

Group Members: Parabjot Chander, Benjamin Mushyakov and Andrew Zheng
CSCI 355: Internet and Web Technologies
Professor Fried
February 10, 2023

Gray Paper (34 different points)

What each WebPage Contains:

1. A header 1 tag in which tells the page you are in, for example, in the homepage, the h1 tag contains Chander Mushyakov Zheng Pizzeria
The CMZ Pizzeria Logo (Image) and nav tag nested with a unordered list of anchor tags that are created to reference other web pages are both in separate divs, nested in a bigger div called “container”

```
<div id="container1">

    <div id="logo">

        

    </div>

    <div id="navigation-bar">

        <header>

            <nav>

                <ul>

                    <li> <a href="homepage.html"> Home </a> </li>

                    <li> <a href="menu.html"> Our Menu </a> </li>

                    <li> <a href="aboutus.html"> About Us </a>
</li>

                    <li> <a href="contactus.html"> Contact Us </a>
</li>
```

```

        </ul>

    </nav>

</header>

</div>

</div>

```

2. The div container is displayed as flex, so that the web page's navigation bar and business logo's sizes will change as the user changes the web browser size. We used the flex property so that the logo doesn't grow or shrink at the same level as the nav bar when the user increases the browser size.

```

#container1{
  background-color: rgb(215, 0, 64);
  display: flex;
  height: 7em;
}

#logo{
  flex:1;
}

#navigation-bar{
  flex:2;
}

```

3. The image has a width and height of 50% and 100% (RELATIVE UNITS)
The navigation bar is styled to inline block so the anchor tags can line up with one another, and a padding of 1em
The anchor tags are of color white, the list type is set to none so bullet points can disappear, are in bold, font size is set 150% (24px), when a anchor tag is hovered, the background for the anchor link turn black

```

#navigation-bar nav ul li{
  list-style-type: none;
  display: inline-block;
  padding: 1em 1em;
}

```

```
#navigation-bar nav ul li a{
  color: white;
  text-decoration: none;
  font-weight: bold;
  font-size: 150%;
}

#navigation-bar nav ul li a:hover{
  background: black;
}

#logo img{
  width: 50%;
  height: 100%;
}
```

4. A div is created for the footer of the Web Page's, two p tags are used naming the group members and Copyrights of the business: CMZ Pizzeria

```
<div id="footer">
  <p id="firstFooter"> Founders: Parabjot Chander, Benjamin
Mushyakov and Andrew Zheng </p>
  <p> Copyright 2023 &copy; CMZ Pizzeria </p>
</div>
```

5. The footer has a red background and has a height of 5 em, the p tags are in the color of white, are centered, the first footer has a padding on the top of 1 em to make the footer look more appealing

```
#footer {
  background-color: rgb(215, 0, 64);
  color: white;
  text-align: center;
  height: 5em;
}
```

```
#firstFooter {
  padding-top: 1em;
}
```

Homepage:

6. A div container is created for the slogan of the business, slogan is in a h2 tag

```
<div id="slogan">
  <h2>You Don't Have To Say Thank You, Our Pizza Is The
Desire.</h2>
</div>
```

7. The div has a red background and is given a 200% height, the h2 tag is centered and has a white color

```
#slogan{
  background-color: rgb(215, 0, 64);
  height: 200%;
}
```

```
#slogan h2 {
  color: white;
  text-align: center;
}
```

8. A div tag with the id of cards has nested divs with classes that represent the panels that will be displayed, each with h3 saying the pizza name and a background img of that pizza type. The div with a panel active class will be displayed.

```
<div class="cards">

  <div class="panel active" style="background-image:
url('image/largepie.jpg') ">
    <h3>Regular Cheese Pizza</h3>
  </div>

  <div class="panel" style="background-image:
url('image/pepperonipizza.jpg') ">
    <h3>Pepperoni Pizza</h3>
  </div>

  <div class="panel" style="background-image:
url('image/vegatarianpizza.jpg') ">
    <h3>Vegetarian Pizza</h3>
  </div>
```

```

        <div class="panel" style="background-image:
url('image/chickenandbrocpizza.jpg')">
            <h3>Chicken and Broccoli Pizza</h3>
        </div>

        <div class="panel" style="background-image:
url('image/spinachpizza.jpg')">
            <h3>Spinach Pizza</h3>
        </div>

    </div>

```

9. The panel div's are flex items, the active panel has a higher flex size (displayed), the h3 tags are displayed at the bottom left with 0 margins. The active panel has an opacity of 1, can see the h3 tag but not the other panels with opacity of 0. The active panel's h3 has a white text color, red background. When a user hovers over the panel div's, the mouse has a pointer cursor.

```

.cards{
    display: flex;
    width: 90vw;
}

.panel{
    background-size: auto 100%;
    background-position: center;
    background-repeat: no-repeat;
    height: 80vh;
    border-radius: 50px;
    color: white;
    cursor: pointer;
    flex: 0.5;
    margin: 10px;
    position: relative;
    transition: flex 0.7s ease-in;
}

.panel h3{
    font-size: 2em;
    position: absolute;
    bottom: 20px;

```

```

    left: 20px;
    margin: 0;
    opacity: 0;
  }

  .panel.active{
    flex: 5;
  }

  .panel.active h3{
    opacity: 1;
    transition: opacity 0.3s ease-in 0.4s;
    left: 10em;
    background-color: rgb(215, 0, 64);
    color: white;
  }

```

JavaScript Expanding Cards:

10. A `querySelectorAll` method is called first so that all the panel div's with the panel class are referenced. An `addEventListener` is created so that when a panel is clicked, a helper function called `removeActiveClasses()` is called which makes all the panel div's not active. The panel div that is clicked is set to active so that it can be displayed for the user to see.

```

const panels = document.querySelectorAll('.panel');

panels.forEach( (panel) => {

  panel.addEventListener('click', () => {

    removeActiveClasses()
    panel.classList.add('active')

  })

})

function removeActiveClasses() {

```

```
panels.forEach( (panel) => {
  panel.classList.remove('active')
})
}
```

11. A div container is created for the discounts title is in a h2 tag, the div has a red background, the h2 is of white color and is centered

```
<div id="discounts">
  <h2>Deals and Discounts</h2>
</div>
```

```
#discounts{
  background-color: rgb(215, 0, 64);
}

#discounts h2 {
  color: white;
  text-align: center;
}
```

12. A div class body 2 is created as the whole container for the split landing page, having 2 nested divs with paragraphs and images inside each.nested div

```
<div class="body2">
  <div class="udemyCourse2 hover-left">

    <div class="split left">
      <p class="fontsplit">
        Monday to Friday <br> <br>
        Buy One Pizza Get One Drink Free <br>
        10 % Off All Pizzas <br>
        5 % Off All Drinks
      </p>

      
    </div>
```

```

        <div class="split right">
            <p class="fontsplit">
                Saturday to Sunday's <br> <br>
                Buy One Pizza Get Two Drinks Free <br>
                15 % Off All Pizza's <br>
                10 % Off All Drinks
            </p>

            
        </div>

    </div>

</div>

```

13. The split left and split right divs start at the respective corners with left:0 and right: 0 CSS properties. At the webpage start (initially) where the split left has a hover width of 75% and right hover width 25%, the speed in which the hover of the div's will change at 1000ms (1 second). CSS variables are created –hover-width and –other-width for when the hover-left .left, hover-left .right, hover-right .right, and hover-right .left classes are clicked, the CSS variables are used. The font split class represents the p tags having a red background, text color of white, absolute positioning of left: 60% and top: 20% and with a font size of 1.5 em (24px usually). The whole body container has a 30vh, the overflow is hidden to avoid scrollbars .

```

:root{
    --left-bg-color:white ;
    --right-bg-color: rgba(87,84,236,0.7);
    --hover-width: 75%;
    --other-width: 25%;
    --speed: 1000ms;
}

```

```

.body2{
    height: 30vh;
    overflow: hidden;
    margin:0;
}

```



```
}

.udemyCourse2{
  position: relative;
  width: 100%;
  height: 100%;
  background: papayawhip;
  overflow: hidden;
}

.split {
  position: absolute;
  width: 50%;
  height: 100%;
  overflow: hidden;
}

.split.left{
  left: 0;
  background-repeat: no-repeat;
  background-size: cover;
}

.split.left::before{
  content: '';
  position: absolute;
  width: 100%;
  height: 100%;
  /* background-color: var(--left-bg-color); */
}

.split.right{
  right: 0;
  background-repeat: no-repeat;
  background-size: cover;
}

.split.right::before{
  content: '';
  position: absolute;
```

```

width: 100%;
height: 100%;
/* background-color: var(--right-bg-color); */
}

.split.right, .split.left, .split.right::before, .split.left::before{
  transition: all var(--speed) ease-in;
}

.fontsplit{
  background-color: #D63230;
  font-size: 1.5rem;
  color: white;
  position: absolute;
  left: 60%;
  top: 20%;
  transform: translate(-50%);
  white-space: nowrap;
}

.hover-left .left{
  width: var(--hover-width);
}

.hover-left .right{
  width: var(--other-width);
}

.hover-right .right{
  width: var(--hover-width);
}

.hover-right .left{
  width: var(--other-width);
}

```

JavaScript Split landing Page Explained:

14. The left split div and the split right div, and the whole container div are referenced using `querySelector()` helper method, Four event listeners are implemented so that when a mouse is on the left split div, their will being hovering to the left, the class becomes `hover-left`, (75%) view of whole container is devoted to that split left, when you hover off or mouse leave, the split left div is removed from the `hover-left` nested class, the view becomes 50% for both, this applies the split right div as well.

```
function removeActiveClasses() {  
  
    panels.forEach( (panel) => {  
        panel.classList.remove('active')  
    })  
  
}  
  
const left = document.querySelector('.left');  
const right = document.querySelector('.right');  
const container = document.querySelector('.udemyCourse2');  
  
left.addEventListener('mouseenter', () =>  
    container.classList.add('hover-left') );  
left.addEventListener('mouseleave', () =>  
    container.classList.remove('hover-left') );  
  
right.addEventListener('mouseenter', () =>  
    container.classList.add('hover-right') );  
right.addEventListener('mouseleave', () =>  
    container.classList.remove('hover-right') );
```

About Us:

HTML

15. This is one of the FAQ boxes that utilized unordered list for different info

```
<p class="faq-text">  
    There are multiple ways to order from CMZ Pizzeria.  
</p>  
<ul class="faq-text">  
    <li>You can order through our Website by selecting the menu items. </li>  
    <li>You can place an order through a phone call number:number:718-453-1234.</li>  
    <li>You can walkin to our store location and place an order.</li>  
</ul>
```

16. This is button class a setup for udemy idea FAQ collapse, it use “fas fa-chevron-down” for the button to toggle down the FAQ box

```
<button class="faq-toggle">
  <i class="fas fa-chevron-down"></i>
  <i class="fas fa-times"></i>
</button>
</div>
```

17. This is for general info section below FAQ, it use grid-container to contain heading for each different section info

```
<h3 class="gititle">General Information</h3>

<div class="grid-container">

  <div class="companyInfo">
    <h3> Business Ethos </h3>
  </div>

  <div class="companyInfo">
    <p> Selling pizza through delivery or store pickup with the best of our
ability. <br>
    Driving force is make sure our customers are satisfied with our quality.
  </p>
  </div>

  <div class="companyInfo">
    <h3> Target Audience </h3>
  </div>
```

```
<h3 class="gititle">One of the Founders Explains The Origins</h3>

<div class="flex-founders">

  <div>
```

```

        
    </div>

    <div id="founderScope">
        <p id="founderInfo"> Hello, my name is Parabjot Chander. A little background
about me is <br>
            that I am from <em> NYC </em>, I enjoy eating pizza while watching
sports. <br>
            I am a NBA and NFL fan, my favorite teams from the respective sports are
the <br>
            New York Knicks and the New York Jets. Me, Andrew and Benjamin decided to
start this <br>
            adventure because we adore pizza and are long time friends and business
partners.<br>
        </p>
    </div>

</div>

```

CSS

18. Aside from the same color background for each web page, this is style for javascript (FAQ box) The style of width, font, border etc is written for active and active:before. Before is when the FAQ hasn't collapsed and the active container is when the addEventListener is toggle/clicked on by the user and it will expand the FAQ box with its own style.

```

.faq-container {
    max-width: 600px;
    margin: 0 auto;
}

.faq {
    background-color: transparent;
    border: 1px solid #9fa4a8;
    border-radius: 10px;
    margin: 20px 0;
    padding: 30px;
    position: relative;
    overflow: hidden;
    transition: 0.3s ease;

```

```
}

.faq.active {
  background-color: #fff;
  box-shadow: 0 3px 6px rgba(0, 0, 0, 0.1), 0 3px 6px rgba(0, 0, 0, 0.1);
}

.faq.active::before,
.faq.active::after {
  content: '\f075';
  font-family: 'Font Awesome 5 Free';
  color: red;
  background-color: transparent;
  font-size: 7rem;
  position: absolute;
  opacity: 0.2;
  top: 20px;
  left: 20px;
  z-index: 0;
}

.faq.active::before {
  color: orange;
  background-color: transparent;
  top: -10px;
  left: -30px;
  transform: rotateY(180deg);
}

.faq-title {
  margin: 0 35px 0 0;
}

.faq-text {
  display: none;
  margin: 30px 0 0;
}

.faq.active .faq-text {
  display: block;
}

.faq-toggle {
  background-color: transparent;
  border: 0;
  border-radius: 50%;
  cursor: pointer;
  display: flex;
  align-items: center;
```

```

justify-content: center;
font-size: 16px;
padding: 0;
position: absolute;
top: 30px;
right: 30px;
height: 30px;
width: 30px;
}
.faq-toggle:focus {
outline: 0;
}
.faq-toggle .fa-times {
display: none;
}
.faq.active .faq-toggle .fa-times {
color: #fff;
display: block;
}
.faq.active .faq-toggle .fa-chevron-down {
display: none;
}
.faq.active .faq-toggle {
background-color: #9fa4a8;
}

.center {
display: block;
margin-left: auto;
margin-right: auto;
width: 40%;
}

```

19. This is style for the general info section for the page, the display is flex and use similar coloring matching the web page.

```

.companyInfo{
background-color: rgb(215, 0, 64);
}

.companyInfo p, .companyInfo h3{
color: white;
text-align: center;
}

```

```
}

.gititle{
  text-align: center;
  color: white;
  background-color: rgb(215, 0, 64);
}

.flex-founders{
  display: flex;
  justify-content: center;
  gap: 10px;
}

#founders-pic{
  width: 300px;
  height: 300px;
}

#founderInfo{
  background-color: rgb(215, 0, 64);
  color: white;
  text-align: center;
}

#founderScope{
  width: 200%;
}

.companyInfo{
  background-color: rgb(215, 0, 64);
}

.companyInfo p, .companyInfo h3{
  color: white;
  text-align: center;
}

.gititle{
  text-align: center;
  color: white;
  background-color: rgb(215, 0, 64);
}
```



```

.flex-founders{
  display: flex;
  justify-content: center;
  gap: 10px;
}

#founders-pic{
  width: 300px;
  height: 300px;
}

#founderInfo{
  background-color: rgb(215, 0, 64);
  color: white;
  text-align: center;
}

#founderScope{
  width: 200%;
}

```

Javascript

20. This is the javascript used from the udemy idea FAQ collapse, this script is used for the about us page. To explain there is a toggle button on the web page, when it is clicked through the addEventListener registered the user interaction. It will switch the FAQ box to active, expanding the box and collapse it when clicking on it again.

```

const toggles = document.querySelectorAll('.faq-toggle')

toggles.forEach(toggle => {
  toggle.addEventListener('click', () => {
    toggle.parentNode.classList.toggle('active')
  })
})

```

Contact Us:

HTML

21. This is used for the contact form with submit and cancel button by setting a class for input within the textbox. It alerts the user a message after submitting a message. The width of the table is 100%

```
<form name="Contact Form" id="form" onsubmit="alert('Thank you, we will
contact you as soon as possible')">
    <table class="table" width="100%">
        <tr>
            <td><label>Name:</label></td>
            <td><input class="input" required type="text" name="name"
id="name"></td>
        </tr>

        <tr>
            <td><label>Email:</label></td>
            <td><input class="input" required type="text" name="email"
id="email"></td>
        </tr>

        <tr>
            <td><label>Message:</label></td>
            <td><textarea class="input" name="message" ></textarea></td>
        </tr>
        <tr>
            <td></td>
            <td><button class="btn-submit" type="submit">Send Message</button>
                <button class="btn-cancel"
href="contactus.html">Cancel</button></td>
        </tr>
    </table>
</form>
```

22. Google Map Embedded Image:

```
<div class="map" style="width: 100%;height: 500px;" id="map" >
    <iframe
src="https://www.google.com/maps/embed?pb=!1m18!1m12!1m3!1d3022.8088967142417!2d-73.82
57503!3d40.7442304!2m3!1f0!2f0!3f0!3m2!1i1024!2i768!4f13.1!3m3!1m2!1s0x89c2606f162bbc4
5%3A0xeeadcfe214c64145!2s58-59%20Main%20St%2C%20Flushing%2C%20NY%2011355!5e0!3m2!1sen!
2sus!4v1676767167173!5m2!1sen!2sus" width="600" height="450" style="border:0;"
allowfullscreen="" loading="lazy"
referrerpolicy="no-referrer-when-downgrade"></iframe></div>
```

23. Company info such as address, contact info, and store hrs are used in a table and `
` is used to break into different lines.

```
<h2>Address:</h2>
    <b>CMZ Pizzeria</b><br>
    58-59 Main Street,<br>
    Flushing,<br>
    Queens NY 11355.
    <hr>
    <h2>Contact Info:</h2>
    <b>Email:</b>cmz@pizzeria.com<br>
    <b>Number:</b> 718-453-1234
    <div id = table>
    </td>
  </tr>
</table>
<div id=table>
  <table>
    <caption> Store Hours </caption>
    <tr>
      <th> Day </th>
      <th> Time</th>
    </tr>
    <tr>
      <td>Sunday</td>
      <td>10:00am - 5:00pm</td>
    </tr>
    <tr>
      <td>Monday</td>
      <td>9:00am - 8:00pm</td>
    </tr>
    <tr>
      <td>Tuesday</td>
      <td>9:00am - 8:00pm</td>
    </tr>
    <tr>
      <td>Wednesday</td>
      <td>9:00am - 8:00pm</td>
    </tr>
    <tr>
      <td>Thursday</td>
```

```

        <td>10:30am - 9:00pm</td>
      </tr>
    <tr>
      <td>Friday</td>
      <td>10:30am - 9:00pm</td>
    </tr>
    <tr>
      <td>Saturday</td>
      <td>10:30am - 9:00pm</td>
    </tr>
  </table>

```

24. This is header that contains logos and navigation bar for each web page.

```

<div id="container3">

  <div id="logo3">
    
  </div>

  <div id="navigation-bar3">

    <header>

      <nav>

        <ul>

          <li> <a href="homepage.html"> Home </a> </li>

          <li> <a href="menu.html"> Our Menu </a> </li>

          <li> <a href="aboutus.html"> About Us </a> </li>

          <li> <a href="contactus.html"> Contact Us </a> </li>

        </ul>

```

25. This class is used for the udemy idea of loading the webpage with a blur effect.

```

<div class="bg">

```

CSS

26. The web page background is set the same as all the other pages with a red and orange/white like color. This includes the tables for store info and contact form

```
body{
  background-color: papayawhip;
}

h1 {
  background-color: rgb(215, 0, 64);
  color: white;
  text-align: center;
}

#footer1{
  background-color: rgb(215, 0, 64);
  color: white;
  text-align: center;
  height: 5em;
}

#table table{
  border-collapse: collapse;
  font-size: 0.9em;
  margin: 1em 0;
  min-width: 16em;
}

#table table tr th{
  background-color:white;
  text-align: left;
  color: black;
}

#table table tr td{
  background-color: white;
}

#table table caption{
  background-color: #F71735;
  color: white;
}

.btn-submit{
  background-color: #F71735;
```

```

color: white;
}
.btn-cancel{
background-color: #F71735;
color: white;
}
.input{
width: 80%;
height: 30%;
}

```

JavaScript

27. This is Udemy's idea of loading with a blur effect. This was imitated and used to load the background of the contact us webpage. It works by setting

```

if (document.querySelector('.bg') != undefined ) {
  const loadingText = document.querySelector('.loading-text');
  const bg = document.querySelector('.bg');
  var load = 0;
  var int = setInterval(blurring, 30)
  function blurring() {
    load++;
    if (load > 99) {
      clearInterval(int);
    }
    loadingText.innerHTML = `${load}%`;
    loadingText.style.opacity = scale(load, 0, 100, 1, 0);
    bg.style.filter = `blur(${scale(load, 0, 100, 20, 0)}px)`
  }

  const scale = (num, in_min, in_max, out_min, out_max) => {
    return (num - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
  }
}

```

Menu Page

```

<table>

<tr>
  <td>

    <div id = "menu1">

      <h2>Classics</h2>

      <button class = "ripple">
        <section>
          
          <h3>Cheese Pizza</h3>
          <footer>Plain cheese pizza hand tossed with high quality organic ingredients. Always made fresh.</footer>
          <p class="price"> Price: $15.00</p>
          <p>Click for Ripple Effect</p>
        </section>

      </button>

      <button class = "ripple">

        <section>
          
          <h3>Pepperoni Pizza</h3>
          <footer>Cheese pizza generously topped with succulent pepperoni. Always made fresh.</footer>
          <p class="price"> Price: $20.00</p>
          <p>Click for Ripple Effect</p>
        </section>

      </button>

      <button class = "ripple">

        <section>
          
          <h3>Vegetable Pizza</h3>
          <footer>Cheese pizza topped with organic red and green peppers, black olives, and onions. Always made fresh.</footer>
          <p class="price"> Price: $17.50</p>
          <p>Click for Ripple Effect</p>
        </section>

      </button>

    <button class = "ripple">

      <section>
        
        <h3>Margherita</h3>
        <footer>Classic Pizza Margherita with our authentic Italian mozzarella cheese and our signature tomato sauce made from the highest quality, organic tomatos,
        <p class="price"> Price: $19.00</p>
        <p>Click for Ripple Effect</p>
      </section>

    </button>

    <button class = "ripple">

      <section>
        
        <h3>White Cheese Pizza</h3>
        <footer>Made from only white cheeses of the highest quality. Always made fresh.</footer>
        <p class="price"> Price: $15.00</p>
        <p>Click for Ripple Effect</p>
      </section>

    </button>

  </div>

</td>

```

28. Above is the html file for our menu page. We represented the menu as a table with each row representing the category of each item on the menu. The three categories we chose were Classics, Hot Picks, and Beverages. Each item itself is a section within a button, including the item's image, name, and description within. The snippet above is the html that handles the Classics category.

```
<tr>

  <td>

    <div id = "menu2">

      <h2>Hot Picks</h2>

      <button class = "ripple">

        <section>
          
          <h3>Hawaiian Pizza</h3>
          <footer>Tropical-themed pizza topped with bacon and pineapple. Always made fresh.</footer>
          <p class="price"> Price: $18.00</p>
          <p>Click for Ripple Effect</p>
        </section>

      </button>

      <button class = "ripple">

        <section>
          
          <h3>Jalapeno Pizza</h3>
          <footer>Our classic cheese pizza topped with jalapenos for an extra kick of heat. Always made fresh.</footer>
          <p class="price"> Price: $20.00</p>
          <p>Click for Ripple Effect</p>
        </section>

      </button>

    </div>

  </td>

</tr>
```

29. The snippet above is the html that handles the Hot Picks category.


```

<tr>

<td>

<div id="menu3">

<h2>Beverages</h2>

<button class ="ripple">

<section>

<h3>Coca-Cola</h3>
<p class="price"> Price: $2.50</p>
<p>Click for Ripple Effect</p>
</section>

</button>

<button class ="ripple">

<section>

<h3>Sprite</h3>
<p class="price"> Price: $3.00</p>
<p>Click for Ripple Effect</p>
</section>

</button>

<button class ="ripple">

<section>

<h3>Water</h3>
<p class="price"> Price: $2.00</p>
<p>Click for Ripple Effect</p>
</section>

</button>

</div>

</td>

</tr>

```

30. The snippet above is the html that handles the Beverages category.

```
body{
  background-color: papayawhip;
}

h1 {
  background-color: rgb(215, 0, 64);
  color: white;
  text-align: center;
}

h2{
  background-color: rgb(215, 0, 64);
  color: white;
  text-align: center;
}

.price{
  background-color: rgb(215, 0, 64);
  color: white;
  font-size: 1.5em;
}

#footer2{
  background-color: rgb(215, 0, 64);
  color: white;
  text-align: center;
  height: 5em;
}

#firstFooter2 {
  padding-top: 1em;
}

#container2{
  background-color: rgb(215, 0, 64);
  display: flex;
  height: 7em;
}

#logo2{
  flex:1;
}

#navigation-bar2{
  flex:2;
}
```

31. Above is the css for our menu page. The snippet above takes care of styling the headers, pricing, footers, logo, and navigation bar.

```

#navigation-bar2 nav ul li{
  list-style-type: none;
  display: inline-block;
  padding: 1em 1em;
}

#navigation-bar2 nav ul li a{
  color: white;
  text-decoration: none;
  font-weight: bold;
  font-size: 150%;
}

#navigation-bar2 nav ul li a:hover{
  background: black;
}

#logo2 img{
  width: 50%;
  height: 100%;
}

section img{
  width: 200px;
  display: block;
  margin: 0px auto;
  border: 1px solid gray;
  box-shadow: 0px 0px 10px black;
  box-sizing: border-box;
}

section {
  width: 250px;
  height: 500px;
  text-align: center;
  background: rgb(253, 254, 255);
  box-shadow: inset 0px 0px 20px black;
  padding-top: 20px;
  float: left;
  box-sizing: border-box;
  margin: 10px;
}

```

32. The snippet above takes care of the styling of lists nested within the nav bar and most importantly the styling of the sections, which wrap the content of each item on the menu.

```

button{
  overflow: hidden;
  position: relative;
  background-color: rgb(215, 0, 64);
}

button:focus{
  outline: none;
}

button .circle{
  position: absolute;
  background-color: white;
  width: 600px;
  height: 600px;
  border-radius: 50%;
  transform: translate(-50%, -50%) scale(0);
  animation: scale .5s ease-out;
}

@keyframes scale{
  to{
    transform: translate(-50%, -50%) scale(3);
    opacity: 0;
  }
}

```

33. Lastly, the snippet above is the final portion of the css for our menu, containing the styling for the buttons (each menu item) and handles how the ripple effect looks visually. The last part of this snippet handles the actual “ripple” effect, making the circle scale to a higher number and then fade out because of the opacity.

```

const buttons = document.querySelectorAll('.ripple')

buttons.forEach(button => {
  button.addEventListener('click', function(e) {
    const x = e.clientX
    const y = e.clientY

    const buttonTop = e.target.offsetTop
    const buttonLeft = e.target.offsetLeft

    const xInside = x - buttonLeft
    const yInside = y - buttonTop

    const circle = document.createElement('span')
    circle.classList.add('circle')
    circle.style.top = yInside + 'px'
    circle.style.left = xInside + 'px'

    this.appendChild(circle)

    setTimeout(() => circle.remove(), 500)
  })
})

```

34. Finally, this is the javascript for our menu, which handles the event of the user clicking their mouse over the button (items on the menu). Each button gets an event listener that calculates where the mouse click occurred relative to the webpage and the button itself. Finally, it performs the ripple effect with `this.appendChild(circle)`.