54 20 Wiring up with javascript the button of the game

54 40 Building our personal skills section

Create a new section tag with a h2 tag. Put skills inside the h2 tag

55 20 we will use a table tag

use the table tag <table>

<table>

<thead>

<tr>

<td></td>

</tr>

</thead>

<tbody></tbody>

</table>

56 30 add area and technologies for your table in the td tags

57 05 add in the tbody other tr tags. Create 4 tr and each with 2 td. 4 rows with 2 columns each

58 15 add Javascript, Python, and other 2 technologies

1 00 00 Change the td to th in area and technologies

It will look like a fine table at the moment

You could add logos too

------SECTION 3 PROJECTS

1 01 40 create a third section tag, with h2 tag inside and put Latest Projects

We will create a template for each project

1 02 10 we will create a div with a class of cards-grid, beneath h2

1 02 40 inside this div open another div with class project-card

we will put information element and image

1 02 50 in information we will create a div for the information

it will have the date in a paragraph tag

1 03 40 h4 will have the title of the project, underneath the p tag

1 04 05 we will add a row of labels with the technologies we use

it will show employers what technologies we have used

1 04 40 we will create a div with class tech-list and inside it, another div with class card-button-secondary in it

1 05 10 add a paragraph with technologies

add 2 copies of this div

1 06 00 add a <p> tag underneath the tech-list div and add some information

1 07 00 Beneath the paragraph add another div class project-links

add an a tag with \_blank attribute and href / at the moment

1 07 50 It should look like this

<a target="\_blank" href="/">Live</a>

<a target="\_blank" href="/">Link</a>

1 08 05 within the project-card div, right at the bottom add another div class= img-div. it will contain the image

create a img tag with class big-img, src / at the moemnt and alt erick-thumbnail

the source of the image will be in the public directory that we made

so add the full address /1\_web\_portfolio/public/fantacss.png

-------2 ADD PORTFOLIO CONTENT

1 09 10 now the image will show up. This code will change later

1 09 35 copy the project-card div and add it twice beneath itself

there will be copies of the date

1 10 20 we will add the style now

1 10 35 for style we are using fantacss which is why we have some styling.

You can fiddle with the values

we will use our style.css

-----------STYLE.CSS

1 11 05 We will be working in the style.css

All of our styles will be in our style css, which we reference in our main content index.html

we will start from top to bottom, so the style approximately match the contents of the page

1 11 50 we will work with the body tag

write the body and the {} curly braces

<link rel="stylesheet" href="style.css"/>

11 12 10 Give it a min height parameter to 100vh, it is the view height of the element

display: flex;

benath we set the flex-direction: column;

1 12 40 Now the body has 3 direct children

header,

main,

footer

max-width: 800px as that is the general width of blogs

width= 100% so it fills until the 800 px

margin will be 0 and auto, the content will self center on the screen

1 14 00 here we can see that the maximum width is 800 px and then the rest is margins. it is also centered

1 14 20 the header will have display flex so the icons sit alongside the text

items will be align in the center, and will be justified with a space between. It will put any direct children with a space between them, putting all items evely separated

1 15 15 padding: 2rem to all: so it will push everything a little bit

header,

main,

footer

1 15 20 giv the header a gap of 1 rem so it is pushed a bit from the borders

1 15 37 Moving to the "nav" element, we do not need a justify, and give it :

nav {

display: flex;

align-items: center;

gap: 1rem;

}

They'll be aligned in the same row

1 16 10 below nav we want to select the icon

it will target any icon tabs in the nav like this

nav i {

font-size: 1.3rem;

}

This will complete the style for the header

1 16 30 we will work wit the main

main,

section {

display: flex;

flex-direction: column;

}

1 16 52 we will work with the main

main {

gap: 4rem;

}

1 17 25 section will have a gap of 1 rem

------CLASSES

1 17 50 styling intro header

remember that in order to stylize classes you have to use a dot .<class>

we will stylize the paragraph within the class. Giving a bit of space

.intro-header p {

padding-top: 0.25rem; /\*will create a bit of space between title and intro space\*/

}

1 18 30 add .intro-header and give it a max-width of 500px

It will sit to the left with this

1 18 55 main, header and footer use maximun width , intro-description will have a maxwidth of 500px

1 19 10 in links-container, we will spread them with display flex, align-items center

.links-container {

display: flex;

align-items: center;

gap: 1rem;

padding: 1rem 0;

}

they will have a gap and accross the row, they're centered vertically

1 20 00 for the arrow icon, we will be be using a var to match the links

transform will allow us to rotate the arrow in this case, with a function

it will point to the upper right direction

1 21 20 we will work with the .color-div

.color-div {

display: flex;

flex-direction: column;

gap: 0.5rem;

}

1 21 50 stylizing latest projects

.cards-grid has all the cards. we will make it spread the cards and differentiate it between them

.cards-grid{

display: flex;

flex-direction: column;

gap: 2rem;

}

1 22 25 We add style to .project-card

1 22 40 in order to not rewrite the same information of styles over and over, you could find all the classes that use certain information, like display: flex; and add them together in a single statement.

1 23 00 now the image and the information element are separated by a gap. This will be important when going to the mobile version

1 23 20 add .project-links {}

these are the links in the section, they will have a small gape between themselves and the sections

1 23 50 we can see how it'd look in mobile with inspect > Ctrl+Shift + M, the responsive design mode

We can see the complete layout of the ap.

The images are rounded due to fantacss

We need to add mediaqueries

1 24 22 After project links we add the .tech-list{}

the list of technologies of each project

.tech-list{

display: flex;

}

This is only the list, and we will work with each div in

.tech-list div{

display: flex;

}

1 25 10 we add a bit of spacing for every technology used and round their corners

.tech-list div {

padding: 0.1rem 0.5rem; /\*first is vertical, second is horizontal\*/

border-radius: var(--border-radius-small);

}

1 25 34 let's align the items centrally in .tech-list wit align-items: center;

1 25 56 flex-wrap: wrap will let the row wrap to the next line if it overflows the screen

1 26 24 before the media queries

.tech-list p {

font-size: 0.8rem;

}

so it is slightly smaller than the norm

1 27 08 We can see the cards have spacing in the mobile version

we will throw a media query

1 27 15 Media Query: dictate what style apply to your document at what screen sizes.

Media queries allow you to apply CSS styles depending on a device's media type (such as print vs. screen) or other features or characteristics such as screen resolution or orientation, aspect ratio, browser viewport width or height, user preferences such as preferring reduced motion, data usage, or transparency.

Work with mobile first

1 27 25 All the tech classes apply to mobile, so at the bottom add:

when the page reaches width of 640 px it will override previous styles with the following parameters

@media (min-width: 640px) {

.project-card{

flex-direction: row; /\*everthing is vertically oriented, so this will be in row\*/

}

}

1 28 28 there's 2 direct children

one is the info section, which will have vertical orientation, but as a row, it will have the image along the info, and not below

1 28 55 we will select the prject card class and img-div class in the project card, and it will have a max width of 30% of the parent element

.project-card .img-div{

max-width: 30%;

}

it can be smaller than 30%, but then stops growing

The element to the side takes the rest of the space.

After the 640px it will be row, and the image 30% size max

1 30 00 the img will be 30% of the 800 px that we set in

header,

main,

footer {

max-width: 800px;

width: 100%;

margin: 0 auto;

padding: 2rem;

}

1 30 10 Now the CSS is done, time for the Javascript