|  |  |
| --- | --- |
| @font-face {  font-family: Pixelify-Sans;  src: url("PixelifySans-VariableFont\_wght.ttf");  } | This CSS helps to register the new external font that I intend to use for this page |
| .container{  display: grid;  width: 100%;  height: 850px;  grid-template-columns: repeat(3, 1fr);  grid-template-rows: repeat(10, 1fr);  background-color: beige;  grid-gap: 2px;  align-items: stretch;  justify-items: stretch;  align-content: start;  justify-content: end;    } | This CSS gives the characteristics of the main container for the page. I decided to make it grid style, width if the page is extended, and the rest I give the measurements of how I want the space to be divided. Three columns, 10 rows and give the properties of how elements align along the axis. If the area gets stretched, the element also stretches to fill its area |
| .mainheader{  background-color: rgb(145, 15, 145);  grid-column-start: 1;  grid-column-end: 4;  grid-row-start: 1;  grid-row-end: 1;  text-align: center;  color: rgb(228, 228, 0);  font-family: Pixelify-Sans;  font-size: 38px;  } | This is the header for the website. It states the color of the background and the space designated for it in the grid format. It specifies beginning to end. Also the text align for the name of the website and color. In here the font-family I used the font I downloaded separately. |
| .mainheader h2{  margin-bottom: 5px;  }  .mainheader h1{  margin-top: 5px;  } | This was intented to reduce a gap at the bottom of the text in the header. Only for aesthetic purposes |
| .navigationbar{  background-color: black;  grid-column-start: 1;  grid-column-end: 4;  grid-row: 2/2;  font-family: Pixelify-Sans;  text-align: center;  font-size: 25px;  justify-content: center;  padding: 15px;    } | This is the information of the navigation bar. Designated the size, color and the charactericstics. I wanted it to be a thin black bar, so that’s why I picked row 2/2 and selected the special font again. I wanted to navigation buttons to be in the middle, so I aligned the text to center. The size of the font is selected here as well and the padding was done to create the effect of the text placed in the middle and not at the upper or lower side of the bar. |
| .articlebox{  background-color: rgb(145, 15, 145);  grid-column-start: 1;  grid-column-end: 3;  grid-row: 4/10;  border: 2px solid black;  overflow: auto;  } | These are the properties set for the space designated for the article section using the grid format. Delimited with a 2px border and the overflow set to auto so the text is still readable with just scrolling. |
| .asidebox{  background-color: black;  grid-column: 3/4;  grid-row: 4/10;  justify-content: center;  align-items: center;  display:flex;  border:2px solid black;  flex-direction: column;      } | This is another container and it is intended to hold a picture. To facilitate the handling of the item within this container, I used display flex , and since I wanted to have a title on top of the image, I used flex-direction column. And all contents are justified in the center. |
| .asidebox img{  max-width: 100%;  max-height: 100%;  margin-left: auto;  margin-right: auto;  display: block;    }  .highlight{  color:rgb(228, 228, 0);  text-decoration: solid;  } | These are the settings for the picture inside the asidebox. Max height and width ensures theimage does not exceed 100% of its containing element and the margin properties centers the image. The display block changes the display property to block, allowing the margin properties to take effect.  This was done to highlight in yellow several sentences contained in a <p></p> |
| .footerbar{  background-color: black;  grid-column: 1/4;  grid-row: 11/11;  } | These are the characteristics for the footer present in all pages. It is in the last row and in all columns |
| .navigationbar a{  text-decoration: none;  color: white;  margin: 10px;  } | Set of properties to make the navigation buttons free of hyperlink underling and to create space in-between them |
| p{  color:white;  font-family: Arial, "Helvetica Neue", Helvetica, sans-serif;  line-height: 1.5;  font-size: 20px;  text-align: justify;  margin: 8px;  } | These are the set of properties for each of the paragraphs. With this set, I picked the color, family font and the height of the lines. I also wanted the text to have a margin with the limits of the container, so I set it up to a margin of 8px. |
| h4{  color: rgb(228, 228, 0);  font-family: Arial, "Helvetica Neue", Helvetica, sans-serif;  font-size: 20px;  text-align: center;  }  h1{  font-size: 18px;  font-family: Arial, "Helvetica Neue", Helvetica, sans-serif;      } | These are the properties for h4 and h1 types of titles used throughout the website. The color and font family are set to keep consistency with the rest, and the align preference to center. |
| nav a:hover{  background-color: rgb(145, 15, 145);  } | This is the color the navigation buttons will get when hover over. Responsive property |
| ul{  color:white;  font-family: Arial, "Helvetica Neue", Helvetica, sans-serif;  line-height: 1.5;  font-size: 20px;  text-align: justify;  margin: 8px;  } | These are the characteristics for the lists used in the website. Every list will follow those characteristics of a margin of 8px with the edges of the container, a size of 20px and text will be justified in Arial family font. |
| .footerbar{  color: white;  text-align: center;  } | These are the properties for the text in the footer. White font and centered. |

FLIPCARD PROPERTIES

|  |  |
| --- | --- |
| .containerfeatured{  display:flex;  flex-wrap: wrap;  background-color: beige;  } | This is the container that will hold the section where I placed all the flip-cards in my page. I set it to be flex to control the layout and give it responsive properties. |
| article{  grid-column-start: 1;  grid-column-end: 4;  grid-row-start: 3;  grid-row-end: 10;  overflow: auto;      } | This article is the parent container for the flip-cards. I designated horizontal space and few rows. That way the website will show the flip cards across the space. Overflow auto, so when the text is longer than the container, the user can scroll and continue reading. |
| .flip-card {  background-color: transparent;  width: 300px;  height: 300px;  border: 1px solid #f1f1f1;  perspective: 1000px;  margin: 5px;    } | These are the properties for the flip card. I gave a width and height; the background is transparent and there is the 3D effect given by the perspective 1000px. And a margin of 5px to keep distance between the flip cards. |
| .flip-card-inner {  position: relative;  width: 100%;  height: 100%;  text-align: center;  transition: transform 0.8s;  transform-style: preserve-3d;  } | In here, positions the inner card to relative, width and height to 100% of its container, text aligned to the center and then the details of the flipping effect and the time |
| .flip-card:hover .flip-card-inner {  transform: rotateY(180deg);  } | The hover effect when mouse placed on flip card it will rotate over the Y axis |
| .flip-card-front, .flip-card-back {  position: absolute;  width: 100%;  height: 100%;  -webkit-backface-visibility: hidden;  backface-visibility: hidden;  } | This establishes that both elements are positioned absolutely within their container, take up the full space of height and width and hide their back faces when rotated. |
| .flip-card-front {  background-color: #bbb;  color: black;  } | This establishes the color and background of the front of the flipcard. |
| .flip-card-back {  background-color: rgb(145, 15, 145);  color: white;  transform: rotateY(180deg);    } | In here we see the background color (purple) and the text color (white) and also the rotate effect on the Y axis for the flipcard |

MEDIA QUERY

|  |  |
| --- | --- |
| @media (max-width: 768px) {  .container {  grid-template-columns: 1fr;  grid-template-rows: auto;  } | This is set for screens up to 768px and establishes that the container will fit in a 1 column and sets the grid layout to have rows that automatically adjust their height based on the content |
| .mainheader{  grid-row:1/1;  } | The header will shrink to 1/1 row |
| .navigationbar{  grid-column-start: 1;  grid-column-end: 2;  grid-row: 2/2;  } | The navigation bar will occupy column 1 and 2 and the row 2/2 |
| .articlebox{  grid-column-start: 1;  grid-column-end: 2;  grid-row-start:4;  grid-row-end: 8;  } | Here the query is set to shrink the article box to column 1 and 2, and to occupy row 4 to 8 so the user can still read the articles or images contained in there |
| .asidebox {  grid-column-start: 1;  grid-column-end: 2;  grid-row-start: 9;  grid-row-end: 12;  }  } | For the asidebox, it will be placed in column 1 and 2 and occupy row 9 to 12 when media query is applied. |