Difference in Nipples or New Lump
4) About Us Page
    - Banner Image
    - Information about Breast Cancer Charity
    - Image and Text about Breast Cancer Charity
5) Book An Appointment Page
    - Appointment Booking System
6) Sign-Up Page
    - Sign – Up system

Functional Requirements

The functional requirement of the website is to create Breast cancer awareness amongst the youth.

As per the client brief the main functional requirements of the website is a login function, a task manager, and a booking system. A log in function will help the user register on the website and get a token emailed to them which can be used to book an appointment for a free check-up using the booking system on the page. The website should also have a task manager which will help the user maintain their tasks for regular medical check-ups.

So a booking system will be embedded in the website, and a task manager as well along with a user login system in place.

Since this is a customised website, the user will have to go into the code to change any information on the website.

Metaphors

Metaphor Exploration – Functional Metaphor

Functional metaphors are the tasks that a user can perform on the website, below listed are all the functional metaphors for the Breast Cancer Charity website.

- Registering on the website
- Signing into the website
- Adding a task to the list, editing the list, or deleting the list
- Booking an appointment on the website

Metaphor Exploration – Visual Metaphor

Visual metaphors relate common graphics elements on your site to others familiar out in the world.

- Banner Images
- Patient Video

- Breast Cancer check Video
- Pink, Grey, Black colours (main color Pink)

Section 4 Site Structure

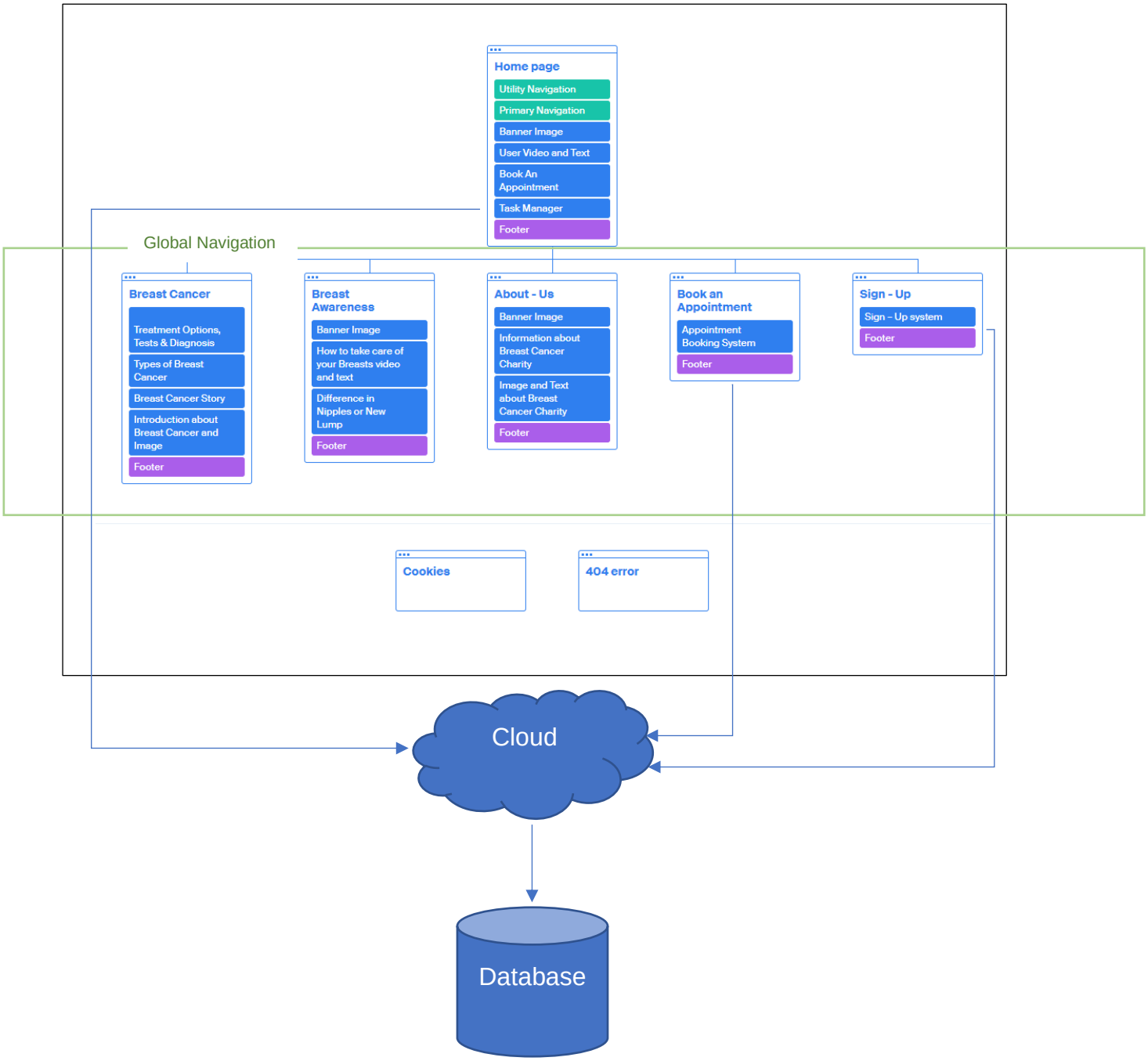
Site Structure Listing

The navigation for the Breast Cancer Charity website will be straight forward and easy for a user to use. After doing the Competition analysis we have come up with the navigation, also making sure it meets the client's requirements.

- On the top left of the website will be the logo which will also be linked to the homepage.
- Next to the logo to the right side will be the search button with search field.
- Next to the search button will be the Facebook plugin for the Facebook page.
- Next to the Facebook plugin will be the Sign-up button which the user will use to Register or Login to the website.
- Below will the primary navigation with the following pages:
    - Breast Cancer Page
    - Breast Awareness Page
    - About Us Page
    - Book an Appointment Page

When the user lands on a page that page will be highlighted in the navigation letting the user know their position on the website. If the user is on the home page none of the navigation tabs will be highlighted.

Sitemap


Home page
Utility Navigation
Primary Navigation
Banner Image
User Video and Text
Book An Appointment
Task Manager
Footer
Global Navigation
Breast Cancer
Treatment Options, Tests & Diagnosis
Types of Breast Cancer
Breast Cancer Story
Introduction about Breast Cancer and Image
Footer
Breast Awareness
Banner Image
How to take care of your Breasts video and text
Difference in Nipples or New Lump
Footer
About - Us
Banner Image
Information about Breast Cancer Charity
Image and Text about Breast Cancer Charity
Footer
Book an Appointment
Appointment Booking System
Footer
Sign - Up
Sign – Up system
Footer
Cookies
404 error
Cloud
Database

Section 5 – Visual Design

Layout Grids

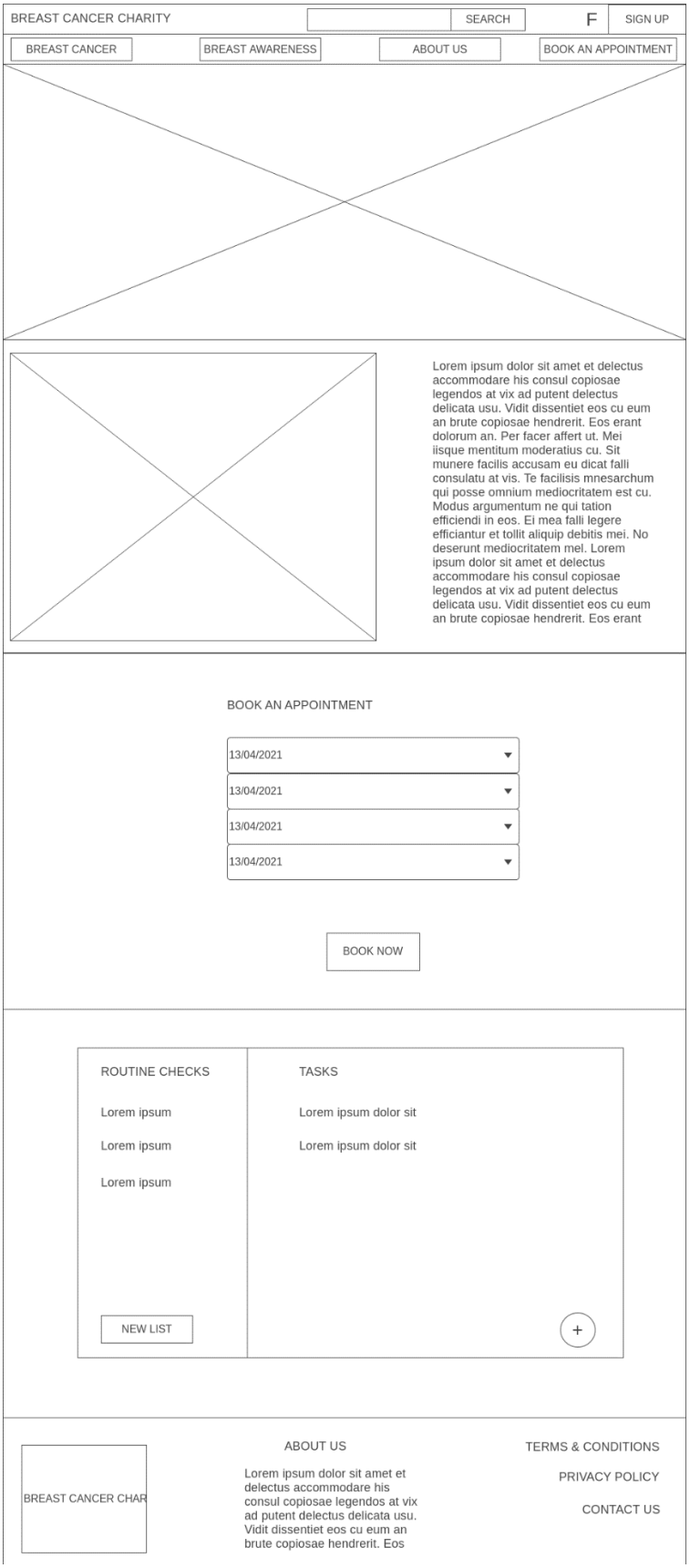
Below are the wireframes and mock-ups for the Breast Cancer Charity website. The wireframe is created using https://wireframe.cc/.

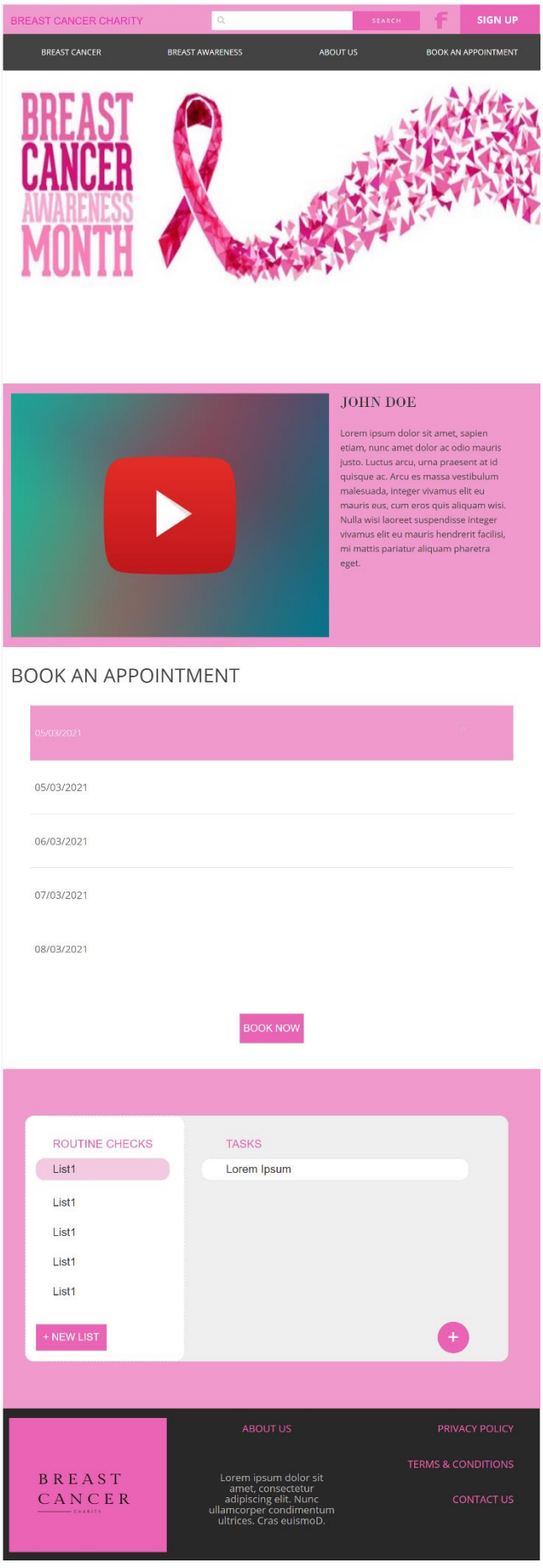
The mock-ups of the pages are created using “Justin Mind” which is an application for windows to create Mock-ups.

There are a total of 6 pages:-

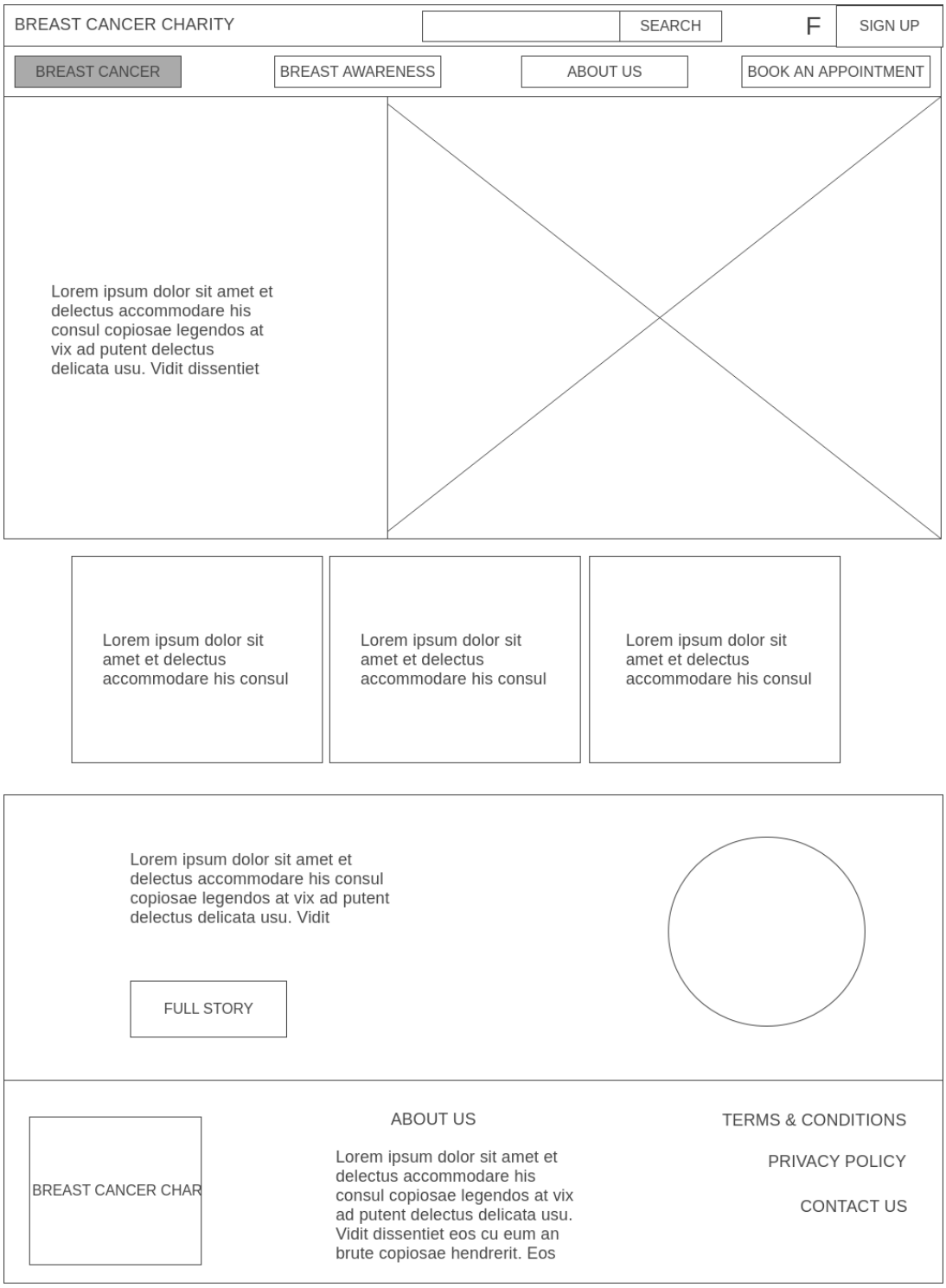
1) Home Page
2) Breast Cancer Page
3) Breast Awareness Page
4) About – Us Page
5) Book an Appointment Page
6) Sign – Up Page

Home Page Layout



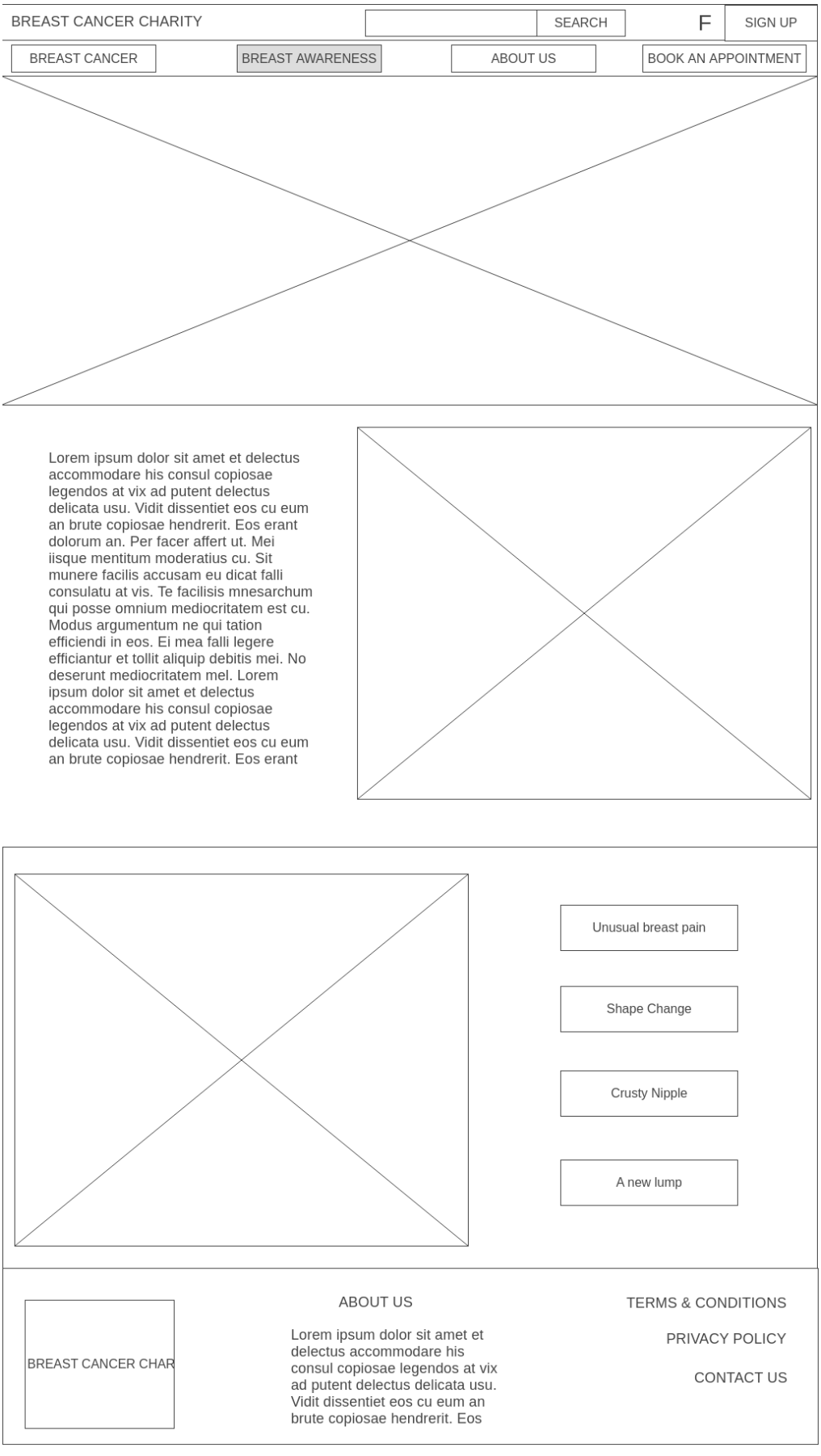

BREAST CANCER CHARITY
SEARCH
SIGN UP
BREAST CANCER
BREAST AWARENESS
ABOUT US
BOOK AN APPOINTMENT
BREAST CANCER AWARENESS MONTH
JOHN DOE
Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu es massa vestibulum malesuada, integer vivamus elit eu mauris eus, cum eros quis aliquam wisi. Nulla wisi laoreet suspendisse integer vivamus elit eu mauris hendrerit facilisi, mi mattis pariatur aliquam pharetra eget.
BOOK AN APPOINTMENT
05/03/2021
05/03/2021
06/03/2021
07/03/2021
08/03/2021
BOOK NOW
ROUTINE CHECKS
List1
List1
List1
List1
List1
+ NEW LIST
TASKS
Lorem Ipsum
BREAST CANCER
ABOUT US
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ullamcorper condimentum ultrices. Cras euismoD.
PRIVACY POLICY
TERMS & CONDITIONS
CONTACT US

Breast Cancer Page Layout




BREAST CANCER CHARITY
SEARCH
SIGN UP
BREAST CANCER
BREAST AWARENESS
ABOUT US
BOOK AN APPOINTMENT
Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu es massa vestibulum malesuada, integer vivamus elit eu mauris eus, cum eros quis aliquam wisi. Nulla wisi laoreet suspendisse integer vivamus elit eu mauris hendrerit facilisi, mi mattis pariatur aliquam pharetra eget.
Arcu es massa vestibulum malesuada, integer vivamus elit eu mauris eus, cum eros quis aliquam wisi. Nulla wisi laoreet
Arcu es massa vestibulum malesuada, integer vivamus elit eu mauris eus, cum eros quis aliquam wisi. Nulla wisi laoreet
Arcu es massa vestibulum malesuada, integer vivamus elit eu mauris eus, cum eros quis aliquam wisi. Nulla wisi laoreet
Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu es massa vestibulum malesuada, integer vivamus elit eu mauris eus, cum eros quis aliquam wisi. Nulla wisi laoreet suspendisse integer vivamus elit eu mauris hendrerit facilisi, mi mattis pariatur aliquam pharetra eget.
FULL STORY
BREAST CANCER
CHARITY
ABOUT US
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ullamcorper condimentum ultrices. Cras euismoD.
PRIVACY POLICY
TERMS & CONDITIONS
CONTACT US

Breast Awareness Page Layout



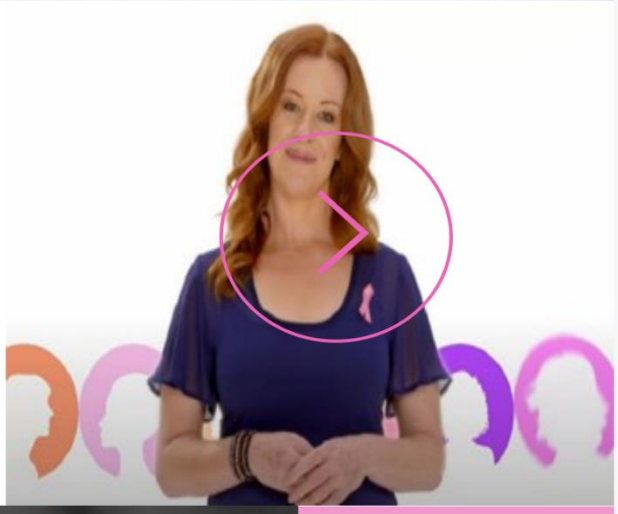
BREAST CANCER CHARITY

SEARCH

SIGN UP

BREAST CANCER | BREAST AWARENESS | ABOUT US | BOOK AN APPOINTMENT

Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu es massa vestibulum malesuada, integer vivamus elit eu mauris eus, cum eros quis aliquam wisi. Nulla wisi laoreet suspendisse integer vivamus elit eu mauris hendrerit facilisi, mi mattis pariatur aliquam pharetra eget.



UNUSUAL BREAST PAIN

SHAPE CHANGE

CRUSTY NIPPLE

A NEW LUMP

BREAST CANCER CHARITY

ABOUT US

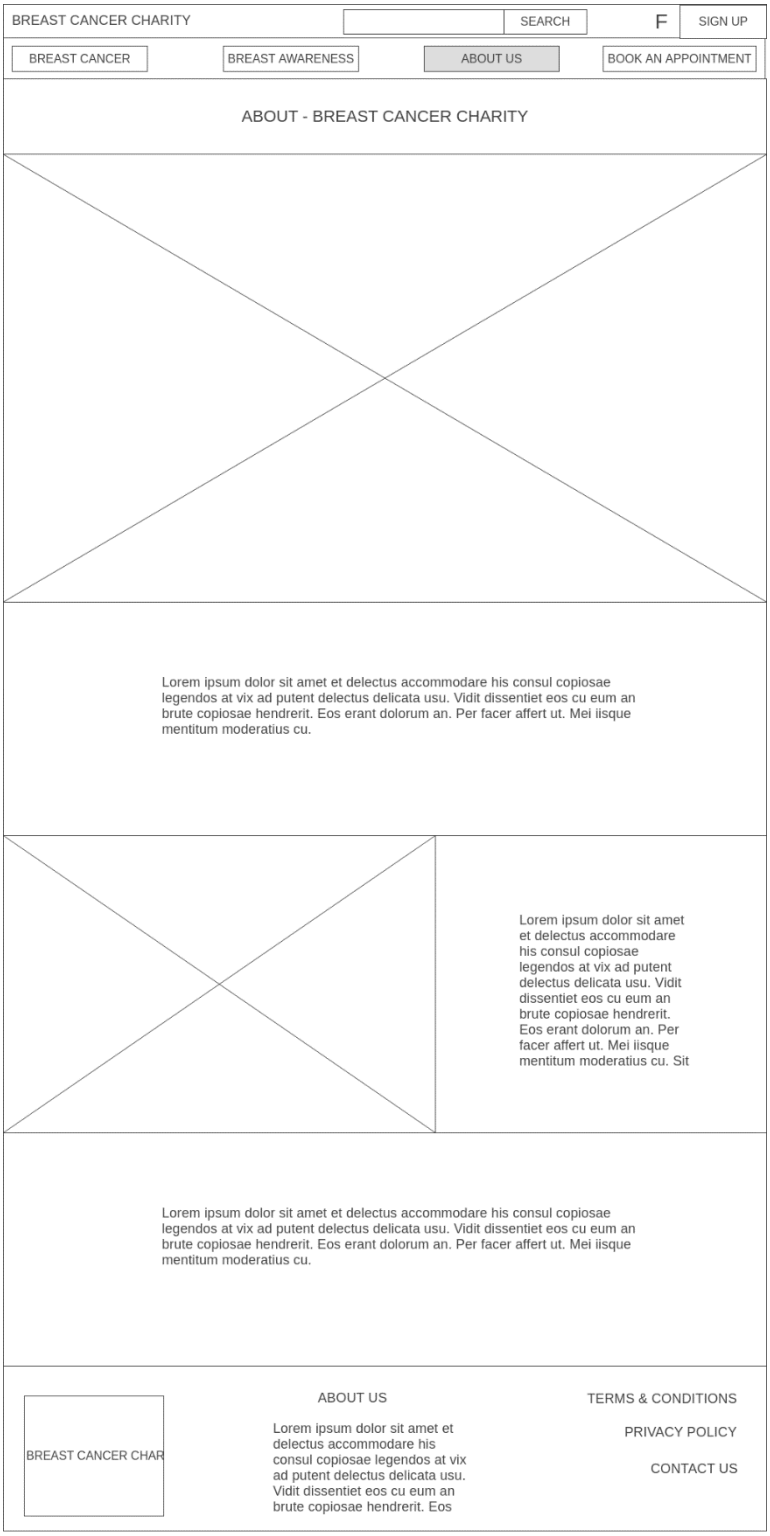
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ullamcorper condimentum ultrices. Cras euismoD.

PRIVACY POLICY

TERMS & CONDITIONS

CONTACT US

About - Us Page Layout



BREAST CANCER CHARITY

SEARCH

SIGN UP

BREAST CANCER | BREAST AWARENESS | ABOUT US | BOOK AN APPOINTMENT

ABOUT - BREAST CANCER CHARITY

Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu es massa vestibulum malesuada, integer vivamus elit eu mauris eus, cum eros quis aliquam wisi. Nulla wisi laoreet suspendisse integer vivamus elit eu mauris hendrerit facilisi, mi mattis pariatur aliquam pharetra eget.

Lorem ipsum dolor sit amet, sapien etiam, nunc amet dolor ac odio mauris justo. Luctus arcu, urna praesent at id quisque ac. Arcu es massa vestibulum malesuada, integer vivamus elit eu mauris eus, cum eros quis aliquam wisi. Nulla wisi laoreet suspendisse integer vivamus elit eu mauris hendrerit facilisi, mi mattis pariatur aliquam pharetra eget.

BREAST CANCER
CHARITY

ABOUT US

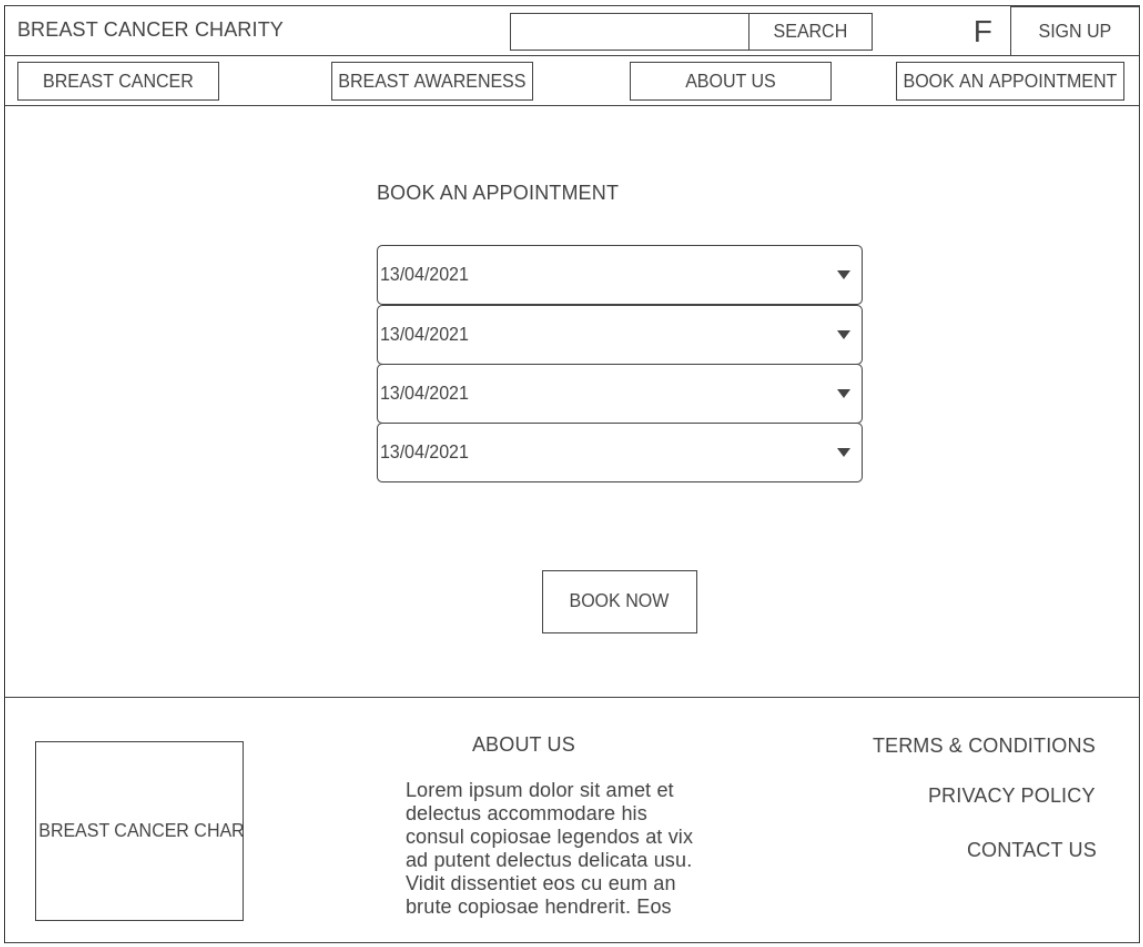
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ullamcorper condimentum ultrices. Cras euismoD.

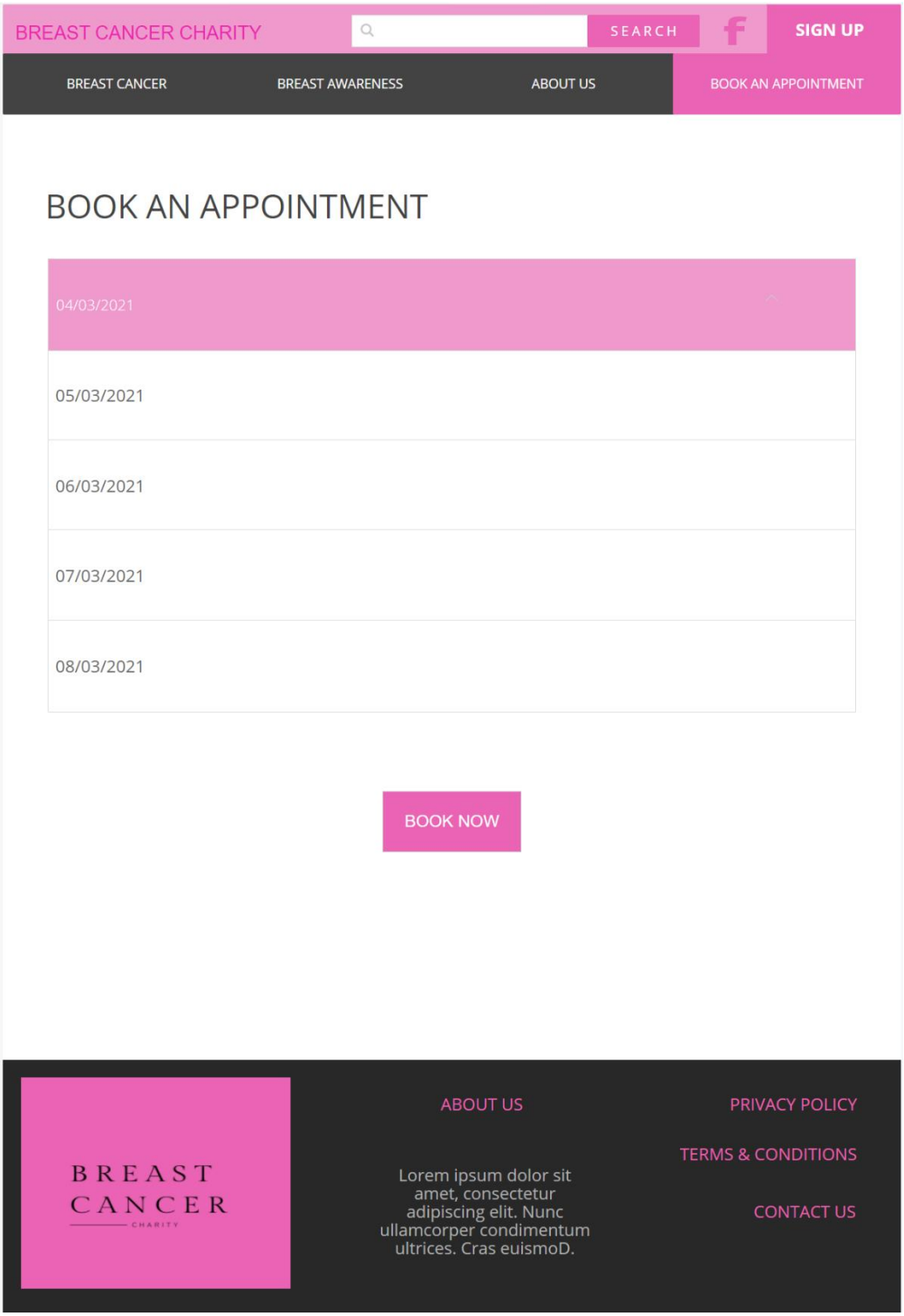
PRIVACY POLICY

TERMS & CONDITIONS

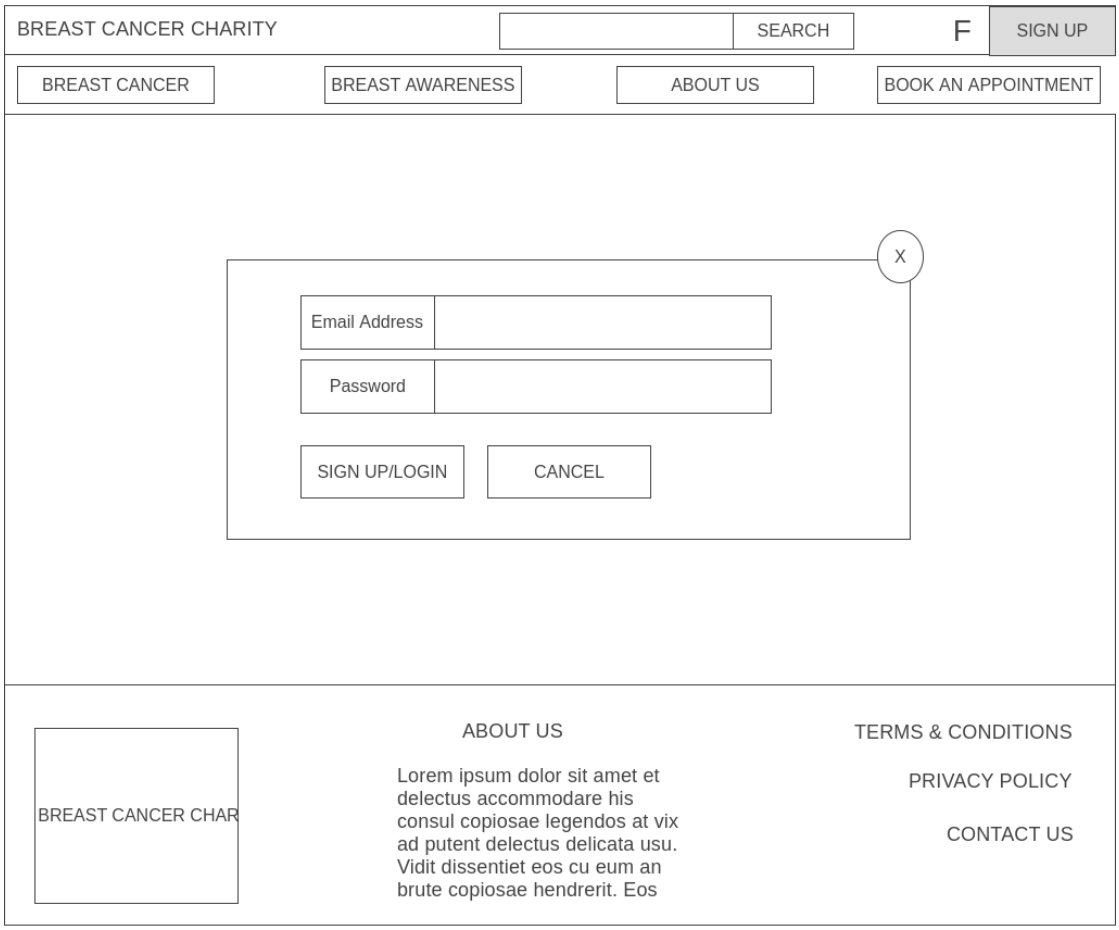
CONTACT US

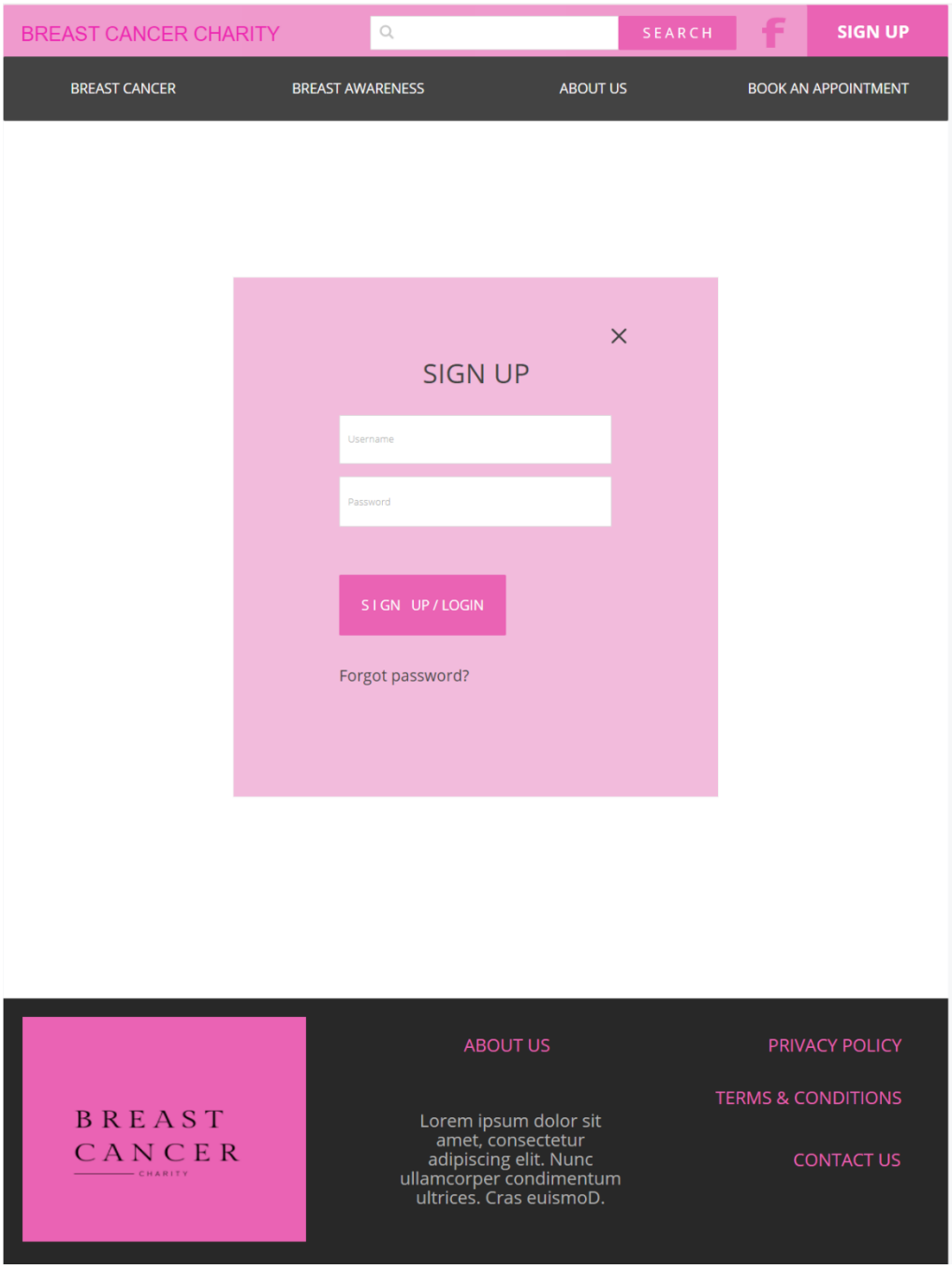
Book an Appointment Page Layout




BREAST CANCER CHARITY
SEARCH
SIGN UP
BREAST CANCER
BREAST AWARENESS
ABOUT US
BOOK AN APPOINTMENT
BOOK AN APPOINTMENT
04/03/2021
05/03/2021
06/03/2021
07/03/2021
08/03/2021
BOOK NOW
BREAST CANCER CHARITY
ABOUT US
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ullamcorper condimentum ultrices. Cras euismoD.
PRIVACY POLICY
TERMS & CONDITIONS
CONTACT US

Sign-Up Page Layout




BREAST CANCER CHARITY
SEARCH
SIGN UP
BREAST CANCER
BREAST AWARENESS
ABOUT US
BOOK AN APPOINTMENT
SIGN UP
Username
Password
SIGN UP / LOGIN
Forgot password?
BREAST CANCER
CHARITY
ABOUT US
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ullamcorper condimentum ultrices. Cras euismoD.
PRIVACY POLICY
TERMS & CONDITIONS
CONTACT US

Section 5 CRUD Analysis

CRUD Table

User Table

UserID	Email Address	Password	Admin Email Address	Admin Password
C	C	C	C	C
R	R	R	R	R
R	RU	RU	RU	RU
R,D	R,D	R,D	R,D	R,D

Task Manager Table

TaskID	ListID	Task	List
C	C	C	C
R	R	R	R
R,D	R,D	R,U	R,U
		R,D	R,D

Booking Appointment Table

UserID	BookingID	Booking Details
R	C	C
	R	R,U
	R,D	R,D

API Prototype

Githib URL - https://github.com/praptimane26/BreastCancerCharity

Below is the link to the API prototype. I have created the Task manager API first, as it performs the same functions as the User, but I just decided to create and dabble with that one first. So far the API works fine it can create a list, retrieve, update and delete the list.

To run the API – in the API folder start Code and then enter in the terminal “nodemon app.js”

To run the frontend – in the frontend folder start code and then enter in the terminal “ng serve”

Prapti Mane

Assignment - 3

Purpose of web frameworks

Web framework is a software framework which is created to support the development of dynamic websites, web applications and web services. The development of a dynamic website is made easy with the use of a web framework. It provides a basic pattern of the code which helps quicken the process of web development. Web frameworks help simplify the process of web development and keep it as modern as possible. Web frameworks automates the repetitive web development tasks. Web frameworks gives you a process for turning an idea into code. A framework is a basic conceptional structure. Most of the applications have a common set of functionalities such as handling session data validation etc, and web framework prevent a developer from re-writing the same code every time to create a web app. The purpose of frame works is to allow developers to build unique features for their web projects instead of rewriting the same code repeatedly. Each framework may provide you a wide range of web app functionalities, resulting in a less error-prone program. In a nutshell, a framework aids in the prototyping, design, and implementation phases of the app development lifecycle, as well as the continuing maintenance and enhancement of a web application. Frameworks are very helpful when it comes to online security. Coming up with your own security is costly and dangerous if you do not do it right. To make anything more resilient, a framework can incorporate security aspects into its functioning and combine bug-fixing efforts. A framework can be secure by default, at least in theory. Most applications/websites these days require login, signup, password rest, authorization, API's, logging, database, sending emails, payment processing, deployment, input validation etc, if a beginner must implement all of this in their design without a framework, they would be lost, but with the help of a framework these tasks can be implemented easily and can be reused if possible.

There are two types of Frameworks Client side(frontend) and Server(Backend) side frameworks

Server side – These control the database, the logic of the application, payment methods, security of the website and the server configurations. Some of the common server side frameworks are, expressjs, ruby on rails, Laravel, Django etc

Client side – Everything that you see on the website in the UI is controlled using the client side frameworks. Some of the common client side frameworks are, vue.js, angular, react etc.

Common Framework Features

Some of the common features in a web framework are:-

Caching – Web caching helps save documents while reducing bandwidth consumption., server load and perceived lag. A web caches stores copies of pages that travel through it and if certain requirements are met, the following request can be served from the cache. Some application frameworks have features for caching documents and skipping certain stages of page preparation such as database authentication. (Web)

Security – Most frameworks these days come with authentication and authorization, which helps with the login, register etc of a web application, this helps restrict access to users which don't meet the defined criteria. Security in a system allows users to view pages based on their roles and has a web-based interface for establishing users and assigning them roles. (Web)

Database access, mapping and configuration - Many web frameworks provide a single API to a database backend, allowing web applications to interact with a range of databases with no code modifications and programmers to focus on higher-level concepts. (Web)

URL Mapping - The process through which a framework reads URLs is known as URL mapping or routing. Shorter, more "friendly URLs" can be used with a URL mapping system that employs pattern matching or rewriting to route and manage requests, boosting the site's simplicity and allowing for easier indexing by search engines.

AJAX (“Asynchronous JavaScript and XML”) – Is a technique which is used for creating web applications. The goal is to make web pages feel more responsive by exchanging little pieces of data with the server behind the scenes, rather than reloading the complete page every time the user demands a change. This is done to improve the interactivity, speed, and usability of a web page.

Web Services - Some frameworks include web service creation and provisioning features. These tools may be comparable to those found in the rest of the web application.

HTML (or Hypertext Markup Language) - It is a computer language used to build websites.

CSS (or Cascading Style Sheets) – It is a language used for styling websites. HTML elements are represented using CSS on a web page. It helps with design, layout, screen sizes etc. It helps beautify a web page.

Components – Web Component-based custom components and widgets will function in all modern browsers and may be used with any JavaScript library or framework that works with HTML.

Features used in Website

The features which are used in the project are URL Mapping, the URL routing is more user friendly. So, the users can easily tell what page they're on instead of reading through some complicated routes. We have also used the security feature on the website, we have used JWT authorization to always keep the user accounts secure. We also have different login accounts of different users, e.g., the content visible to an admin user is restricted to an admin account, whereas the user account will only have access to the content we want visible for regular users. Hence the security on the website has been given priority. We have also used Database access, mapping, and configuration, we have used MongoDb to store our database. We also have the feature of web service on the website since we do have a server running on our computer, listening to port over a network, serving our web documents. (htt21). For the front end we have used components (Material components). These are interactive building blocks for creating a user interface. Angular has a proper documentation which explains well how to use these components on the website. Everything you see in the UI is made up of different components.

Proposed Web Frameworks

The two frameworks we recommend for this project are Angular and VueJs.

Why Angular:

Because it eliminates the need for extraneous code, Angular ensures that development is simple. It has an MVC architecture that eliminates the need to write getters and setters. Because directives are not part of the app code, they can be controlled by another team. Developers can expect less coding and lighter, faster apps in general.

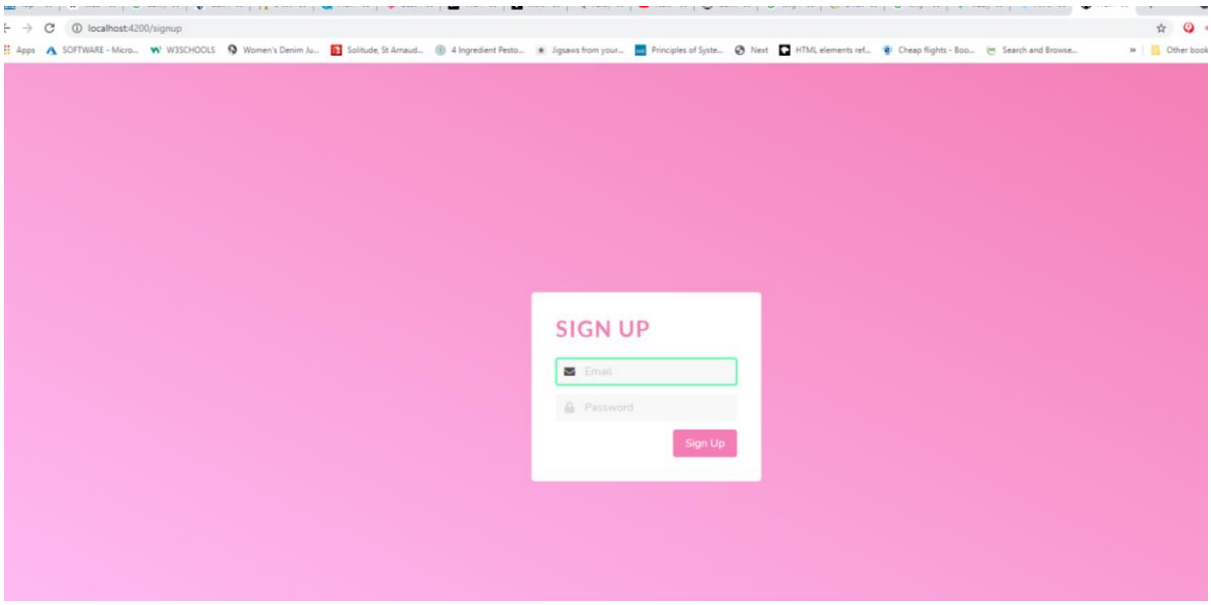
Why VueJs:

Vue.js is a JavaScript progressive framework for building web interfaces and one-page applications. Vue.js is utilized not only for online interfaces, but also for desktop and mobile app development utilizing the Electron framework.

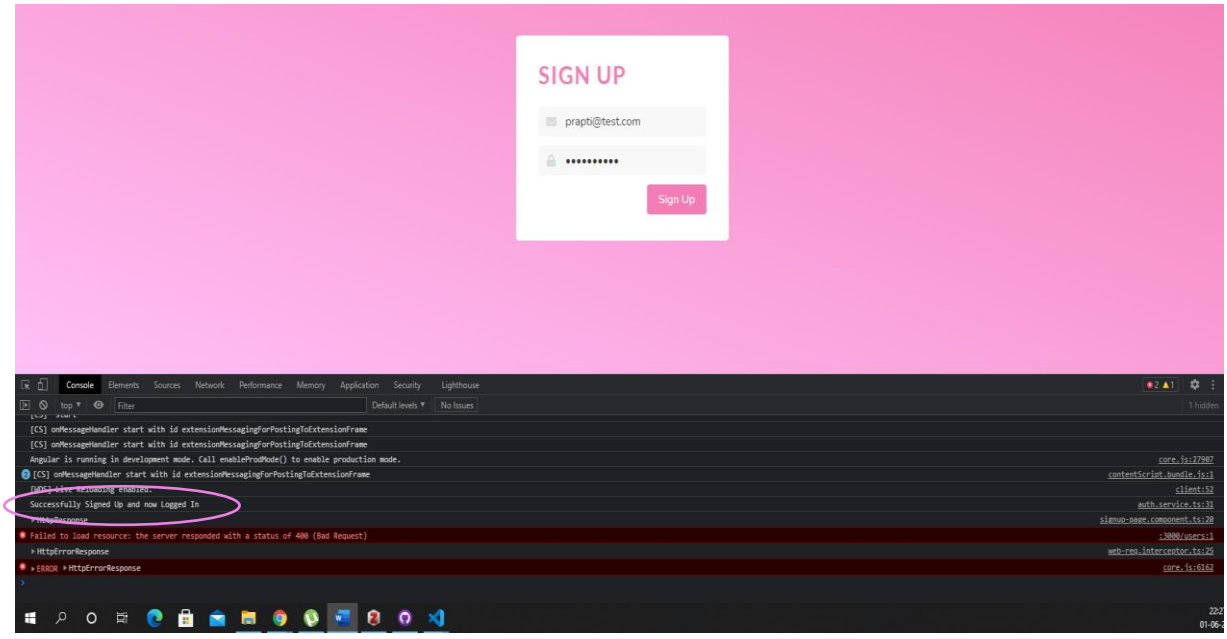
Register, Login, Administer Accounts

Angular

Below is the screenshot of the Sign Up Page using Angular - http://localhost:4200/signup



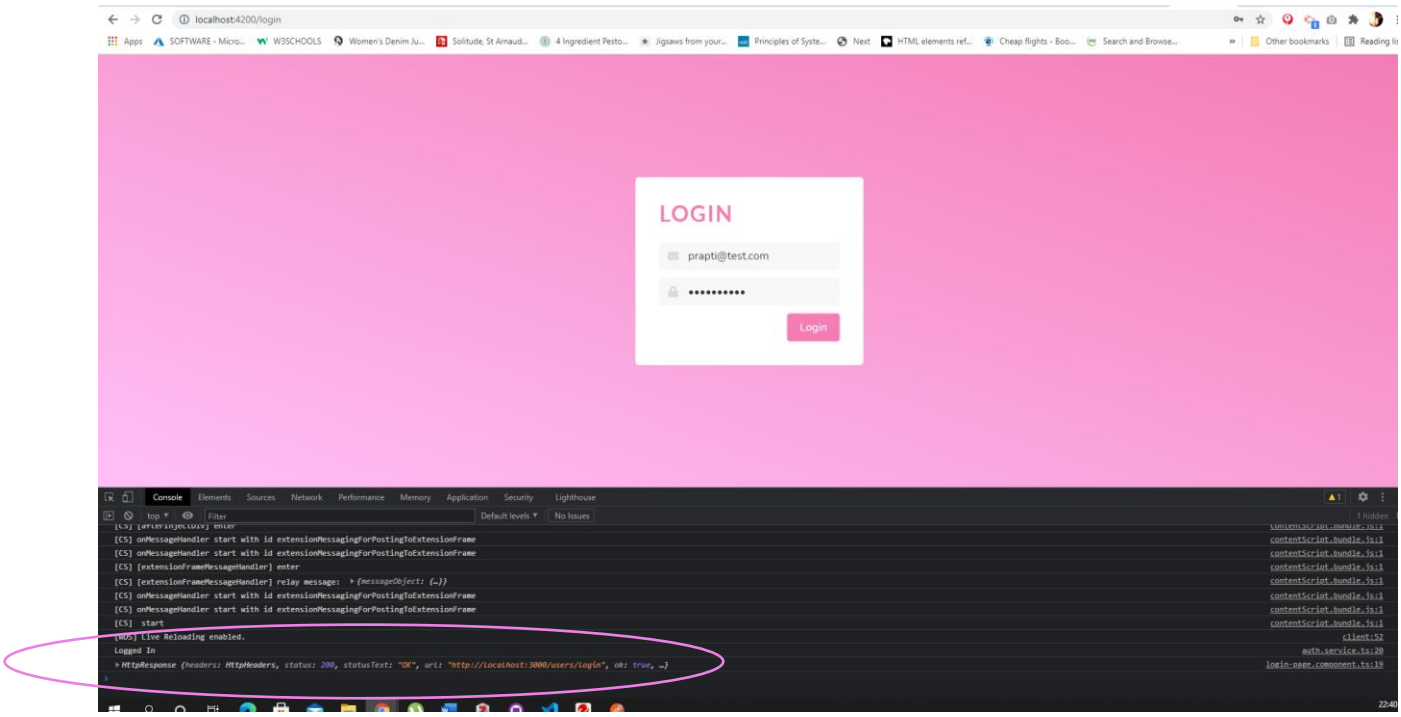
If you inspect the page after putting in your details and clicking the sign up button, you will see in the console that you get a “successfully signed up and logged in” message. Below is the screenshot showing that information



Below is the screenshot of the login page - http://localhost:4200/login

We logged in using – prapti@test.com password – helloworld

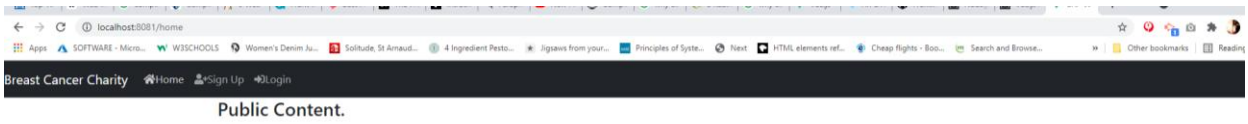
Once you inspect the page and check the console you should be able to see the “logged in” message with the status as 200 and true. Which means the user has successfully logged into the app.



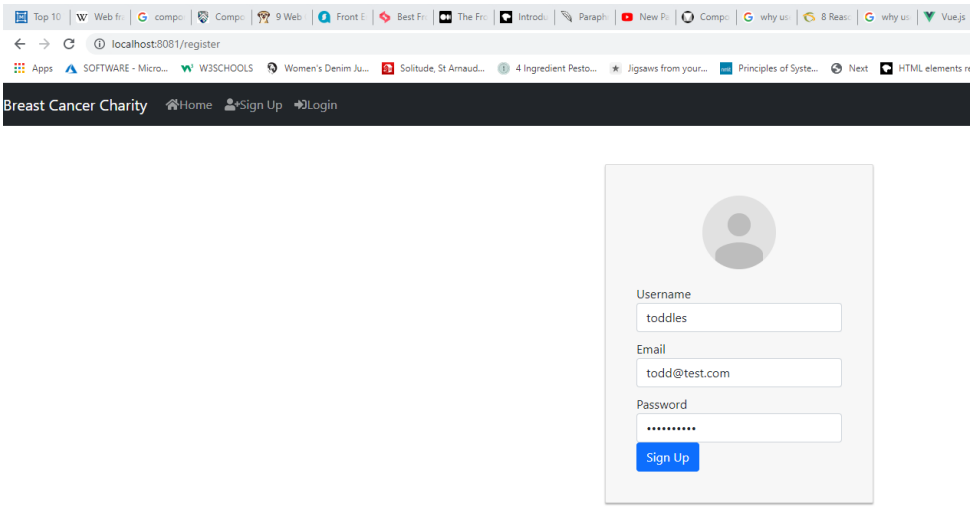
At this stage for Angular we do not have different login’s set up, but for the final product we will have different login and content for Admin, Moderator and User, we do have a working example of that in VueJs which we will show you next.

VueJs

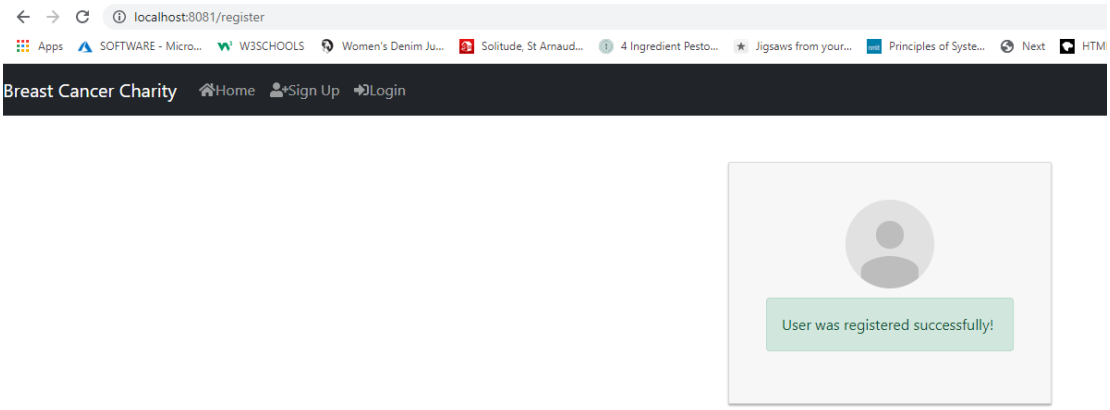
Below is the main screen which will consist of the public content of the website, which is general content, so any information about the charity etc and content that is not targeted towards certain users only will be on this page. This will be the Homepage of the website. We just have the skeleton of the website at this stage for demo purpose.



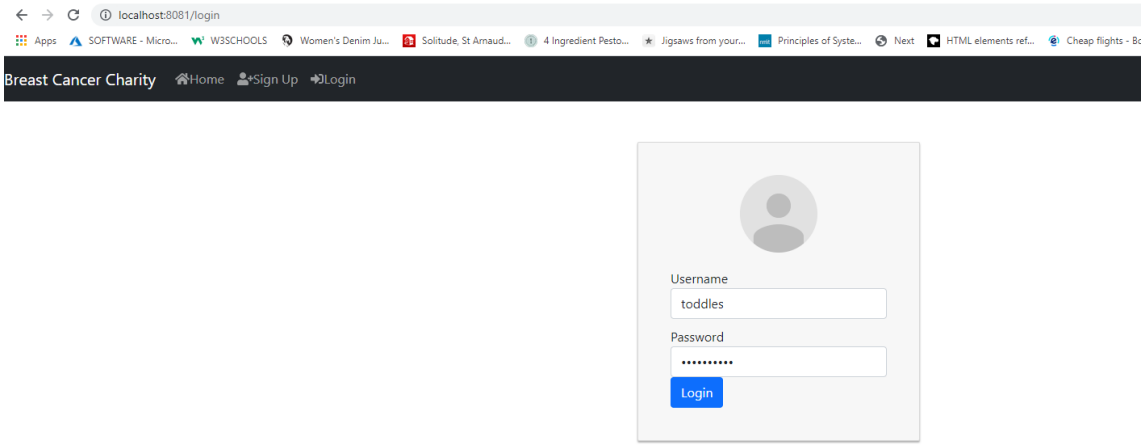
Click on the Sign Up in the navigation bar and it will direct you to this page - http://localhost:8081/register



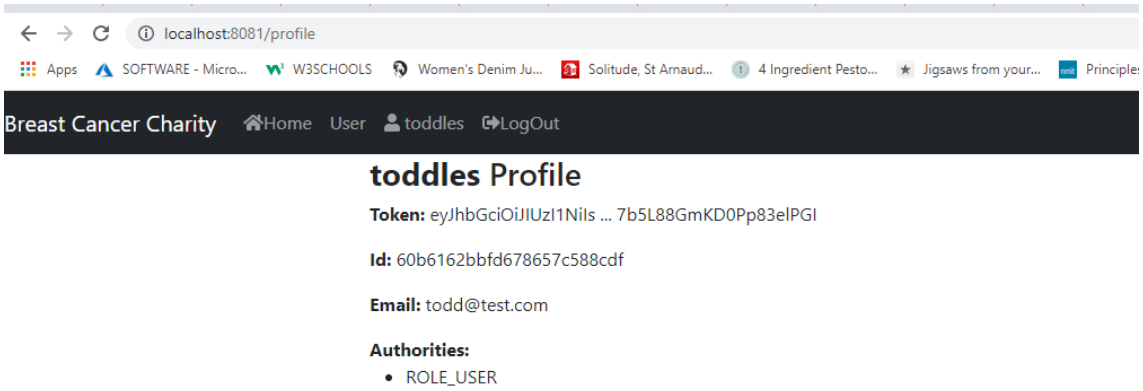
Enter you details as shown in the above picture and then press the “Sign Up” button. Once you register you will be shown the below screen



Then go to the Navigation bar and click on the Login button, you will be directed to the below screen : http://localhost:8081/login

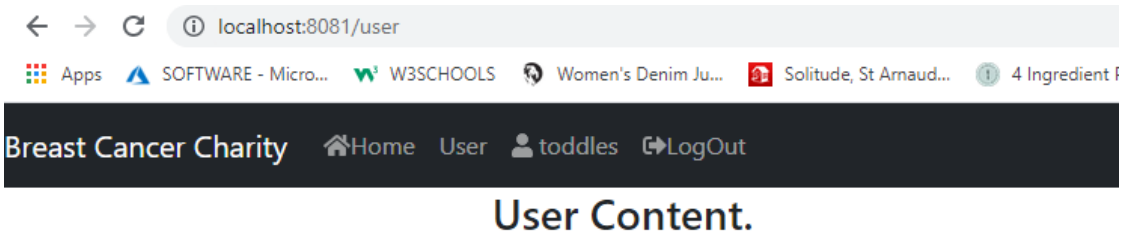


Enter your details and click on the “Login” button, which will take you to the below screen, which is the Profile screen - http://localhost:8081/profile



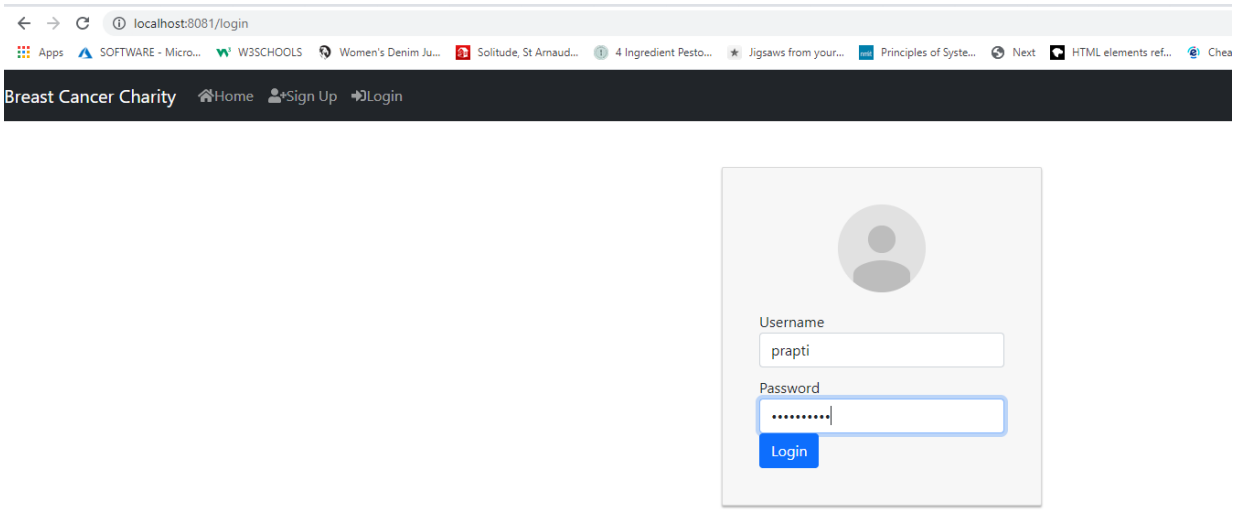
As you can see in the above image, the profile that we have created is for that of a User, so when a regular user goes onto the website and register themselves, they only have access to the content that is created just for the user's. If you click on "User" in the navigation bar it will direct, you to the below screen.

http://localhost:8081/user

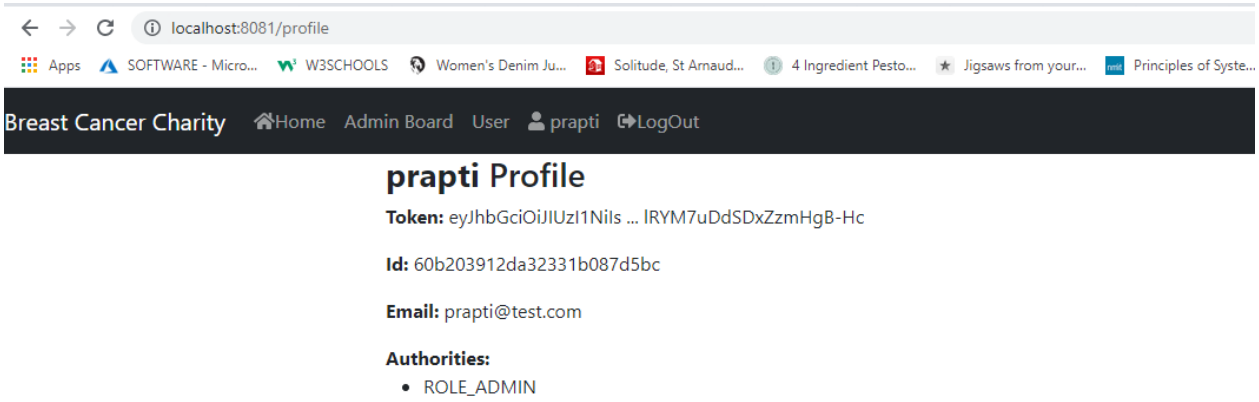


As you can clearly see that the content that the content on this page is specifically designed for the users.

If you want to create a Moderator or Admin account, it can be done through the backend. I have created an Admin account using Postman for this website. I am going to login into that account and show you the difference between the two. As you can see there is no way a user can register as an Administrator unless they do it through the backend, which is a good option as you do not want everyone to have access to the admin account.



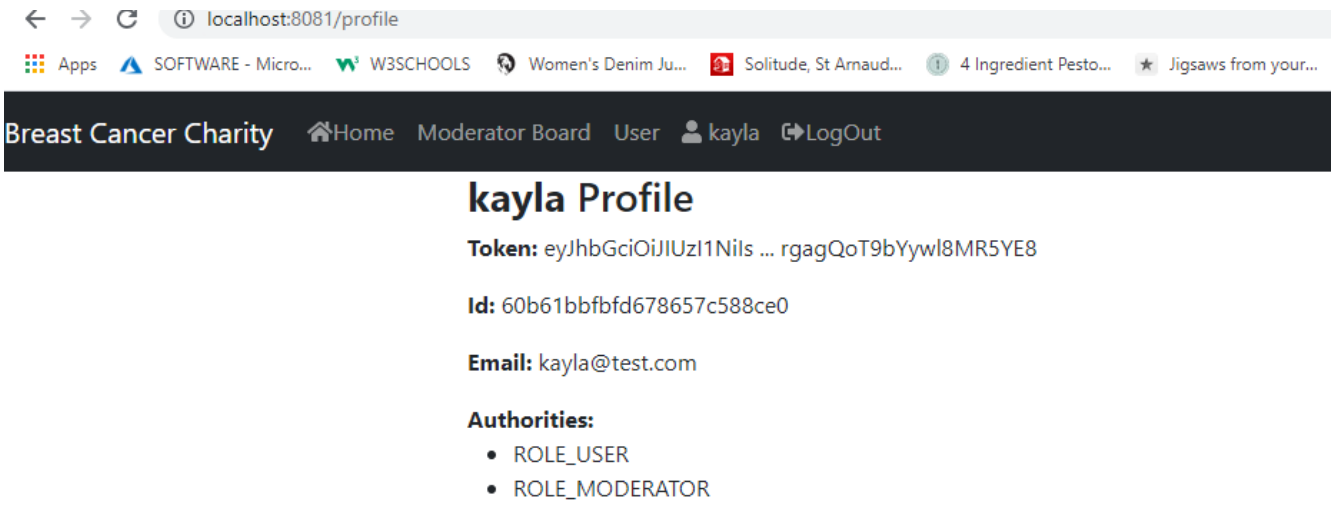
In the above screen as you can see I am going to login as “Prapti” since I have already created an Admin account for me. Once you click the “login” button you will be directed to the below screen

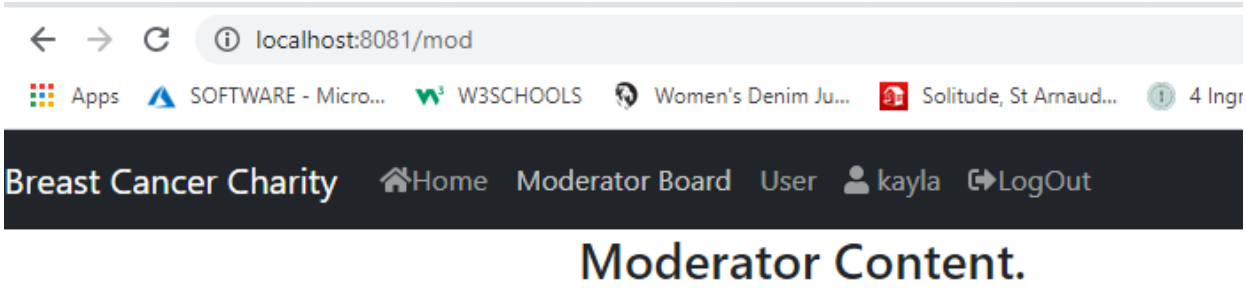


As you can see once logged in as an Administrator, I have access to the admin board, which you cannot see from the user account, but I also have access to the “user account” content. As an admin I can access anything on the website but a regular has access to limited content on the website, they will only see what we show them.

We also have a third type of account on the website which is a moderator account. I have logged in using “kayla” account who is the moderator for the

website, but also have access to the user account. Below is a screenshot of the dashboard for a moderator account.

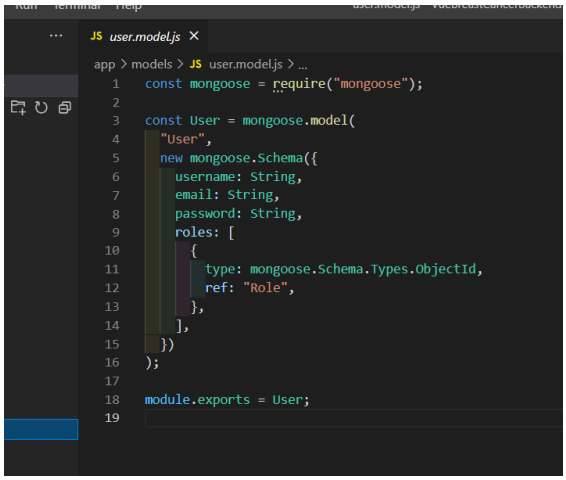




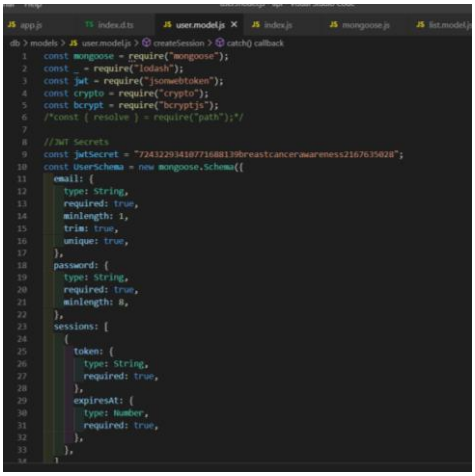
As you can see Kayla's account has access to the Moderator content and the user content but not the Admin content. These roles can be changed using the backend/ postman.

Code Snippets

VueJs – User Model



Angular – User Model



The code in VueJs is pretty straight forward, whereas in Angular in User Model most of the methods are especially including the refreshing of JWT tokens it is all set up in the User Model making the code far more complicated to read. Although this can be sorted if we create a separate model in angular which will deal with the Access tokens this issue can be sorted. The helper methods everything is in the UserModel in angular at this stage. This code could've been written better by segragating it.

The code in VueJs is simple and not at all complicated, it shows exactly how the UserModel is using Mongoose and what values are needed for a user. This is how the data will be collected in MongoDb database. It is easier to understand that the User object will have a roles array that contains ids in roles collection as a reference.

Comparing Frameworks – Registration and Login

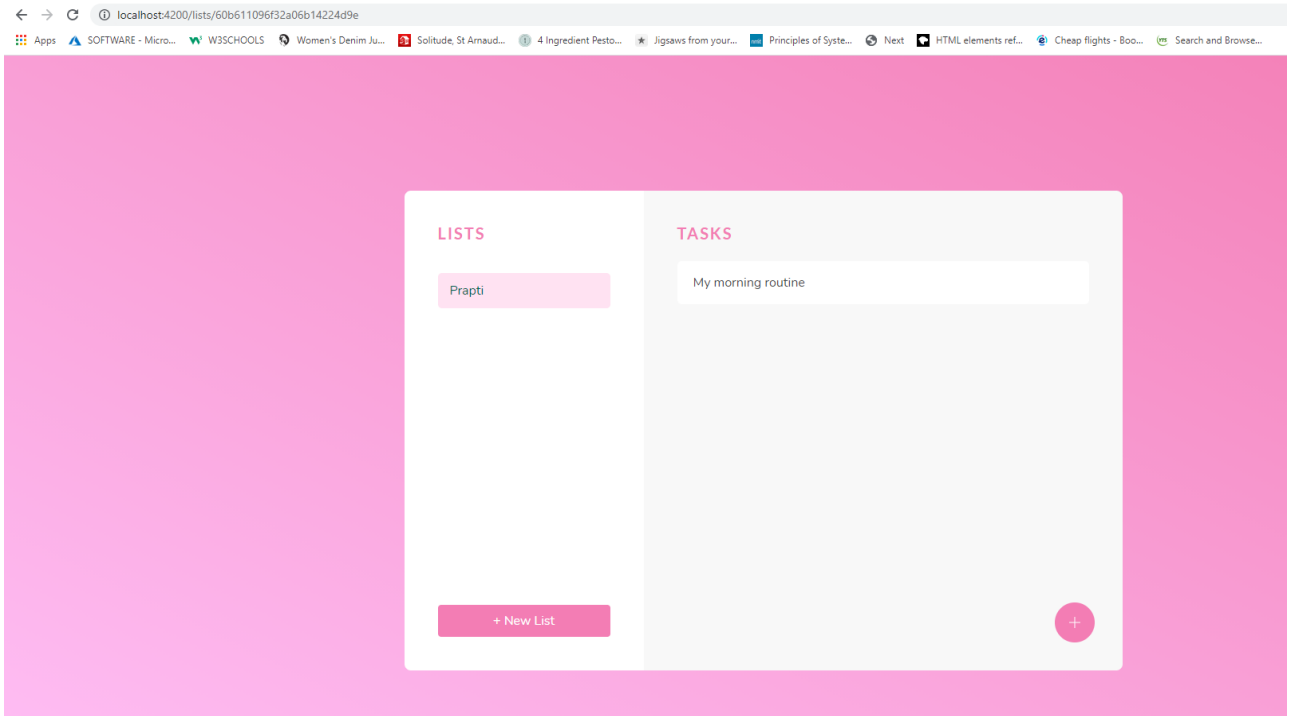
When it came to coding the registration and the sign-up part of the website, I did find the VueJs framework to be easier than Angular. Although both the websites do have similar patterns, certain parts in Angular are much more complicated than VueJs, as mentioned online VueJs initially was inspired by Angular, and any complicated parts of Angular were made simpler in VueJs hence the code also looks similar but is much more complicated in Angular. The security side of things when it comes to login and registration felt the same, although applying JWT felt a bit more complicated in Angular as it did in view, and as you can see in View it was easier to accomplish the exact login functionality as per the requirement whereas in Angular so far we have only been able to login as a regular user with no defined roles. One of the major drawbacks I found in Angular is that Typescript is mandatory. Vue is more flexible when it comes to that.

Angular deals with Modules whereas Vue is about Components. Angular is all about MVC, which we have worked with in react in the past, so understanding it was slightly easier.

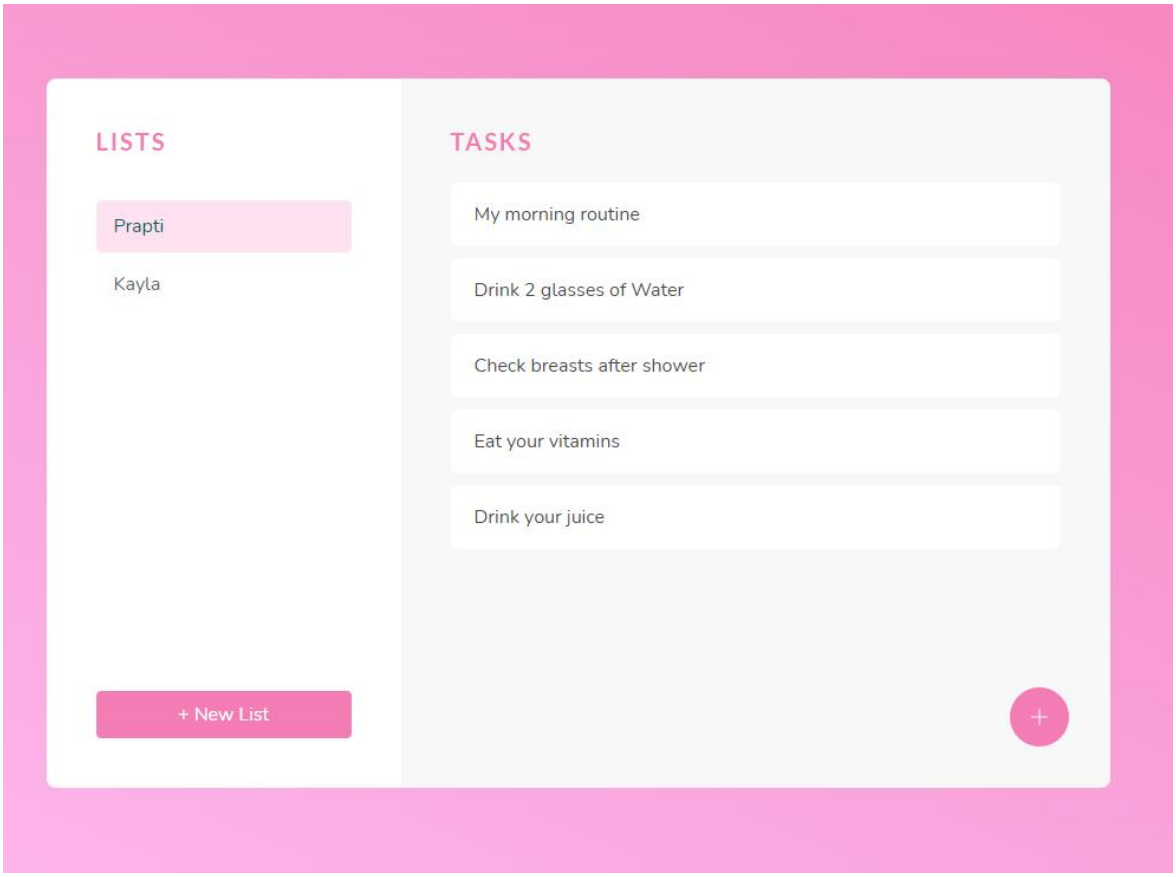
Members Use the Website to register their products.

Angular

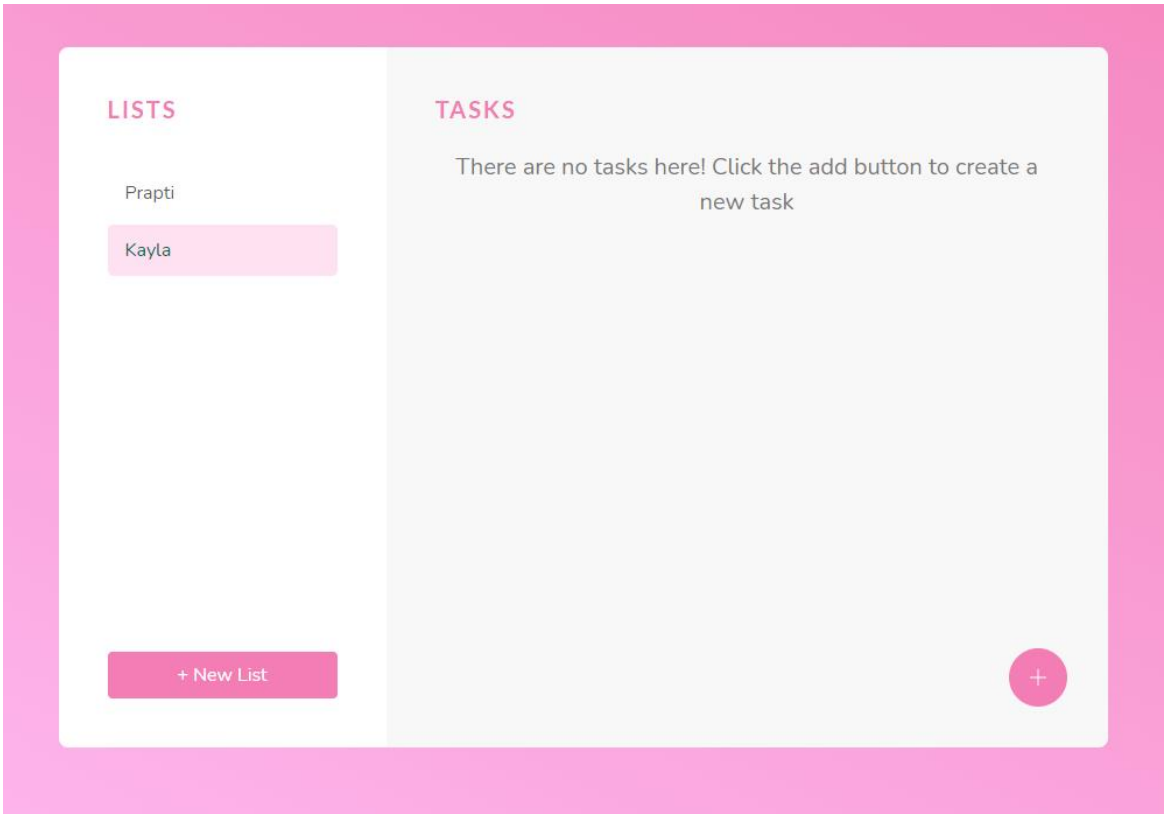
Once logged into the website a user can create a list of tasks, each list will consist of it's own task, this can be used by the users to keep track of their health and their routine checkups. Below is a screenshot of what the list and tasks look like on the website.



The lists can only be seen if the user is logged in. Since I am logged in as Prapti, the tasks you see in the list are part of my account. To add a new list you need to click on the "new list" button and you can create a new list, and to add a task into that list just click on the list and then click on the "+" button on the right side corner of the page, this will allow the user to create a task for that particular list. Let me add few more tasks to my list and show you what it will look like.



I have added some more tasks in the list called Prapti, I have also added another list called “Kayla” this has no tasks in it, let me show you a screenshot of what an empty list looks like.

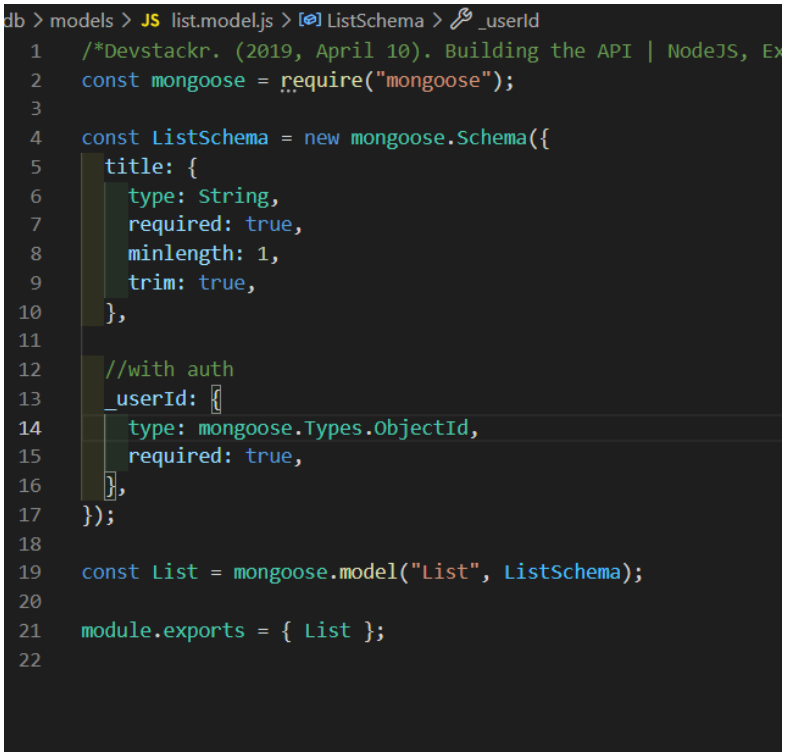


VueJs

We can create the same functionality in Vue, by creating a list model and task model in the backend. And then using the routes in the server.js file direct to these models. And we can make lists based on the roles and the users logged in. And we can populate the content of the website using the App.vue. We would have to create List.vue file and then add it to the navigation and give it the path using the router.js. We would have to load it in the mongoose models aswell so the lists and the tasks can be stored in the database.

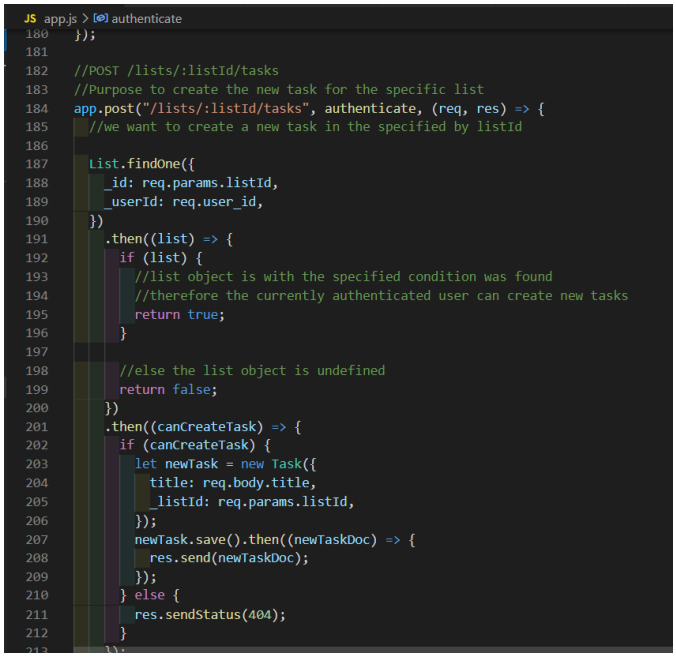
Code Snippet

Below is a screenshot of what the code looks like in Angular for the List Model



It is very similar to the User Model only less complicated. This handles the authentication of list based on the userId. You must export the task and list modules from the models into the index.js.

App.js is where we create our list, and find a list created by a particular user, create a task, based on the user logged in and who created a list and the task in that particular list, all of that is handles in the App.js please find below some screenshot of what App.js looks like on the website. The code is commented for your better understanding.



If you want to see the code and what it is doing in the front end just inspect the page open the console, you will see that the Token is getting refreshed and the user is not getting kicked out, they're still logged in and can continue with their work on the website.

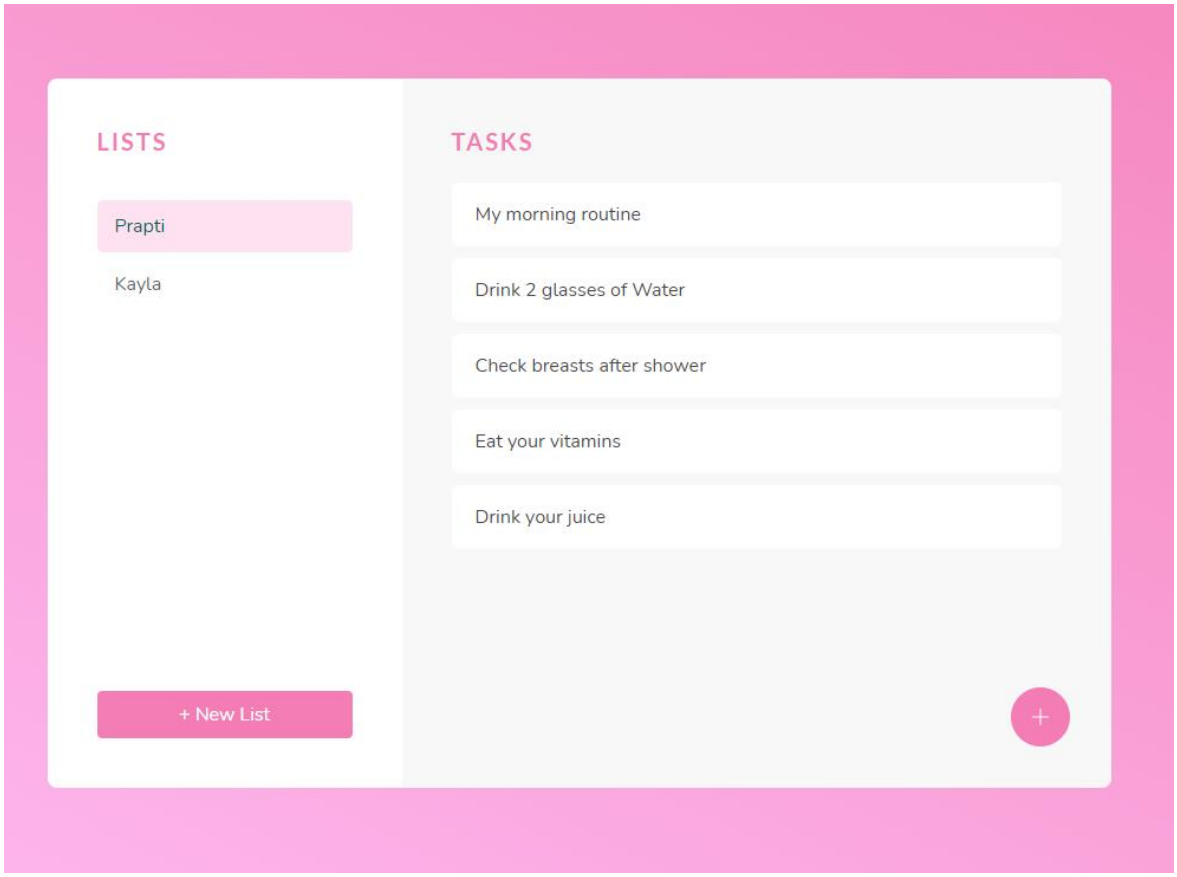
Comparison

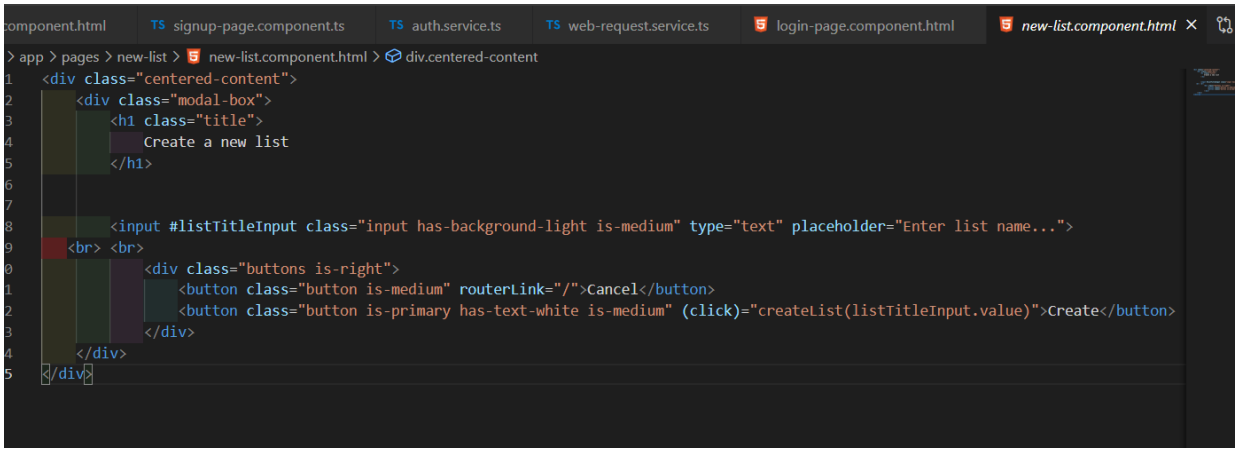
As both the websites are not fully doing what the requirement is, Angular comes very close to show the functionality that is required, same results can be achieved using Vue.js. As explained on the top we would have the create the list and task models and use them accordingly on the website. The steps for this process are mentioned above, creating a list and task was not a hard task in Angular as it was explained well in the tutorial how to do it, but I reckon if we use the same steps in Vue and follow the similar concept of login, and just add it to the List and Task based on userId this can be easily achieved. For this particular functionality I found both Angular and Vue pretty similar.

Interactive Engaging Elements

Angular

We have the lists and the tasks on the website which are some of the interactive elements on the website. A user can create a list and a task inside the list once logged in. The page is styled using CSS. There will also be a booking system on the website which will help a logged in user to redeem a coupon and book a free appointment for checkup. The coupon will be displayed on the website. The interactive element on the website so far is the sign up, login, creating a list and creating a task. In the future we would also allow a user to delete a list or update a list using the front end. This functionality is available using Postman for now, but soon we will have it on the front end aswell. So, the user will never have to use the backend for anything. Anyway in a website we should never give a user access to the backend unless they're the Admin for the website.


LISTS
Prapti
Kayla
+ New List
TASKS
My morning routine
Drink 2 glasses of Water
Check breasts after shower
Eat your vitamins
Drink your juice

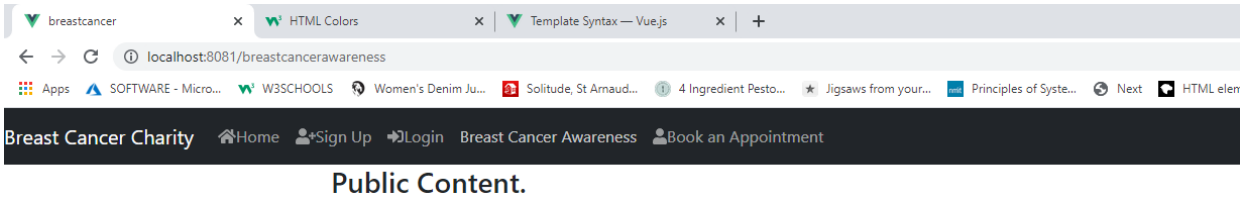

component.html
signup-page.component.ts
auth.service.ts
web-request.service.ts
login-page.component.html
new-list.component.html
app > pages > new-list > new-list.component.html > div.centered-content

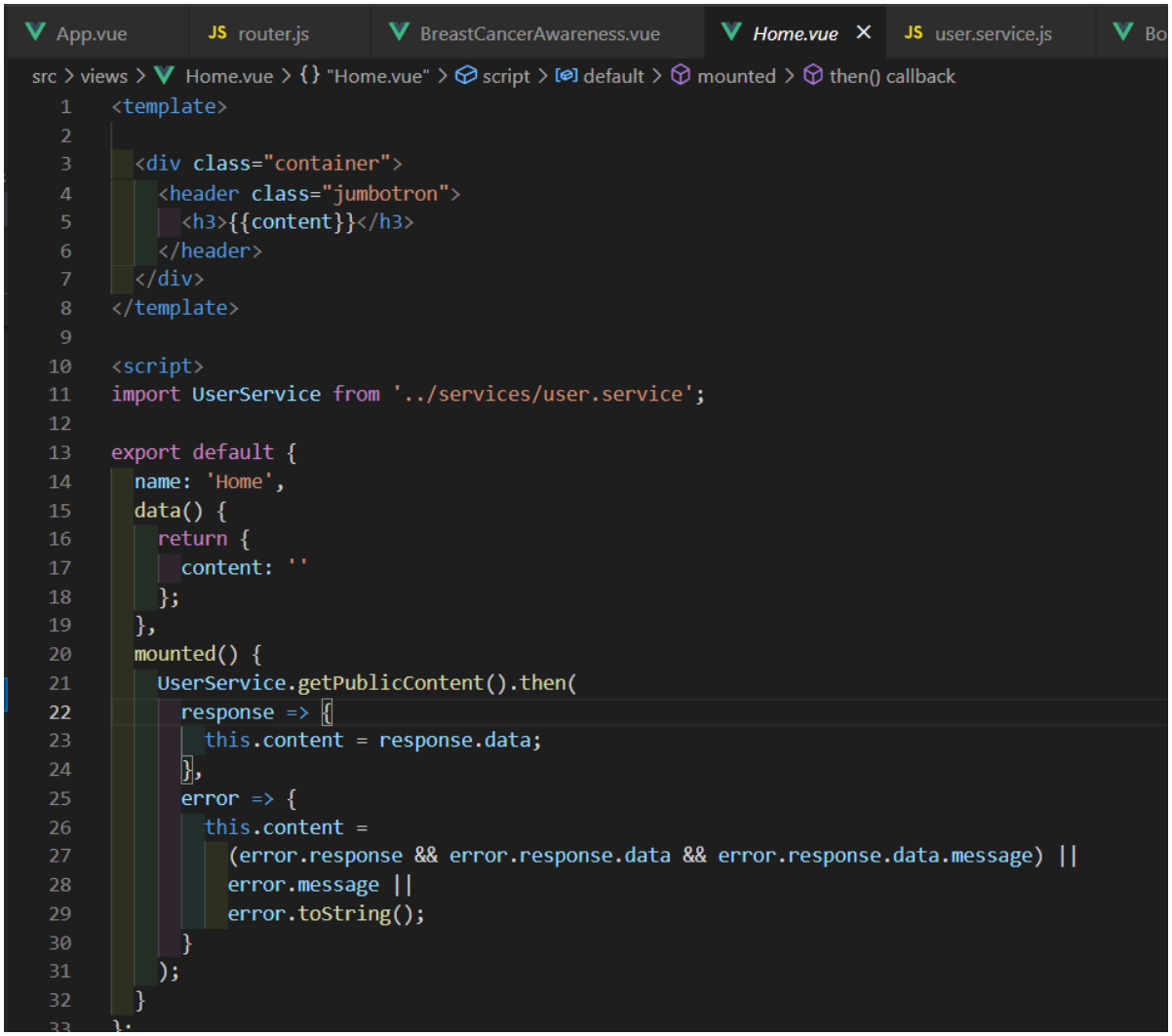
<h1 class="title">
Create a new list
</h1>
<input #listTitleInput class="input has-background-light is-medium" type="text" placeholder="Enter list name...">

<button class="button is-medium" routerLink="/">Cancel</button>
<button class="button is-primary has-text-white is-medium" (click)="createList(listTitleInput.value)">Create</button>

VueJS

The interactive elements in our view app are the sign up and the login, I have also set up two more pages which is the Breast Cancer Awareness and the Book an Appointment, which you can see in the Navigation Bar, but as of now these two new pages have no content in it. The Breast Cancer Awareness page will contain public content as it is the page that should be available for all types of account holders. The Book and Appointment page will be accessible only to the users who are logged in.





One of the problems I am facing in Vue is figuring out how to populate the pages.

Comparison

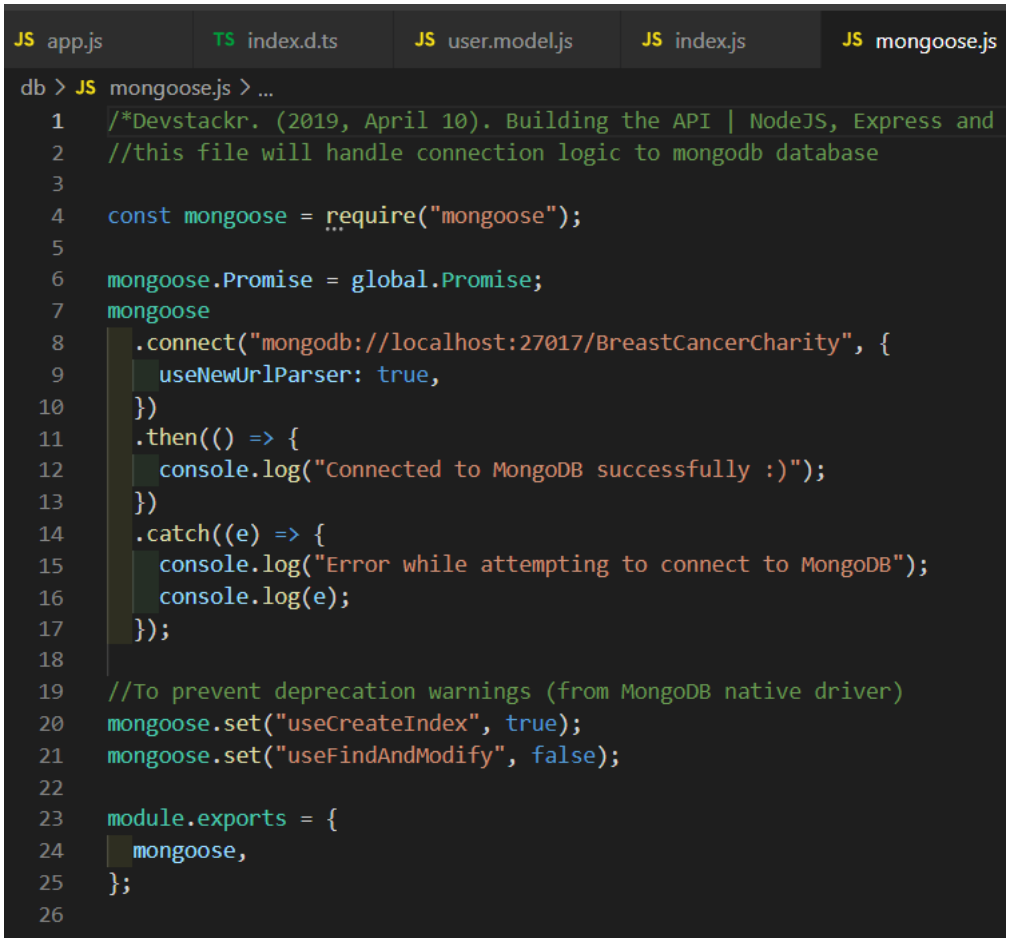
In angular populating pages is a very straight forward concept, you have an html page for your components, in which you write your html code with classes, and you style those classes in your CSS file, you can have one global css file for the entire website, so you do not have to search every time and all the styling is available in one place. But in Vue, that is not the case, there are no HTML pages and when I write the HTML code in the vue files, it gives me an error. This is something I still need to figure out how to do, this can be easily figured out and

should not be a problem for the website. So far in the Vue code we haven't used any CSS, hence the app is not at all appealing to the eyes. But this can be controlled by creating a CSS global file and turning all components into classes, similar to what is done in Angular. For the front end I did enjoy working on Angular, as their documentation was precise and well explained. Vue also has a good documentation, but it is more focused on the backend then it is on the front end and I didn't find it as helpful as the angular one.

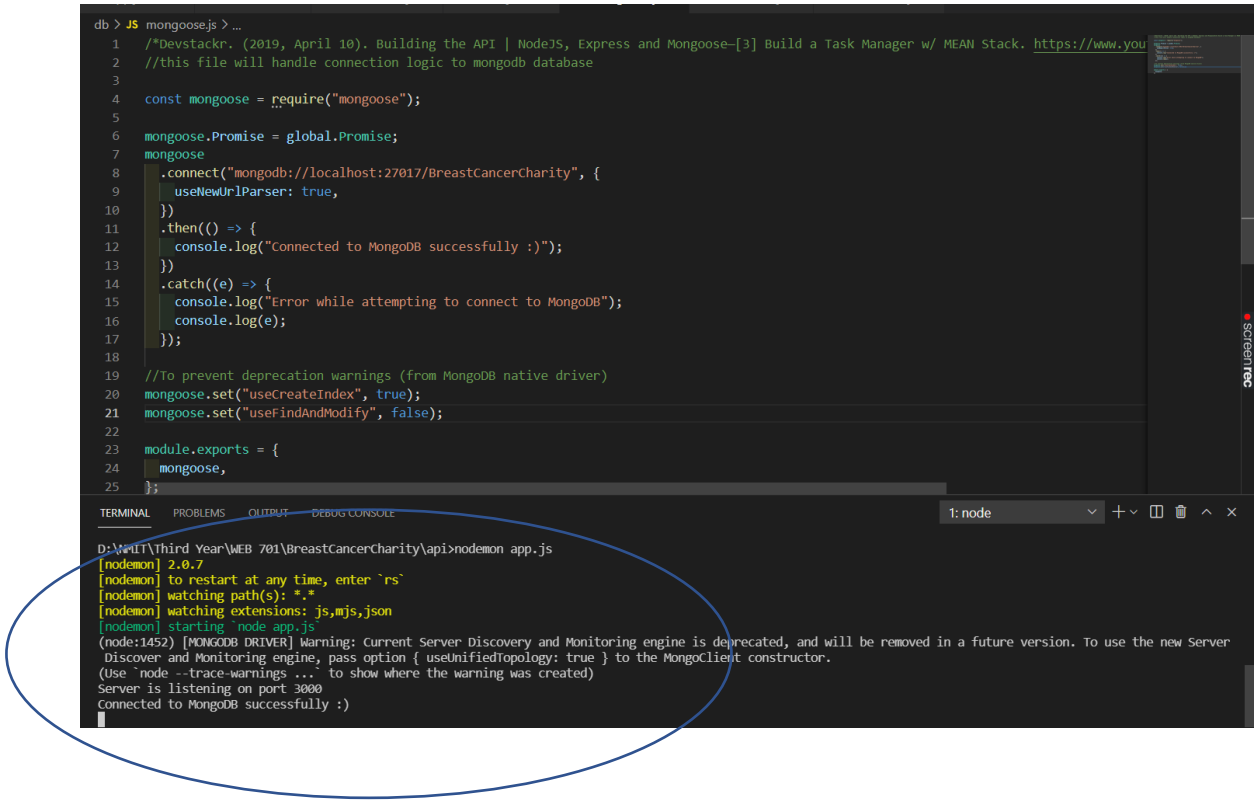
Store and Retrieve Data from Server-Side Database

Angular

We have created our database using MongoDb, to make sure that your application is connected to the Database and working fine, first you need to install Mongoose in Angular. The reason we do this as it helps to create a more secure application. MongoDb stores data in documents that can be retrieved as JSON objects. You have to make sure you have Express API set up for your project and then install mongoDB.



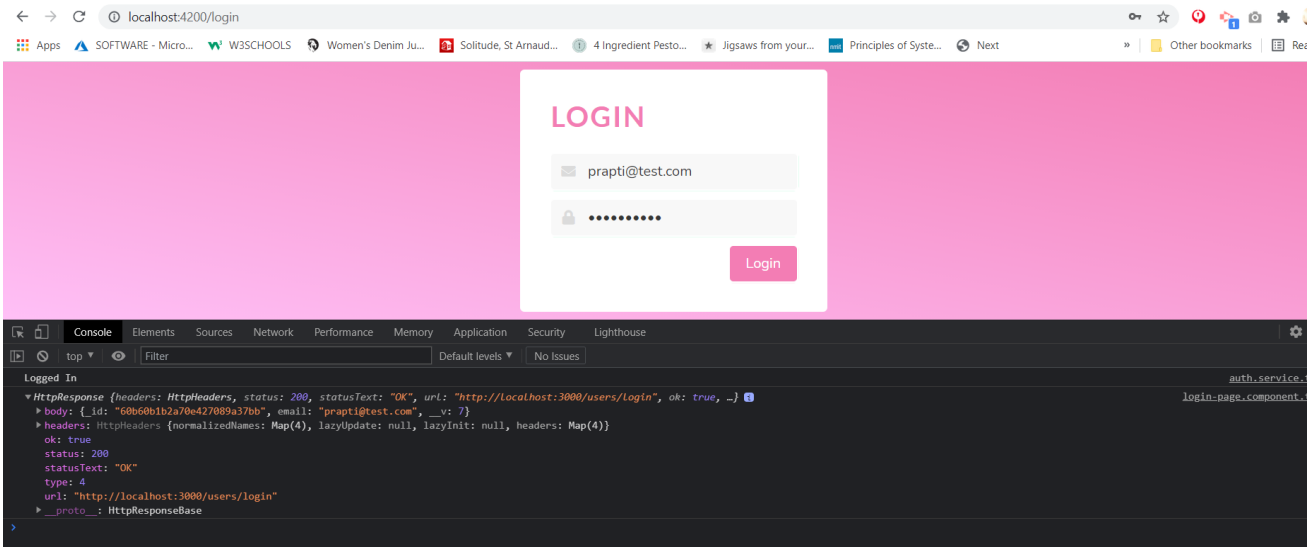
This is the code we have in our mongoose.js file, this code handles the connection logic to mongoDb. If you have everything set up correctly you will see the message “connected to MongoDb successfully” in the terminal, the command we use to start our backend is “nodemon app.js”. Please see the screenshot below to see what it looks like when in the terminal. We have to make sure we have node installed in the application. As you can see we’re running it on a different port as compared to VueJS we’re running it on port 3000.

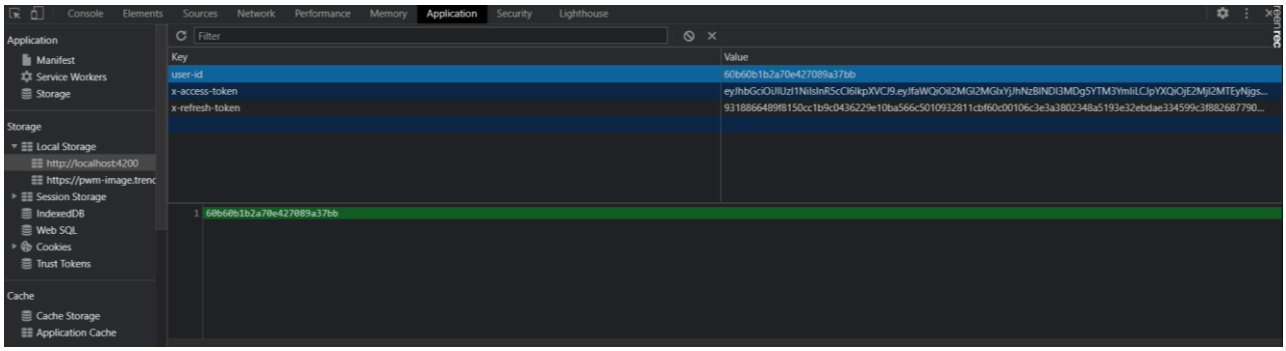


The above screen shows how we have successfully connected to the Database before you run the Front end of the application always make sure your API is running first. We are using our database to store the users and the lists and the tasks in the lists.

JSON Web Tokens, or JWTs, are a popular replacement for cookies. For API call authentications, they are a mechanism to keep user session data in Local Storage. Because of the many supporting libraries for browser and server-side handling of JWTs, this has become more popular than keeping cookies in Local Storage. The data is called using the Http Request using the routes and then that gets sent to the JWT Authentication middleware's and controllers and then the MongoDb database, then the MongoDb gives a response back, and it is then sent to controllers and using Express it the response is sent to Http. The process on how the database works is the same for angular and VueJs.

Below are the screenshots of the frontend showing that the data is getting stored in the database and being called when required, both for the users aswell as the lists.





VueJS

In VueJs we first initialize the node.js app with a package json file. Then we must install the modules which are, express, body-parser, mongoose, jsonwebtoken and bcrypt.js. Once all the modules and packages are correctly installed you can see it in the package json file under dependencies below is a screenshot of what the parameters set for database looks like

JS user.controller.js JS *db.config.js* ×

app > config > JS db.config.js > ...

```
1   module.exports = {
2     HOST: "localhost",
3     PORT: 27017,
4     DB: "breastcancercharity_db",
5   };
6
```

Then we create Models for User and Roles which we will be storing in the database, hence these can also be called the Mongoose models.

Below screenshot will show you how to initialize Mongoose

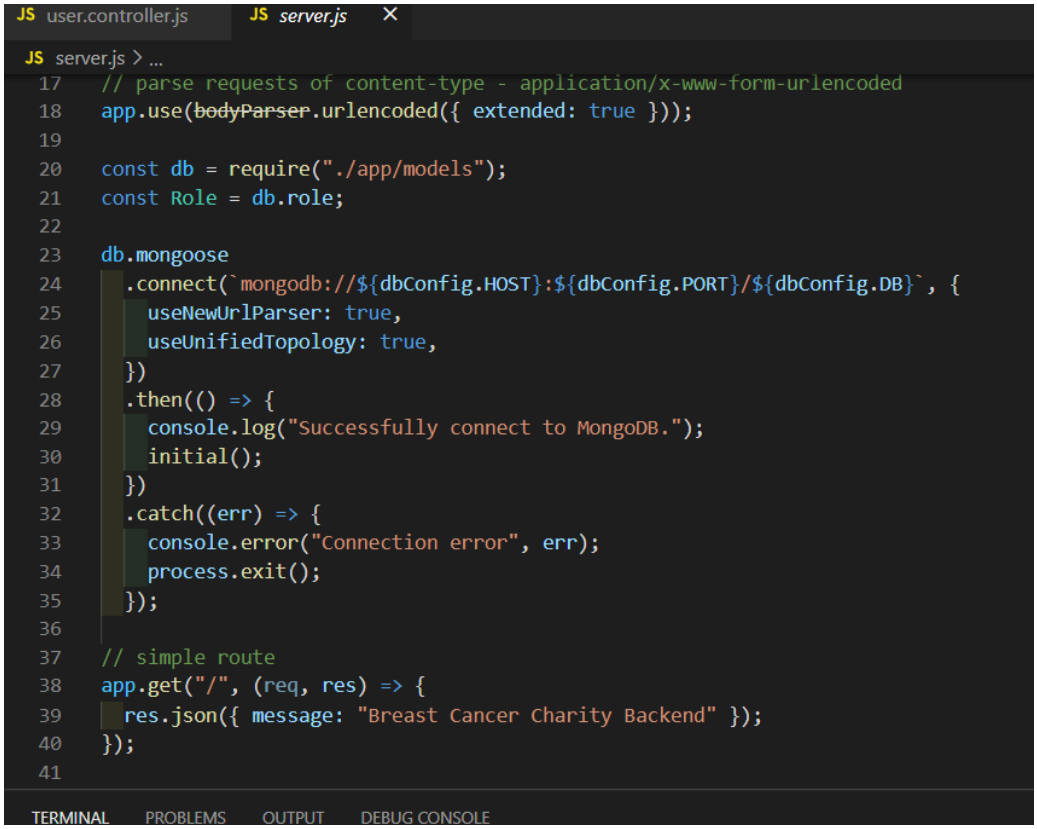
user.controller.js JS *index.js* ×

p > models > JS index.js > ...

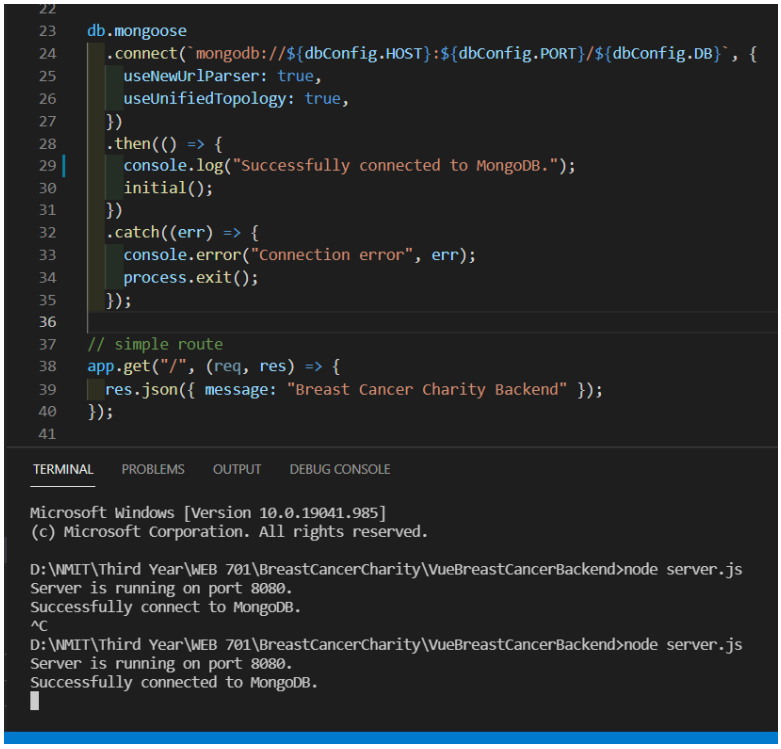
```
1   const mongoose = require("mongoose");
2   mongoose.Promise = global.Promise;
3
4   const db = {};
5
6   db.mongoose = mongoose;
7
8   db.user = require("./user.model");
9   db.role = require("./role.model");
10
11  db.ROLES = ["user", "admin", "moderator"];
12
13  module.exports = db;
14
```

Below code will show you how we create a connection between Mongoose and MongoDb database.

This code looks very similar or almost identical to what we have in Angular, that's because certain things which aren't complicated have been kept the same in VueJs

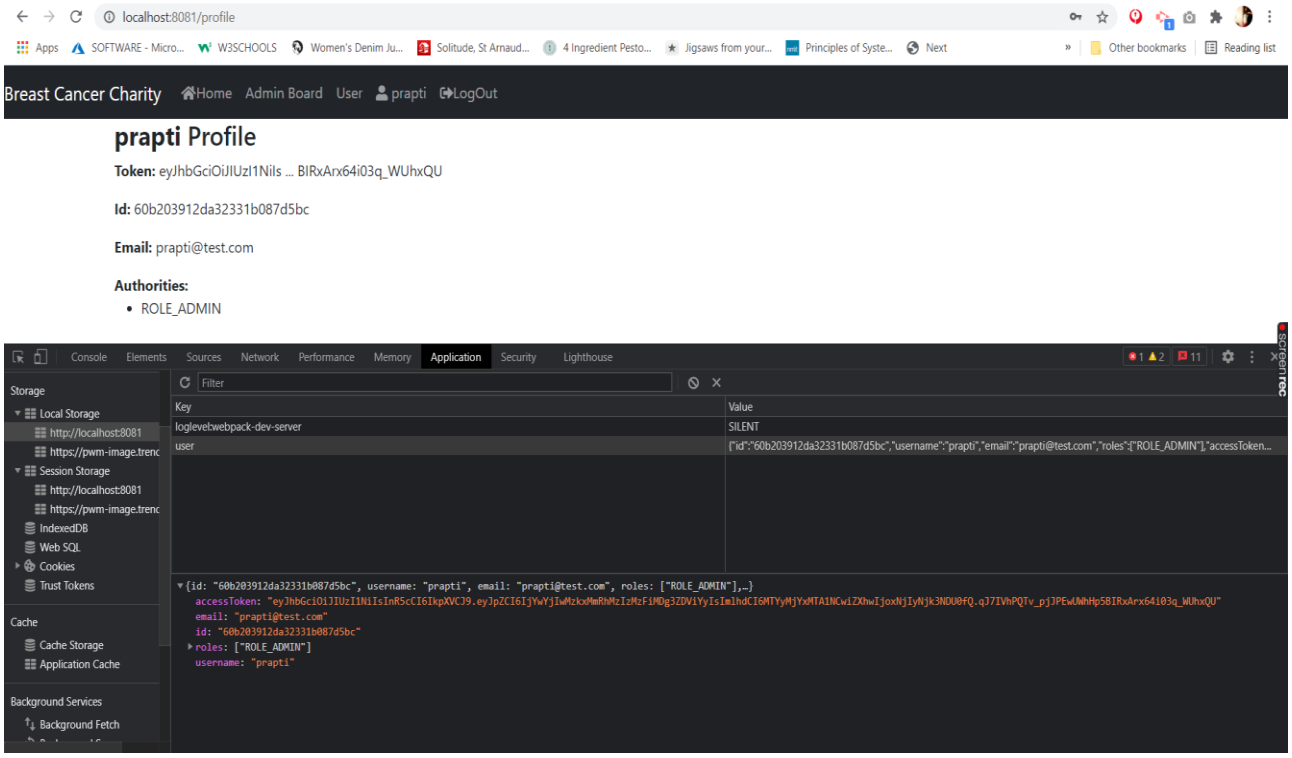


Now if I run the above code using the command "node server.js" this helps the application to run on the port which was mentioned before which asks the application to run it on the 8080 port, lets run this and see the screenshot of what the terminal looks like.



As you can see the mongoDb is running successfully which means the API is connected to the database successfully using Mongoose. If we decide to add more roles or users the database will keep on adding it, it is best to use Mongoose for a functionality like this but if you intend on keeping sensitive information stored on the database, I recommend we use MySql as it is even more secure.

Below is a screenshot showing that the user is stored in the database, if you inspect the page and go to Application you should be able to see the local storage, and that the user is stored in the database, when they first registered.



Comparisons

The API for connecting to database and calling data and sending a response to http works in the same manner in both these frameworks. They're very similar and most parts of the code to achieve the same results are almost identical too. Yes I still haven't been able to figure out how does making a list works in VueJS but I reckon the process is the same as what it is in Angular. In both the frameworks request, respond and next is used.

Recommendation of a Framework

After working on both the frameworks and trying to achieve the same results, I have realized that both the frameworks are very similar and yet very different. I

have worked on both the Client and Server side and have come to realize the framework that we are recommending for this website is the Angular Framework. As we have achieved most of the backend functionalities and now need to focus on the front-end of the website, I reckon it will be easier to work in Angular. We are recommending the MEAN stack approach for this website.

M – MongoDB

E – Express (back-end middleware)

A – Angular (front-end framework)

N – Node.js (run time environment)

We already have these modules set up for our project.

Some of the reasons why we prefer Angular as a framework are :-

Angular is built in the MVC architecture which is the Model – View – Controller architecture. The framework has Model and View synchronized so if the model changes so does the view. Because it has two-way data binding the coding time for developers has significantly dropped. This will result us in not writing unwanted code, which will help us save time and money.

Developers may apply special behaviors to the Document Object Model (DOM) via directives, allowing them to generate dynamic and rich content with HTML.

Dependencies describe how various pieces of code interact with one another and how changes in one component affect the others. Dependencies are usually defined explicitly in the components themselves. As a result, every change in dependency necessitates a change in components. You might utilize injectors in AngularJS to define dependencies as external elements, which would decouple

components from their dependencies. Components become more reusable, manageable, and testable thanks to dependency injection.

Angular has a huge community since it has been around longer than VueJs, because it is widely popular there is more documentation available for Angular also there is enough training material to solve any problems which arise during the development time of the website. Personally, working on both the frameworks I found Angular to be straightforward. Some bits were complicated and hard to understand but once you get the hang of the framework, it does become slightly intuitive.

Angular components are small user interface elements, similar to sections of an application. In Angular, there is a tight hierarchy of components, even though each component is contained with its own functionality. The components are well encapsulated for reusability. Encapsulation is also helpful for new developers to understand the code faster and better. Unit testing and maintenance of the code is much easier in Angular and efficient too. Dependency injection also helps with high performance of the website. (altexsoft)

Prapti Mane

Assignment - 3

Implementation of the Web Technology

Introduction to Angular

Angular 8 is a typescript based full-stack web framework which can be used for building mobile and web applications. One of the best advantages of angular 8 is that the application fits in all screen resolutions. Angular 8 is used to create dynamic web applications. The reason we chose Angular was because we wanted to change the data/information of the web application based on the user-to-user. As mentioned above the admin can see the booking requests made by the users, but the users will not be able to see those bookings.

Implementation of Angular

We have used MEAN stack for the development of our web application.

To start the project the first thing we did is created two folders “frontend” and “api”. Then we created the frontend application using the command “ng new frontend –style-scss –routing-true”. Then we installed Bulma for the front end CSS work, this is a free open source css framework. The command used to save Bulma was “npm install bulma –save”. The global styling of the app is done in the main-style.scss. The defining of routes is done through creating an object and we set the path to route which is a blank string and we set the components for eg: TaskViewComponent, which has to be imported at the top of the app-routing.module.ts. When we tested this our front end was working and we were able to see some content on the web page. The html elements are written in the html file for the specific components. In the implementation I will be talking about the task view component, to give you the basic idea and then the rest of the components are implemented in the same manner.

First let me give you a list of all the components that we have in our application:

1) Admin Dashboard Component
2) Sign Up Page Component
3) Login Page Component
4) Task View Component
5) New List Component

6) Edit List Component
7) New Task Component
8) Edit Task Component

Once we had a working frontend, we decided to set up our API. Open another code editor and navigate into the “api” folder, create a app.js file in the app. Then launch the terminal and install express using “npm install express –save”. Once it is installed, we loaded it into our app.js. We then pointed it to listen to the port 3000. While we were developing the application most of the backend testing was done using postman, since we did not have a database set up in the initial stage of development. We used our app.js for Get, Post, Patch, Delete, Save for all our functionalities in the application eg: register, login, lists, tasks, jwt tokens etc. Then we created our database first we created a mongoose.js file in the app which handles connection logic to mongo db database. Then we went ahead and installed mongoose by using the command “npm install mongoose – save”. Then we use it in the mongoose.js and connect it to the mongoDb’s default port which is port 27017/(nameofdatabase). Then we created a model’s folder and created our Mongoose models, we have the user model, task model, list model, booking model, role model. The models are used for defining the fields in the schema. These models are specific to mongoose. The index.js file is used to combine all the models so that its easier to import them than the other files and then we load these models in the app.js. We used postman to check all our requests. We then created a Task Service which helps modify our data. Then we created a web request service which handles the web request logic we connected it to the port 3000. Then we went ahead and connected the frontend to the api. Once we had one component working it was easier to have the rest of the components work. Angular is a bit more complicated than Vue, but it is interesting to see how it handles certain things. We are also using mongoDb Compass to maintain our database.

For the initial development stage of the application, I used an Angular tutorial to build my Task Manager, and then added functionalities on top of it after understanding how Angular works.

Problems faced during Development.

One of the problems that I am still facing in my application is that when we are using ngFor in the same manner as the TaskViewComponent.html in the AdminDashboard.html, for some reason the list of the bookings is not reflecting on the page. I keep getting the binding error, which I was not able to find the solution for online. Another problem that I am facing is the logout button is not working, even after using the right solution when I am calling it in the LogOut button.

Impact that you foresee this technology having on internet users.

This technology is easy to use, angular applications are built using the Typescript language, which is a superscript of JavaScript, which ensures higher security as it supports types (primitives, interfaces etc). The flow of the application is very simple and self-explanatory, the website has the features that an internet user is looking for in a breast cancer charity. The website allows the user to create a user profile and book a free appointment by filling in the contact form, the website even allows the user to use the task manager to keep track of their health checks. The user data is safe on the web app. An admin can sign up as an admin user on the website and then they get to see the Admin Dashboard option in the navigation, the user profile does not have that option in the navbar as only the account has access to the admin dashboard.

Links to Journal

https://praptimaneweb701.wordpress.com/2021/06/26/mean-stack/

https://praptimaneweb701.wordpress.com/2021/06/26/ia-design-api-design/

https://praptimaneweb701.wordpress.com/2021/06/26/typescript-101/

https://praptimaneweb701.wordpress.com/2021/06/26/web-frameworks/

https://praptimaneweb701.wordpress.com/2021/06/26/vue/

https://praptimaneweb701.wordpress.com/2021/06/26/angular/

https://praptimaneweb701.wordpress.com/2021/06/26/working-on-assignment-2/

https://praptimaneweb701.wordpress.com/2021/06/26/assignment-2-progression/

https://praptimaneweb701.wordpress.com/2021/06/26/assessment-3-progress/

https://praptimaneweb701.wordpress.com/2021/06/26/assignment-3-progress-completion/

Bibliography

What is a Web Framework? (2017, July 28). *GoodFirms Glossary - GoodFirms*. https://www.goodfirms.co/glossary/web-framework/

What is a web framework? | CodingNomads. (n.d.). Retrieved May 30, 2021, from https://codingnomads.co/blog/what-is-a-web-framework/

Why do we use Web Frameworks? (2017, June 19). LispCast. https://lispcast.com/why-web-frameworks/

How useful are web application frameworks? & How do I know which framework would suit me? (2014, September 23). Cuelogic Technologies Pvt. Ltd. https://www.cuelogic.com/blog/how-useful-are-web-application-frameworks-how-do-i-know-which-framework-would-suit-me

Web framework. (2021). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Web_framework&oldid=1022782175

Web service. (2021). In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Web_service&oldid=1021433007

8 Proven Reasons You Need Angular for Your Next Development Project. (n.d.). Grazitti Interactive. Retrieved June 1, 2021, from https://www.grazitti.com/blog/8-proven-reasons-you-need-angular-for-your-next-development-project

The Good and the Bad of Vue.js Framework Programming. (n.d.). *AltexSoft*. Retrieved June 1, 2021, from https://www.altexsoft.com/blog/engineering/pros-and-cons-of-vue-js/

How to Connect Angular and MongoDB to Build a Secure App. (n.d.). Okta Developer. Retrieved June 2, 2021, from https://developer.okta.com/blog/2019/09/11/angular-mongodb

Managing Local Storage in Angular. (n.d.). Briebug Blog. Retrieved June 2, 2021, from https://blog.briebug.com/blog/managing-local-storage-in-angular

The Good and the Bad of Angular Development. (n.d.). *AltexSoft*. Retrieved June 2, 2021, from https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-angular-development/

Wireframe.cc | The go-to wireframing tool. (n.d.). Wireframe.Cc. Retrieved September 17, 2020, from https://wireframe.cc/

Devstackr. (2019, April 10). *Building the API | NodeJS, Express and Mongoose—[3] Build a Task Manager w/ MEAN Stack*. https://www.youtube.com/watch?v=P3R-8jj3S7U

Angular 8—Introduction—Tutorialspoint. (n.d.). Retrieved June 25, 2021, from https://www.tutorialspoint.com/angular8/angular8_introduction.htm

Angular 8 Introduction | What is Angular 8—Javatpoint. (n.d.). Www.Javatpoint.Com. Retrieved June 25, 2021, from https://www.javatpoint.com/angular-8-introduction