

# PIXU Front End Development Course Page

---

For this technical challenge, I was tasked with designing a website, then building it on HTML and CSS. I opted for the option of designing a site which lets a user book onto a training course.

## Design & Planning

---

I first created a logo for the course using [Canva](#). The blue colour I chose for this became the starting point for the colour palette which I worked with for the rest of the project. I used the [Eye Dropper](#) Chrome extension tool to get the HEX code for this colour, then used an online tool called [Color Blender](#) to get 10 hues between the selected blue and white, which gave me my colour palette. I also used a [Colour Contrast Checker](#) to make sure the colours I chose passed AAA accessibility standards when used in conjunction with each other.

I then decided on what else the design would need:

1. A clear value proposition stating what the website offers and why users should chose this training course.
2. A simple, minimalistic, user centered design.
3. Clear information hierarchy using visual cues such as size, placement, and font weight.
4. A contact section with a call to action.
5. Some images to bring the design to life. I used [Unsplash](#) for this.
6. A form for the user to submit their details.
7. A footer with links to social media platforms, I used [Icons8](#) for this.

It's worth mentioning that I also used a few similar websites for inspiration, including [Makers](#), [Command Shift](#) and [General Assembly](#).

I then used [Figma](#) to make the design, you can see the finished version below:

[Front End Engineer Technical Challenge Design-compressed\\_1.pdf](#)

## Development

---

Semantic HTML is fundamental for:

- Accessibility
- SEO
- Readability and maintainability
- Cross-Browser compatibility

As such, I split the `body` of the webpage into a `header`, `main`, and a `footer`. Each of these was then split once more into other elements such as `section`. This laid out a template for me to flesh out the site and start adding more content using `img`, `nav`, `ul`, `li`, `a`, `p`, `h1`, `form`, `label`, `input` amongst others.

[MDN Web Docs](#) is a very useful resource to learn more about semantic HTML and much more.

I've used a combination of CSS Flexbox and CSS Grid to structure the site and position each element as per the design. Detailed manipulation of the box model was essential to master the spacing and layout of each element, every section, as well as the site as a whole.

[CSS Tricks - A Complete Guide to Flexbox](#) and [CSS Tricks - A Complete Guide to CSS Grid](#) are my go-to guides to remind me of the intricacies of both of these topics.

To make the site responsive, I used `vw` measurements so that fonts and images would shrink and grow relative to the screen size. As such, I only had to write 2 media query breakpoints to cater for screen sizes smaller than 600px and 500px. I used [Chrome Developer Tools](#) extensively to test the site on screens of all sizes.

This tool was essential for me to visualise the box model of each element and figure out the style I needed to change in order for the design to be replicated.

## Deployment

---

I deployed the site using [Render](#). You can check it out [here](#)

## Testing

---

I used [BrowserStack](#) to test for cross browser compatibility. I also used Chrome Developer Tools to test on devices with a slower internet connection, to ensure media was compressed enough to not slow down loading and impact user experience.