Since the project I was given was so open-ended, I decided to make a website about my two favourite things: word games and pigeons (no, I’m not joking). The pigeons are easy; I decided to make a kind of blog-type page.

Incorporating word games required more thought. I took inspiration from the New York Times, where I have a games subscription. The Spelling Bee is my second favourite of their games, after the crosswords. Spelling Bee is a game where you’re given a jumble of letters and you have to try and make as many words out of them as you can, and points are earned depending on the length of the word (you get the most for a pangram - a word that uses each of the letters at least once). Doing all that is beyond what I could have gotten done for this project, but I knew I could probably make a similar, but smaller scale, game.

I used a combination of flexbox and grid to create the layouts on each page. To keep things adaptable to different screen sizes I didn’t use any absolute positioning, except to keep the sidebar in place on desktop.

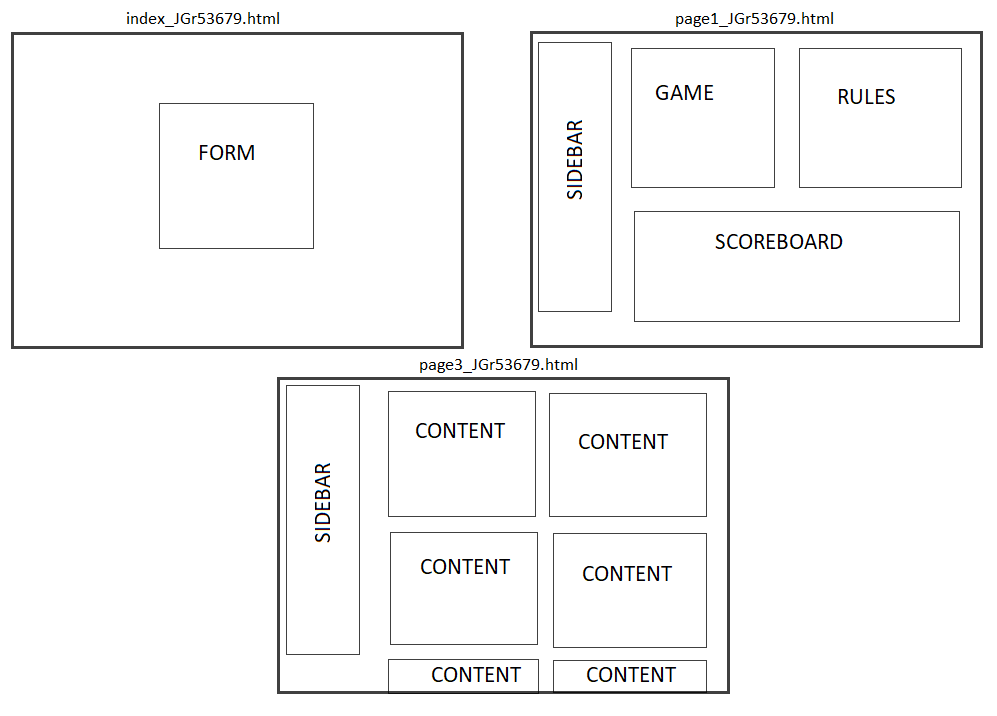
I started with the idea of making my layouts for mobile first. In my opinion, it’s far easier than going the other way around (desktop first). For the most part, web pages are automatically perfectly responsive and scalable to small screens before adding even a single line of CSS - a plaintext site will fit perfectly on any size screen without any overflow, overlapping content, or needing to scroll horizontally, etc. Desktop layouts have a lot more space to fill, and therefore generally need to be more complicated. In my opinion it’s easier to make media queries to add complexity for the desktop layout, rather than making queries to try and remove complexity for smaller devices.

With that in mind, when making the initial mobile layouts I tried to disrupt the default flow of the pages as little as possible, while making it look nice and trying to keep things from feeling cramped. To keep things visually coherent, I matched the backgrounds, and went with a kind of ‘paper card’ style on each page - each chunk of content in an off-white container with slightly rounded corners, and a box shadow to make it pop.

The login page page was the simplest, given that it mostly only consists of a single form and a small logo-type image. I wrapped the form in a section element, and made sure that it stayed centered and un-squished no matter the screen size. Adjusting it for desktop only required a small change to padding and width.

The other two pages have a lot more to them, including a navigation bar. The navigation bar has a pair of handsome pigeons as an icon, links to the other pages, and (on mobile) a title for the current page. On mobile, it’s placed at the top of the page as a header, and on desktop I positioned it as a sidebar instead. When positioned as a sidebar, there isn’t space for a heading anymore, but I did include a little surprise (a heart) when the pigeons are moused over. I removed that element on mobile because it simply isn’t practical - there’s no mouse to hover with, and I didn’t want it to be a clickable element. To compensate for the space taken up when there’s a sidebar, I made sure to include a left margin on the main section elements in desktop mode for both pages that use it.

The Word Scramble page (page 2) uses grid to arrange the big pieces of its desktop layout, and a little bit of flex just to arrange the game itself. On small screens I was okay with everything being in one tall column, but that wouldn’t do on a bigger scale. I used grid to arrange everything in two rows. I put the game and the rules on top, and the high score table underneath. The blog style page also uses a two column grid.



If you go through the site as intended, you land first on a kind of colourful mock login page. The background is an animated gradient. I initially tried to animate a change between the linear gradient and a radial one, but I changed it into its current state once I realized that I misunderstood how CSS animation works. What I did instead was increase the size of the background far beyond the edges of the page, and set it to slide from side to side.

The form asks for your name, age, favourite colour, and if you agree to the terms of use (which are visible if you hover over the text). The form input is validated with inline elements and Javascript. “Name” only checks that *something* is typed into the text box. “Age” uses a number type input, and does a couple more checks. It requires the age to be entered in digits, and for the age to be at least 18 but less than 110. For terms of use, I used two radio buttons (yes and no) and check to make sure that the “yes” option has been selected. There wasn’t really a good way to check colour selection, so you can get access to the entry links even without changing the selector to something other than black. The other options would have either required the user to select specific colours, or a super general colour (ie, anything but black). I didn’t like either of those options (what if black is someone’s favourite colour?), so I left it as is. Once the form is filled in correctly and the user clicks the submit button, two more buttons (really just decorated links) appear to let them in.

The part of my project I’m the proudest of is the word scramble game on page two. It was a lot of fun to program, and it was in the sweet spot for extending my skills without being too difficult to figure out in time.

The word game’s two major parts are the game itself, and the scoreboard. The game is controlled by five event listeners (one for each button). When the page is first loaded, you can start the game with a “start game” button. Subsequent games are started with a “play again” button instead. Apart from order of appearance, the buttons are essentially functionally identical - the only difference is that the “play again” button also sends your score to the scoreboard. When the game starts, several things happen: a word is randomly selected from a list of nearly 500, it’s letters are shuffled into a random order before being displayed onscreen, and a timer starts counting down. You’re given two buttons: ‘Shuffle’ and ‘Guess’. Shuffle reruns the function that scrambled the letters when the word was first chosen. You can do this as much as you need - it can be surprisingly helpful. In that function, as a last step before printing out the letters, I added a check to make sure that the letters didn’t get shuffled back into order. It’s extremely unlikely that that would happen, but just in case it does, the shuffle function would run again.

Either pressing enter or clicking the “Guess” button will submit your guess. If you got it right (it’s not case sensitive), a positive chime will play, your score will increase by one, and a new jumble of letters will appear. If you get it wrong, you’ll get a dull bleep instead of the chime, you’ll lose a point (the score can’t go into negative numbers), and the word won’t change until you either figure it out and time runs out.

At the end of the game, your score will be recorded on the high score table, if it qualifies. You’ll also be able to record your name if you wish, otherwise you’ll be listed anonymously. The table updates when you either submit your name, or choose to play again.

Media and References

--PHOTOS--

transparent astronaut

- https://www.vecteezy.com/png/14220737-cute-astronaut-line-art-design

transparent pigeon pair

- https://www.pngpix.com/download/pigeon-png-transparent-image-2

pixel heart

- https://www.onlygfx.com/6-pixel-heart-png-transparent/

william-of-orange

- This work created by the United Kingdom Government is in the public domain.

Jacobin pigeon

- https://commons.wikimedia.org/w/index.php?curid=38145649"

Fairy Swallow

- https://commons.wikimedia.org/w/index.php?curid=2267365"

--AUDIO--

correct answer ping, chime 1

- https://www.zapsplat.com/?s=correct&post\_type=music&sound-effect-category-id=

Alert, error tone 2

- https://www.zapsplat.com/music/alert-error-tone-2/

UI glassy notification alert 2

- https://www.zapsplat.com/music/ui-glassy-notification-alert-2/

--VIDEO--

Frillback Pigeons

- https://www.youtube.com/watch?v=6ATSWyXlRDc&ab\_channel=FifeFarm%28OH%29

--OTHER–

Grid layout help

- angrytools.com/css-grid/

Color pallettes

- cssgradient.io

Word list

- thefreedictionary.com/7-letter-words.htm

CSS attributes and animation, JS setInterval(), swap multiple variable in one line, flexbox settings

- w3schools.com