**Set-up**

I chose put the page together in Sass, whilst I love vanilla CSS, I thought Sass would give you a better idea of how I approach things in production. I used a fairly basic folder structure, some simple variables and mixins, again just to show how I tend to think in production. Sass was compiled easily using: sass --watch --style expanded css/scss/styles.scss:css/styles.css

I used Chrome exclusively for dev.

**Approach**

I built the page mobile-first, though paradoxically ran out of time to style the type on mobile. The first thing I did was to put together an un-styled semantic HTML skeleton (I’ve attached a grab of this) and then ran this through the W3C validator. When I moved to desktop I worked my way the down the page focussing on specific sections and elements using a mixture of XD and screenshot measuring (cmd shift 4 esc) to guide me but also with a view to checking layout at 1440px with overlays at some point - where I take a screenshot of what I’ve made in the browser and place this ontop of the Sketch file, reduce opacity and compare, simply to check everything is as it should be (example grab attached). I find this is a great way to see where adjustments might need to made and I believe feel this kind of attention to detail can really bring the design and site to life.

**Mark-up and classes**

As a bit of a purist I always want to try and use the cascade and inheritance as much as possible though at times have learned the hard way that, with more complex designs and sites, it can better to style a component by directly attaching specific classes to elements and styling these accordingly. Given that this page was very simple I went for a hybrid approach that I hope demonstrated mixing inheritance and classes. Ironically I didn’t use the only component specific class I’d added (.card) due to the simplicity of the page. I did try keep things as DRY as possible.

**Fonts and typography**

I chose to use Inter as the primary font, I think it’s quite modern and clean, but importantly is also great for legibility. Because of the differences between this and Gotham I was unable to match the typography as exactly as I would have liked and think the design suffers a little as a result. I couldn’t access letter-height or font-weights in XD for some reason so had to approximate these, opting for slightly tighter line-height on headings. I only used a single font weight in the end (400), lighter versions of Inter didn’t really seem to work that well, and I’m always mindful of performance. I would have used clamp() or my own css variable based method to apply fluid type to trickier in-between viewport sizes but I ran out of time unfortunately.

**Units**

I used rem to set type and apply padding in certain instances, though where fluidity was required I opted for the humble percent. I tend to use pixels when I need a fixed unit, for example on gutters.

**Images**

I noticed the images to be used for the cards were really large so cropped these to a more appropriate size, I also ran them through Tinypng to compress them and have added width and height attributes to prevent page jank and enhance performance. All the images have alt tags.

**Layout**

I opted to use Flexbox for the layout for the image ‘grid’ using gap to set the gutters, in the past I’ve used nth-of-type to remove margins and maintain gutter widths, but it feels like gap is reasonably well supported now. I only made a simple responsive change, from a single column on mobile to three columns on wider screens. I planned to add an ‘in-between’ layout whereby the cards would still display as a single column, but the image and text would sit next to each other (image left text right), as things stand the images and card stretch a little too much before the media query kicks in at 880px. Unfortunately I didn’t have time to do this

**Misc**

I used a clipping path in the banner to create the offset effect, but having finished the test realised the clipping path wasn’t quite an exact match, however I don’t think it would be hard to replicate this. I chose the clipping-path to avoid downloading multiple images.

There’s some fairly basic animation on the links in the cards, and I would have animated the hamburger though unfortunately ran out time. In addition I only realised after I’d completed the test that the company logo should have a link back to the homepage on it.

I forgot the version of Atom I was using is running a prettifier of sorts, it can be really useful, but applies trailing slashes to standalone HTML tags, which would be something I’d look to remove in production.

**Time taken and elements not added**

The test took 2hrs and 2 minutes to do. I had hoped to have the whole page completed in about an hour half but set-up and perhaps my process (validation and overlaying etc) possibly cost me time. There was probably a little more nuance to the design than I originally thought too, I possibly underestimated the time required to do both mobile and desktop layouts.

Things I would like to have added or done would include:

* To add a Skip to Content link
* To have added a link back to the homepage on the company logo.
* To transform the hamburger to an X on click and focus - I added the event handler for this is in scripts.js but unfortunately didn’t have time to add the scss.
* To have made the logo smaller on mobile
* To have styled the type for mobile
* To have fixed mismatches made obvious by overlays
* To perhaps apply the image-text layout cards as outlined earlier
* To perhaps have added fluid type on headings
* To perhaps have made header-top fixed, perhaps using the object intersect API to change colour on scroll.
* To perhaps have added the more production focussed element class styling mentioned earlier and maybe used BEM.