

# Rhys Morris Portfolio

# Design Process

## Initial Process:

- Inspiration sought from online resources - Dribbble, behance and online compilations of web developer portfolios.
- Initial process involved selecting a colour scheme, planning the layout and sections of the page, and which components would make up those sections.

# Design Process

Colour scheme :

- Blue theme across the site
- Stored as SASS variables for ability to change across site easily once the project started e.g. the background colour shade was darkened twice early into the project

Fonts:

- Selected and experimented with google fonts - landed on Roboto, Roboto Mono and Dancing Script

# Design Process

## Basic Layout Planning:

- I started working on my portfolio site ahead of the assessment release, so originally had the site planned as a scrollable landing page
- Planned sections:
  - Hero - animated, grab attention
  - About - photo of myself, basic info
  - Work - tech stack experience, project cards
  - Contact - location map and contact form
- Planned repeated components:
  - Navbar top of screen - hamburger at smaller screen sizes
  - Social links
  - Footer text - email and byline

# Navigation

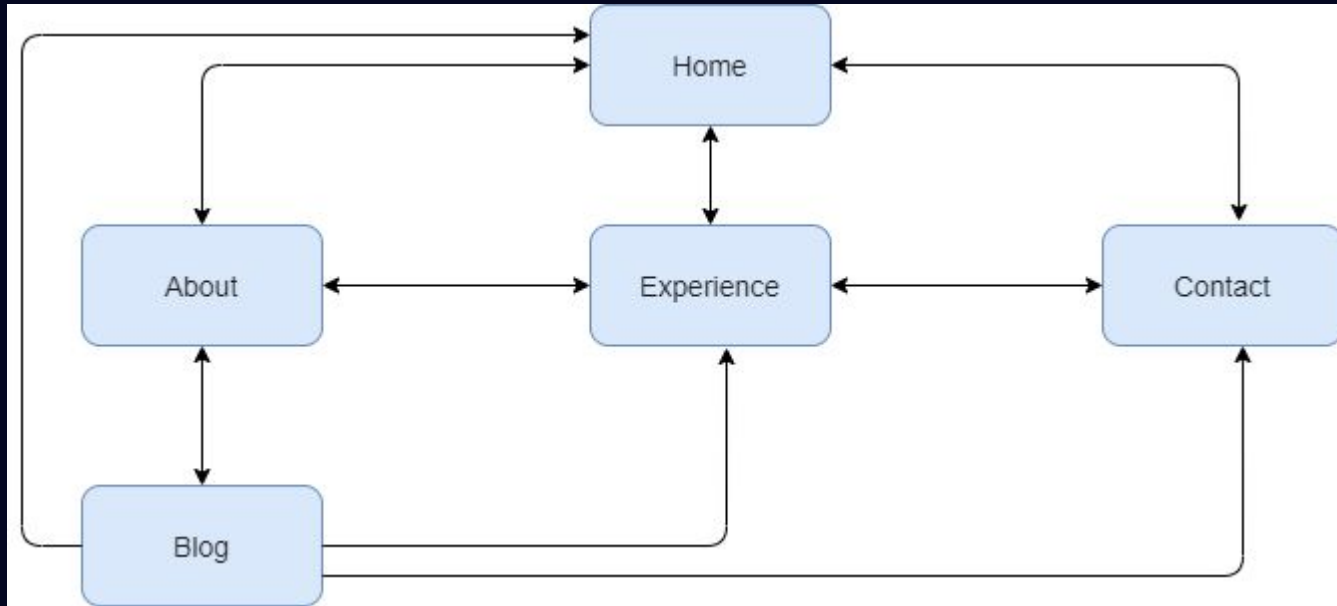
## Site Navigation:

- Navigation bar at the top of each sectional page allows easy inter-page navigation
- All pages bar the blog can be accessed from any other page
- Blog linked exclusively to the about section.
- Resume accessible across all pages as part of the navbar

## External Links:

- Social links on all pages as either a fixed sidebar or attached to the footer
  - Github - verify ability to program, view previous work
  - LinkedIn - view professional network, previous work history
  - Twitter - verify commitment to developer community and a documented learning progress

# Sitemap



# Responsive Design Decisions

Media Queries:

Rather than using set mobile, tablet, desktop media queries I opted to use CSS variables to define certain breakpoints where the page no longer looked as intended.

For example: \$breakpoint1: 1400px

.....

\$breakpoint8: 325px

I used a desktop first approach.

The use of flexbox and CSS grid made responsive design easier. It was mostly just tweaking layout orientations i.e. row, column and small changes to margins and padding for individual elements

# Responsive Design Decisions

Font:

I opted to use rem for font-size measurements which again made responsivity easier as screen width decreased. Rem refers to a font-size that is relative to the root font-size, which in most cases is the html element.

By setting:

```
html {  
  Font-size: 62.5%;  
}
```

The root font size is now set to 10px. This is because the default normal 100% figure is 16px. ( $16 * 0.625 = 10$ ). Now if I want to set a particular element to 14px I can set its font-size property to 1.4rem.



# Responsive Design Decisions

Font Cont'd:

The reason I chose to use rem over em is that if a user wishes to adjust the default size of the font in browser settings this change will be carried through if rem is used, this is not the case for em.

As the size of the screen decreased I could very easily reduce the size of the entirety of my text by adjusting the root font size. This prevents a lot of micro changes and excessive media queries.

Example:

```
html {  
    font-size: 50%      // 1rem is now 8px!  
}
```

# Code Structure & Build Process

All production CSS was written with SASS and then compiled into the build `style.css` file.

I used an npm script to initialise watching `.scss` files for change and auto-compiling to the build `.css` file.

`.scss` files were broken into appropriately named variables to improve code readability. For example: `_variables.scss` `_nav.scss` `_hero.scss`  
These partials were then imported into a single `style.scss` file that was ultimately compiled into CSS.

I used the Block Element Modifier methodology for naming CSS classes. This made nesting SASS much easier to write.

Example: Hero component (`div .hero`) - header in hero component (`.hero__header`)

# Code Structure & Build Process

In .scss file:

```
.hero {  
  &__header {  
    // Stuff  
  }  
}
```

I made further use of npm scripts to autoprefix code for cross-browser support. This required the use of npm packages postCSS-cli and autoprefixer. Finally the compiled CSS style.css was minified and compressed using a built in node-sass feature.

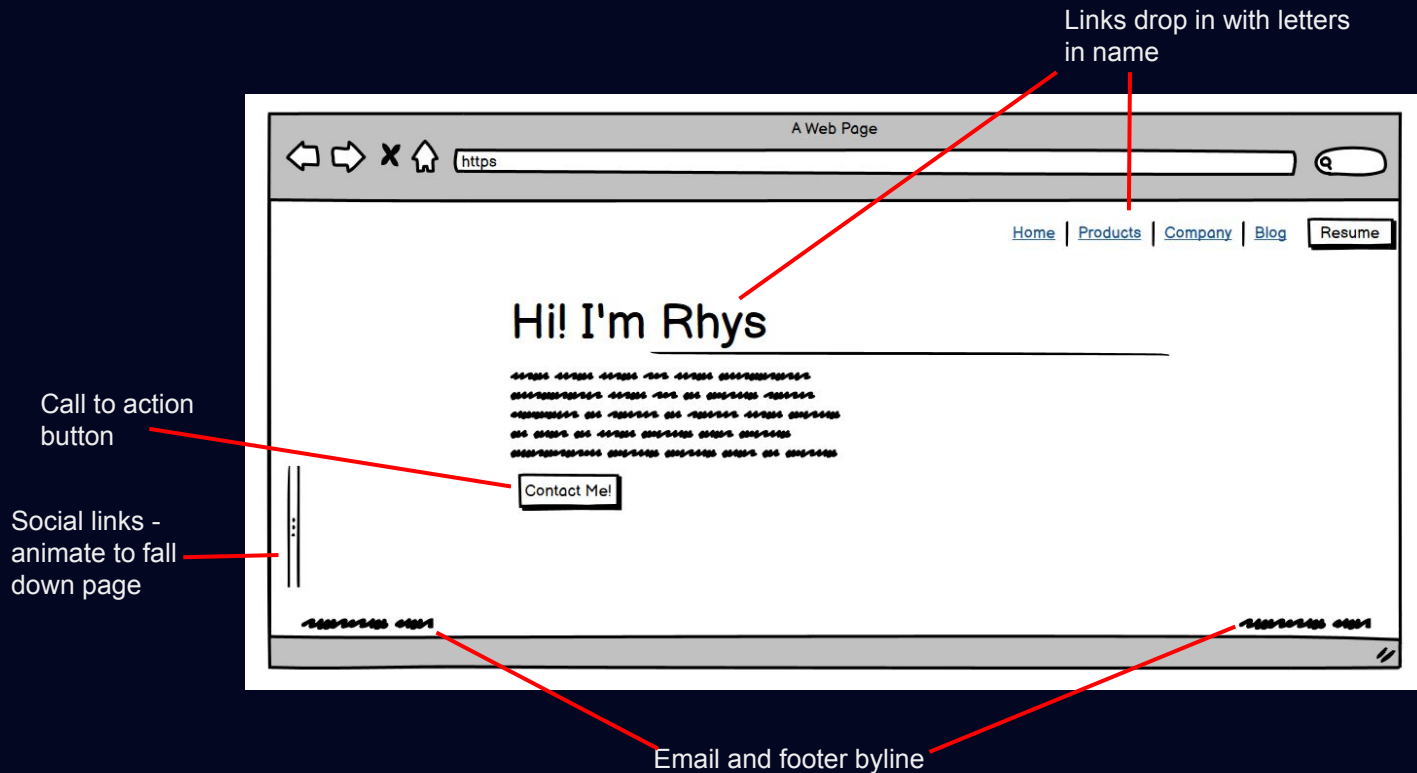
Whilst these steps weren't required for the project brief, I wanted to use the opportunity to make deployment as real world as possible.

# Wireframing

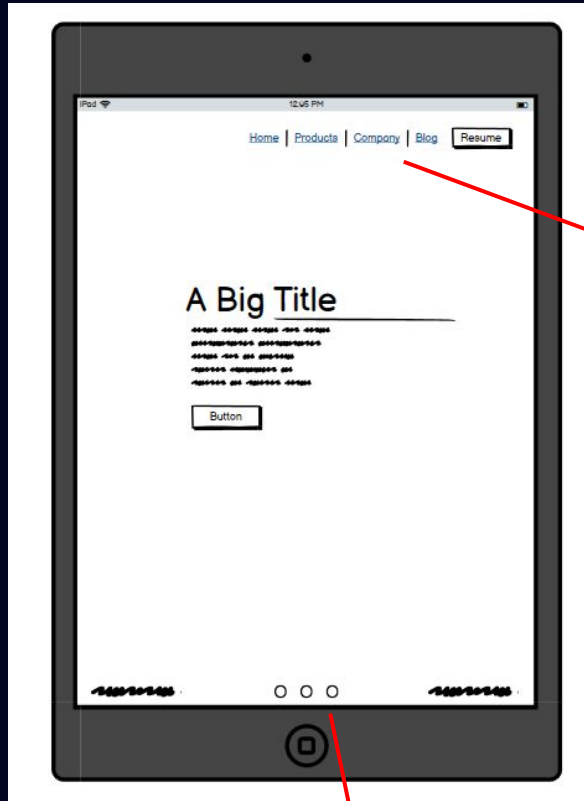
## Process:

- The majority of my early concept wireframing was done with pen and paper, then transferred to Balsamiq
- Smaller components e.g. the layout of the project cards were wireframed by hand with the understanding they'd likely change during the build process

# Hero Page - Desktop



# Hero Page - Ipad



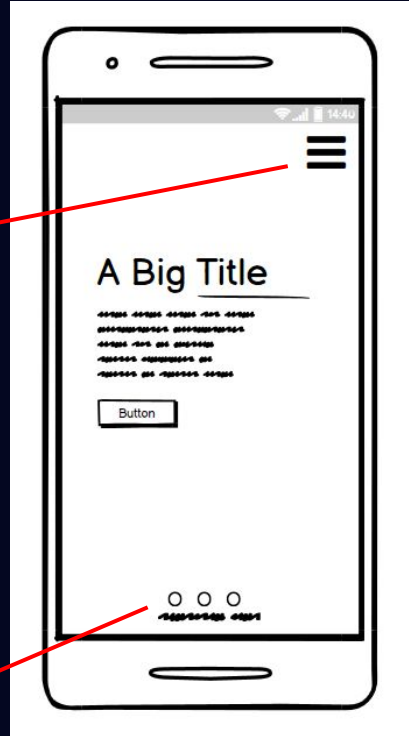
Switch justify-content to space between when links start to cramp to spread across to of page

Social links move to footer

# Hero Page - Mobile

Hamburger nav on  
mobiles

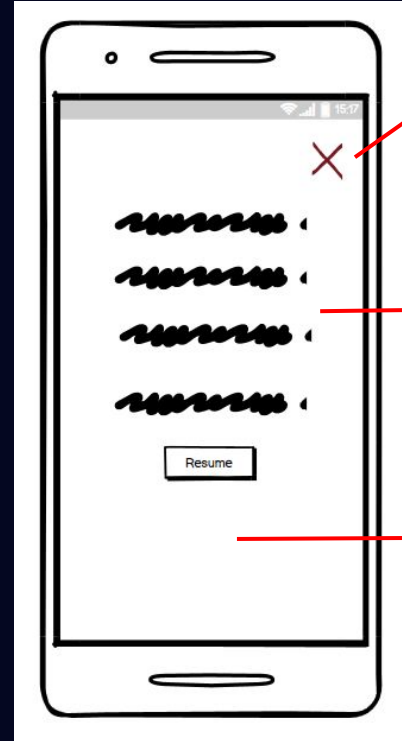
Footer column  
oriented



Hamburger icon  
rotates on touch to X

Links drop down from  
above when  
hamburger touched

Background is  
opacified. If the  
background is  
touched while the  
hamburger menu is  
open - the menu  
closes



# Hero Page

## Function:

To greet the user and direct them to further navigation of the site

## Section Components:

### Hero text:

- H1 element - I'm Rhys.
  - Rhys letters drop in to draw attention and utilise the primary colour for the site for visual emphasis
- H2 element - web developer
  - Developer text uses Dancing Script font for visual interest. Underline effect is achieved with padding and border-bottom - the aim is to draw the eye to the hero text
- P element
  - Basic information on myself and direction to call to action button below



# Hero Page

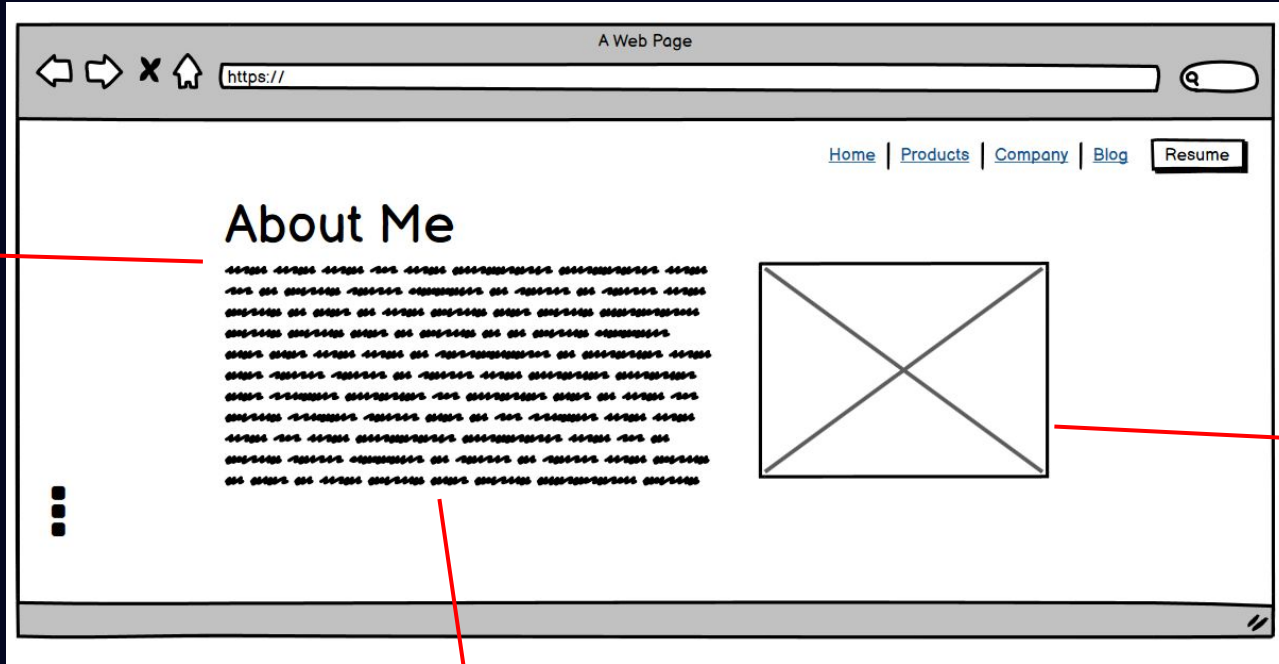
## Section Components:

- Call to action button
  - Immediately navigates to contact page and contact form - ultimate goal of the site is to attain work opportunities
- Ripple effect on click/touch
  - A span element is added to the body of the page at the position of a mouse click or touch. Span is animated and styled with CSS to be a circle with a visible border that grows and fades out, like a ripple in water. This component was designed purely for user interactivity and engagement.

## Future plans:

- Button to switch colour scheme from dark mode to light mode (invert the colour scheme)

# About Page - Desktop



Header - repeated styling throughout site

Socials

Navbar

Photo of myself with frame effect

About me text with links to outside pages

# About Page - Desktop

## Blog posts slideshow concept

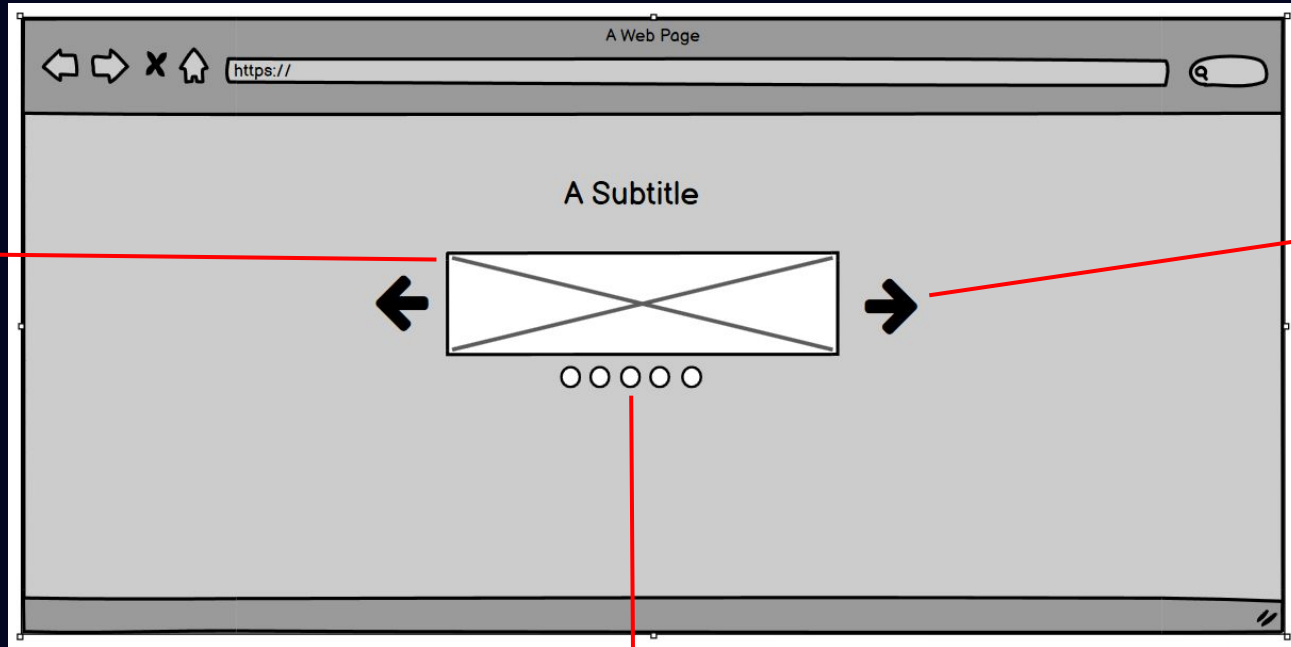
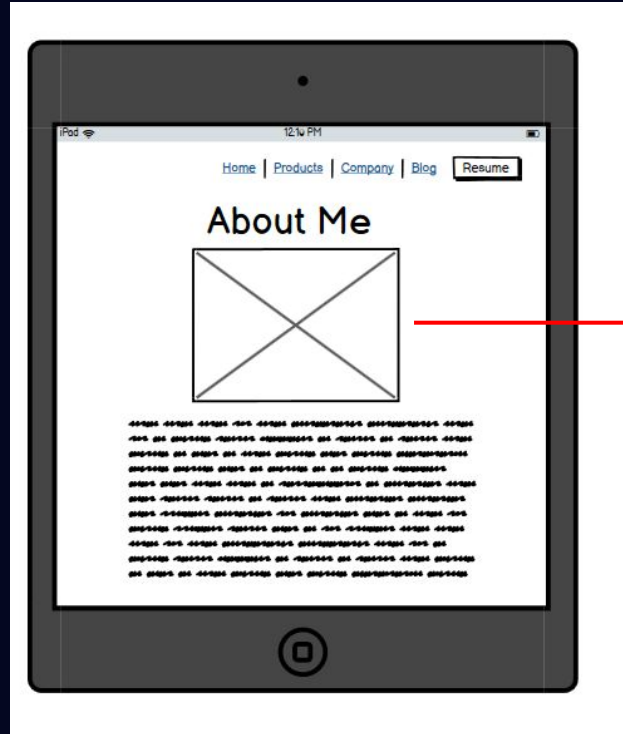


Image and text  
that links to a  
specific blog  
post - timed  
scrolling

Scroll R & L  
icons

Select specific post - updates based  
on current post

# About Page - Ipad



Column  
Orientation

# About Page - Mobile



Selector icons  
are removed on  
slideshow

# About Page

## Function:

Provide background information on myself, my previous employment/study and interests

## Section Components:

### H2 element

- Heading for the page - style is repeated across other pages

### P elements

- Background information and links to external resources
- Custom underline animations for links - easy for the user to identify that this is a link
- Color of links is primary color of page

# About Page

## Section Components:

### Photo of myself

- Greyscale which returns to colour when hovered for visual interest.
- Frame around the photo on desktop screens that also moves inwards.

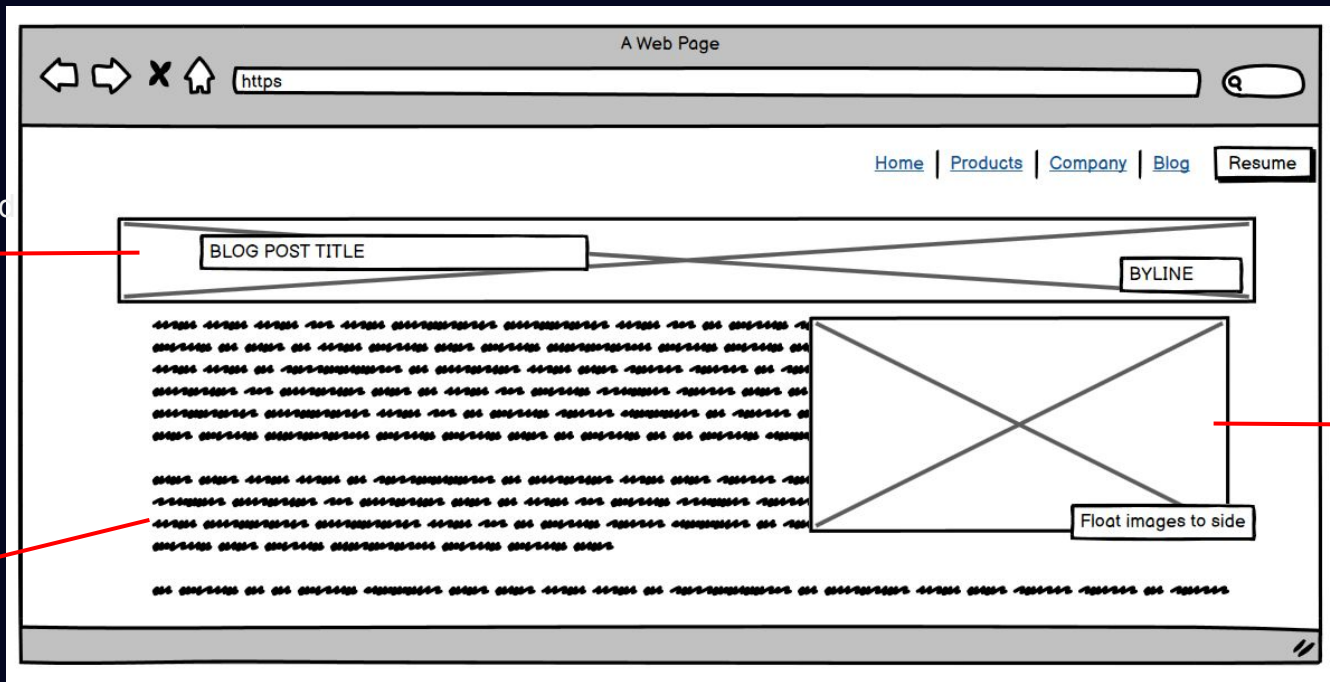
### Blog Slideshow

- Timed slideshow of 'recent' blog posts that user can interact with, and if clicked, will navigate to the associated post on the blog page.
- Mouse users can hover blog posts to have title appear and zoom effect. This is removed for touch screen users (<1200px breakpoint) and the title is static.

### Other

- Easter egg!
- Hint: Life, the universe and everything

# Blog Page - Desktop



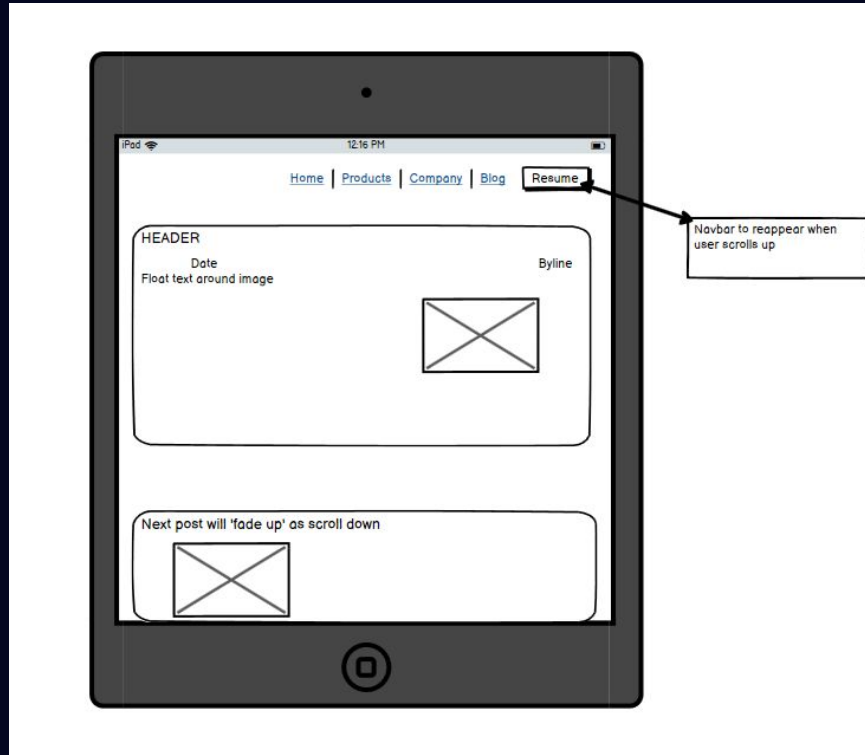
Navbar

Content  
photo floated  
to side -  
box-shadow  
to add depth

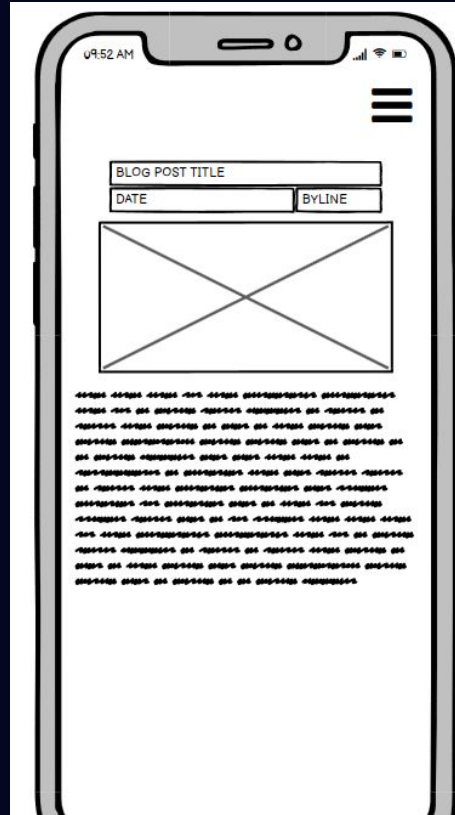
Scroll down for new posts to fade in



# Blog Page - Ipad



# Blog Page - Mobile



Take up full width of device  
with content padding

# Blog Page

## Function:

Page for developer blog posts

## Sectional Components:

### Blog Posts:

- A wrapper div element was used to place a top border only - an attempt to add some 3d perspective to the posts
- Blog posts appear by fading in as the page is scrolled down.

### Blog Post Header:

- Background image of header with a gradient overlay. Text can then be placed over the top of the background and still be readable.
- Uniform headers between blog posts
- Added letter spacing and reduced font weight to headers to make them feel 'softer' to look at

# Blog Page

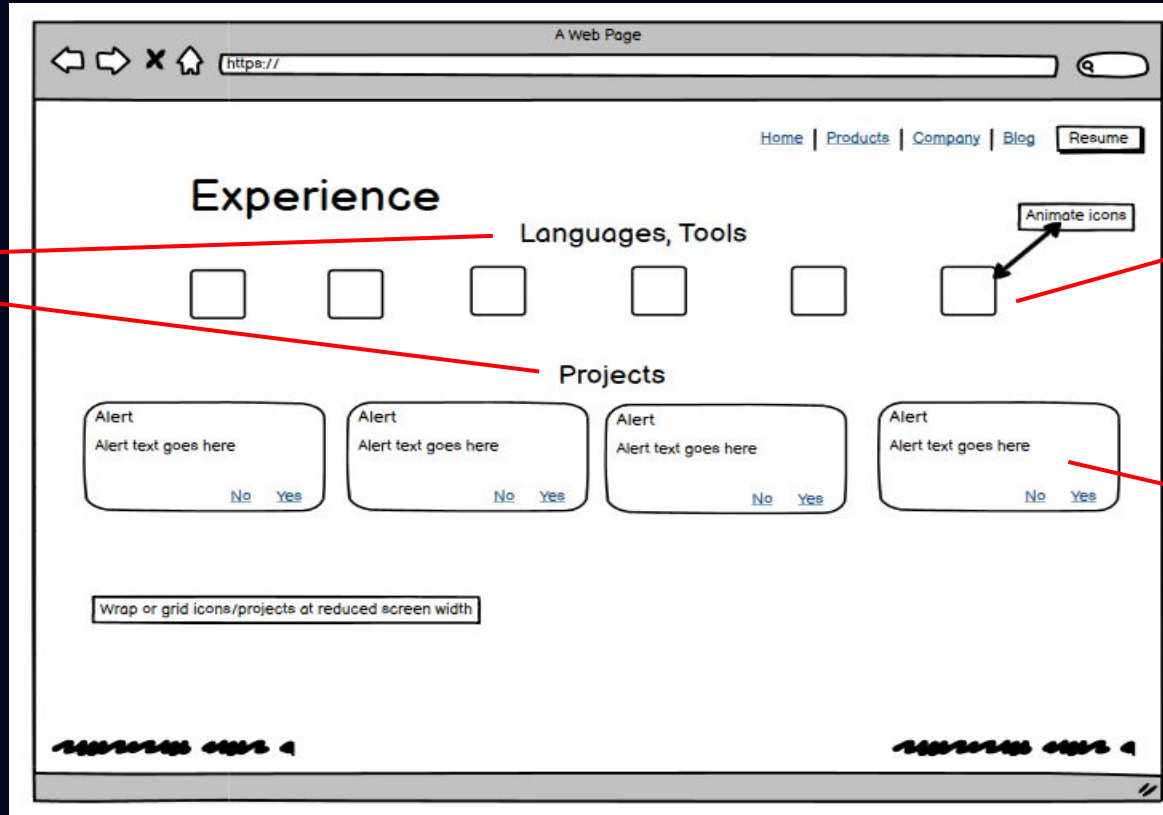
## Blog Post Content:

- At larger screen sizes blog posts are floated to the right and left with text wrapping. At smaller screen sizes the image moves above the text to improve readability of the text.
- Images given a box-shadow effect to provide 3d perspective

## Navbar:

- As this is a longer page I added the ability for the navbar to drop back in when the user scrolls the page upwards. If they scroll down again it will disappear.

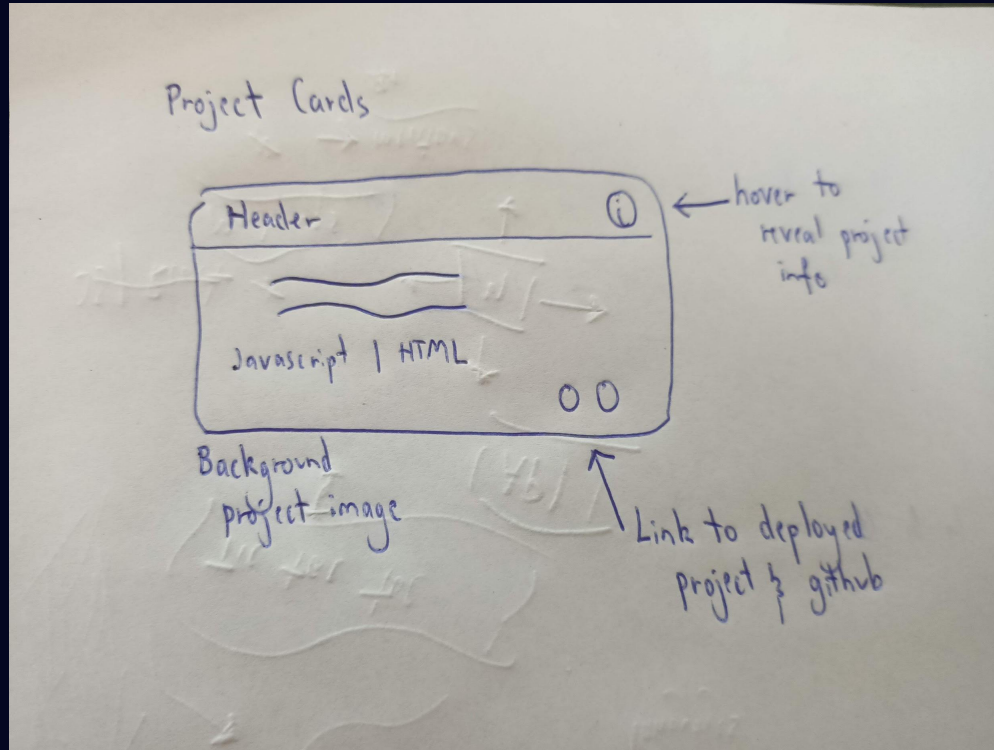
# Work Page - Desktop



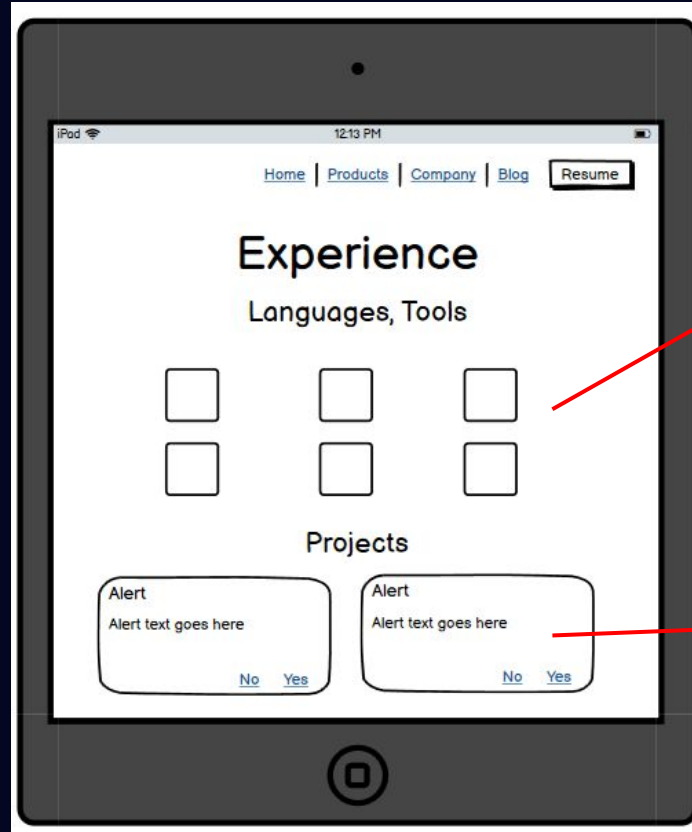
Icons representing which technologies I have proficiency with - animated on mouseover

Project cards

# Work Page - Project Cards



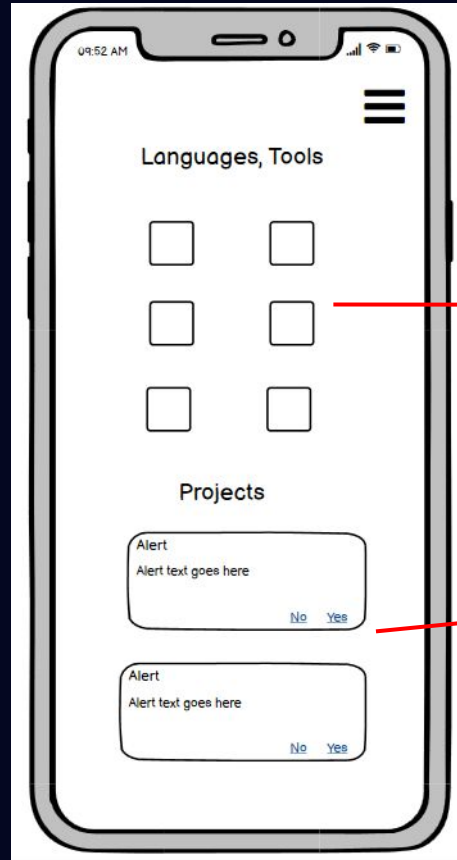
# Work Page - Ipad



Icons aligned  
with CSS Grid

Project cards  
responsive with  
flex-wrap

# Work Page - Mobiles



Continue to reduce # per line  
as screen width reduces

Project cards 1  
per line



# Work Page

## Function:

Demonstrate to a potential employer the technologies I have worked with and examples of my work

## Sectional Components:

### H2 element

- Heading for the page - style is repeated across other pages

### H3 elements

- Subheadings for languages, technology & projects

# Work Page

## Sectional Components:

### Icon Section:

- Demonstration of the technology I have working proficiency with - plan to add to this with time
- Custom animated icons on mouseover with Animate CSS library for user interactivity. For users on touch devices the effects will occur on touch instead
- Opted to use CSS grid for better control of icon placement for responsivity

# Work Page

## Sectional Components:

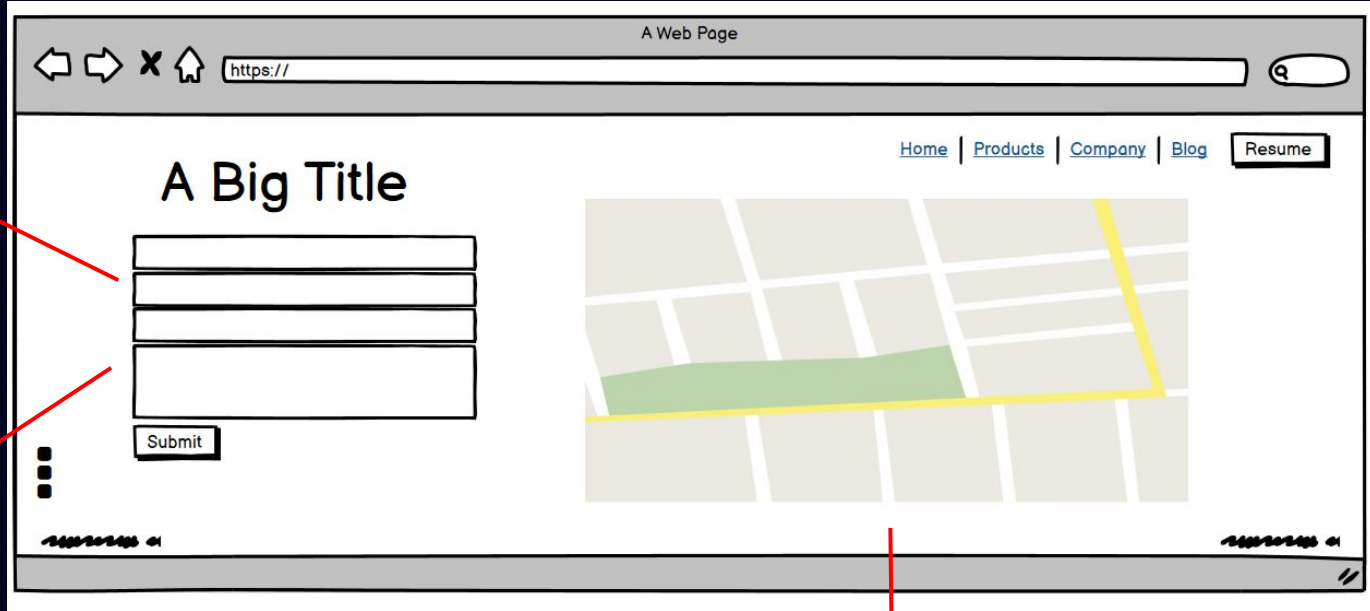
### Project Cards:

- Demonstration of deployed projects I have worked on
- Title and background image of the project itself
- I made the decision not to have information regarding the project overlying the image as it felt too cramped, instead I added an 'info' button that when hovered would grow to occupy the card space and reveal a brief blurb about the project and the technologies used
- Each project card has links to both the deployed project on Github pages, as well as the source code repository - this allows an interested party to view the work more closely if desired
- Plan to add/update project cards over time

# Contact Page - Desktop

Contact form  
- uses  
formspree to  
send

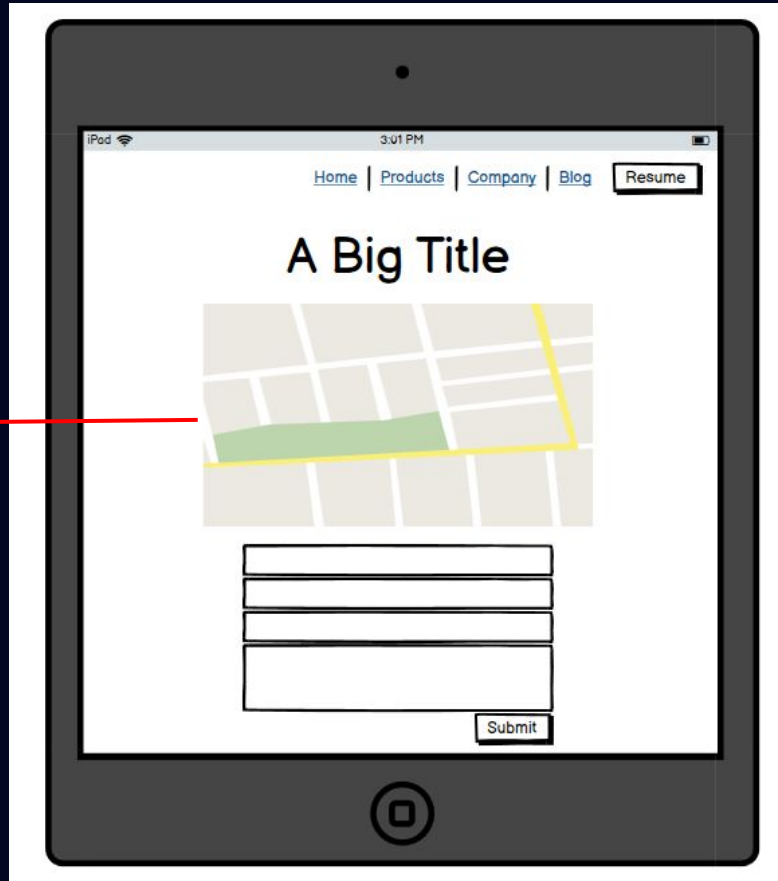
Form  
validation  
styling when  
send button  
is clicked



Location map - for mouse users blue filter  
is applied to match theme - removed on  
hover

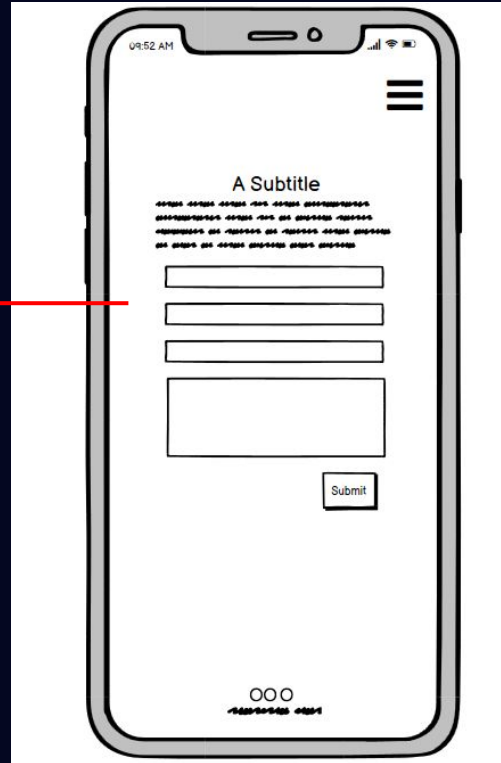
# Contact Page - Ipad

Move form  
under map



# Contact Page - Mobiles

Remove map on mobile devices - too cluttered



# Contact Page

Function:

Ability to contact myself regarding work opportunities, employment or just to get in touch!

Sectional components:

Contact form:

- Form labels are present in HTML so that they can be used for screen reader users, however, I have hidden the element using CSS on page to improve form aesthetics. Placeholder text informs user to the purpose of the input.
- Form is functional using Formspree
- Custom form validation styling that executes when the form is submitted, and is time lapsed to disappear after a default of 4s - this allows the user to easily identify which form inputs require changing without being too obnoxious
- Use of CSS pseudoclass `::focus` to add a bright primary colour box shadow around focused element - user can easily see where they are entering text

# Contact Page

## Location map:

- Inform user of my location - without giving a specific address
- For mouse users the map is turned light blue to match the overall color theme by using several CSS filters including greyscale, sepia, hue-rotate and saturate. The map reverts to a normal colour when hovered
- This effect is removed for touch screen users
- Map moves above the form into a column layout for tablet devices
- On mobile devices the map is removed entirely as it felt too cluttered in a column layout with the form underlying



# Repeated Components

## Hamburger Icon/Menu:

- Replaces navbar on devices with <600px screen width
- When clicked the icon rotates to form an X
- The links drop down from above to fill the page vertically
- The background is opacified using a gradient, the hamburger menu sits on top due to a higher z-index position
- If the screen is touched in a location that is not a link, the menu closes - this required the use of Javascript

# Repeated Components

## Social Links:

- Ability for user to navigate to my social platforms including Twitter, Github and LinkedIn
- Fixed as a sidebar for desktop users - drops into this position using CSS animation on page load. Use of custom cubic-bezier to get the 'bounce-back' effect
- Custom animation on hover to improve user interactivity/experience
- On smaller devices I opted to move the social bar into the footer, this is where the social bar has been marked up in the HTML as it made the most semantic sense
- Animations are removed when placed in the footer

# Repeated Components

## Footer:

- Contains my email - a further point of contact - as well as some byline text
- Layout changes based on screen size and presence of the social links
- On desktops the email and byline are separated to either side of the screen, I felt this looked best
- Transition 1: Socials added centrally, content remains justified to space between
- Transition 2: As screen width reduced, social links position is moved upwards using the relative property. Email and byline remain spaced between.
- Transition 3: Column format: Socials - Email - Byline
- I played around with these transitions based on differing screen width and this is what I felt looked best at certain breakpoints

# Challenges

- Design - still not 100% happy with my project card and will likely tweak the appearance in the future
- Responsivity
- When to consider a page 'finished' - time management
- Scroll in script for blog posts - had to play around with the code to get the scroll effect how I wanted it

# Favourite Parts

- Implementing user interactivity with CSS animations and Javascript
- Writing SASS
- Seeing my ideas come together