Class 3: Building the Menu Page with Category-Wise Display

SESSION OVERVIEW

By the end of this session, students will be able to:

1. Understand the layout requirements for a menu page, focusing on a category-wise display to improve user navigation.
2. Design a visually appealing and responsive menu layout using HTML and CSS that organizes menu items by category.
3. Style category headers and menu items to ensure a clear, consistent appearance across different screen sizes.
4. Apply CSS techniques such as flexbox or grid to align and arrange menu items effectively within each category.
5. Recognize the importance of categorization in user interface design, enhancing the user's browsing and selection experience.

SESSION TOPICS

Building the Menu Page with Category-Wise Display

Understanding Menu Page Layout:

1. **Category Organization:**
    - Learn the purpose of categorizing menu items for an organized, user-friendly experience.
    - Identify categories that suit different types of dishes (e.g., appetizers, main course, beverages) for clear navigation.
2. **Designing a Responsive Layout:**
    - Explore layout approaches like **Flexbox** and **Grid** to structure menu items efficiently within each category.
    - Ensure each category section is responsive to fit various screen sizes and devices.

Styling with CSS:

1. **Category Headers and Item Cards:**
    - Use CSS to style category headers for distinct sectioning.
    - Create visually consistent item cards with features like borders, shadows, or color schemes to make the menu appealing.
2. **Responsive Styling and Media Queries:**

- Learn how to apply media queries to adjust styles, ensuring optimal viewing on desktops, tablets, and mobile devices.
- Customize item layouts to stack or arrange differently based on screen size.

Enhancing User Interaction with CSS Transitions:

Hover Effects and Transitions:

- Add subtle hover effects to items for better user engagement.
- Utilize CSS transitions to create smooth effects that improve the browsing experience without impacting page performance.

Layout of the Menu page:

Let's break down the implementation of a restaurant menu app menu page into the specified subtopics: basic Layout of the Menu page: Navbar, Main Section, and Footer.

1.HTML Basics:

First, let's create the basic HTML structure for our Menu page. HTML structure that defines the overall layout of the Menu page.

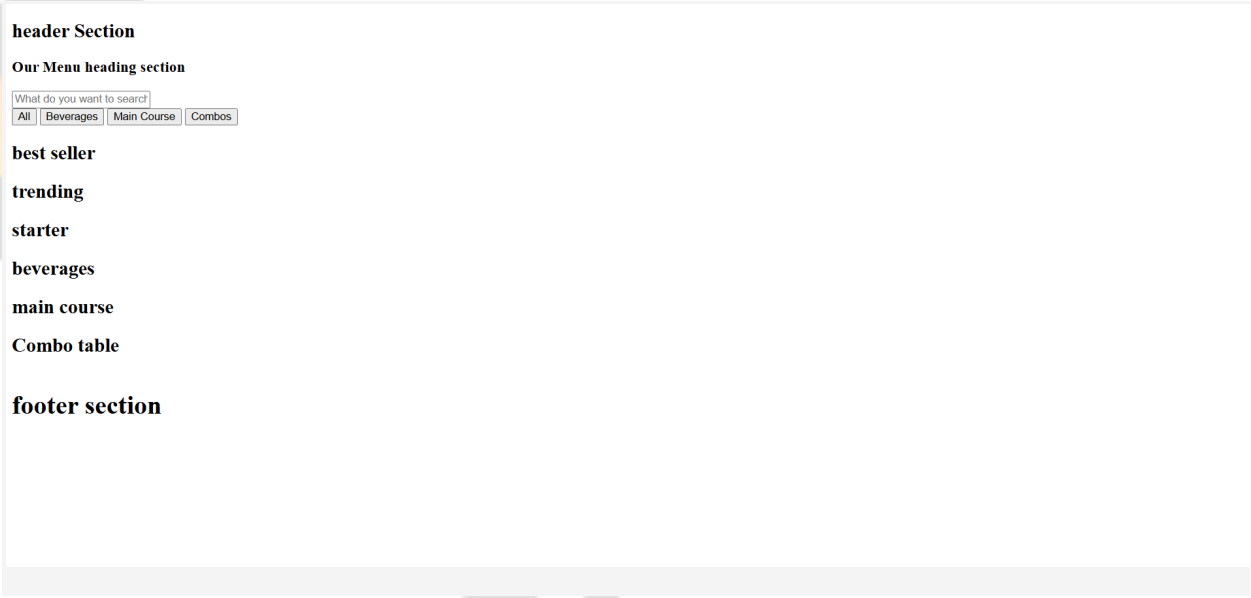
HTML Code

```
<!DOCTYPE html>
<html>
<head>
    <title>Our Menu</title>
</head>
<body>
    <header>
        <nav class="navbar">
         <h1>header Section</h1>
        </nav>
      </header>
    <section class="menu-heading" >
        <h3 class="overlay">Our Menu heading section</h3>
    </section>
    <section class="categories-btn">
        <input class="search-input" placeholder="What do you want to search..."
/>
        <div class="button-group">
```

```
                <button>All</button>
                <button>Beverages</button>
                <button>Main Course</button>
                <button>Combos</button>

        </div>
    </section>
    <section>
<!-- our menu category -->
    <div class="ourmenu-category" id="best-seller">
      <h1>best seller</h1>
           </div>
    <div class="ourmenu-category" id="trending">
      <h1>trending</h1>
           </div>
    <div class="ourmenu-category" id="starter">
      <h1>starter</h1>
         </div>
    <div class="ourmenu-category" id="beverages">
      <h1>beverages</h1>
           </div>
    <div class="ourmenu-category" id="main-course">
      <h1>main course</h1>
           </div>
<!-- combo table -->
<section class="combo-comparison">
    <div class="container">

        <table class="comparison-table">
          <h2>Combo table </h2>
        </table>
    </div>
</section>
    </section>
     <!-- Footer -->
     <footer class="footer">
        <h1>footer section </h1>
      </footer>
```

```
    </body>
  </html>
</body>
</html>
```

Output



Explanation:

- **Header (<header>)**: Contains the navigation bar (<nav>) with a logo , navigation links (<ul>) and Login/signup button.
- **Our menu Section(<section class="menu-heading** display the offers of the restaurant.
- **Main Content Sections (our menu section)**: Divided into multiple categories(bestseller, trending, starters, beverages, main course, combos-table) to organize different parts of the page.
- **Footer (<footer>)**: Includes a simple footer with a shop logo, some quick links, social links and copyright information.

Our Menu Heading Section

Styling and functionality of the our menu heading section using CSS.
Html code :

```
<section class="menu-heading" >
        <h3 class="overlay">Our Menu</h3>
```

```
    </section>
```

CSS code :

```
.menu-heading {
  position: relative;
  height: 400px;
  background-image: url('images.png');
  background-size: cover;
  background-position: center;
  display: flex;
  justify-content: center;
  align-items: center;
  color: white;
  text-align: center;
}

/* Adding an overlay effect */
.menu-heading::before {
  content: "";
  position: absolute;
  top: 0;
  left: 0;
  width: 100%;
  height: 100%;
  background-color: rgba(0, 0, 0, 0.6); /* Black overlay with opacity */
  z-index: 1;
}

/* Text styling for 'Our Menu' */
.menu-heading h3 {
  font-size: 48px;
  letter-spacing: 2px;
  font-weight: 800;
  text-transform: uppercase;
  z-index: 2;
  margin-top: 20%;
}
```

OUTPUT:

Explanation:
.menu-heading: Styles the section with the background image and center-aligned text, using flexbox for layout and positioning. The section includes a hero image for the menu.
.menu-heading::before: Adds a semi-transparent overlay over the background, enhancing text visibility by darkening the image slightly.
.menu-heading h3: Styles the heading text, making it large, bold, and uppercase with added spacing, ensuring it stands out against the background.

Search and filter functionality

Styling and functionality of the Our Menu heading section using CSS.

HTML Code:

```
 <section class="categories-btn">

        <input class="search-input" placeholder="What do you want to search..."
/>
        <div class="button-group">
            <div>
              <button class="active">All</button></a>
            </div>
            <div>
                <a href="#beverages" > <button>Beverages</button></a>

            </div>
            <div>
                <a href="#main-course"><button>Main Course</button></a>
            </div>
            <div>
                <a href="#combos"> <button>Combos</button></a>
            </div>
        </div>
```

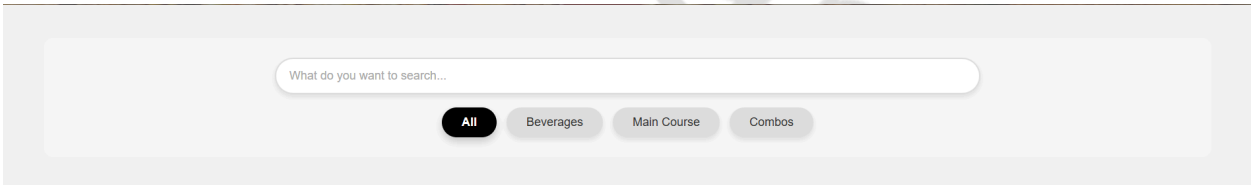
```
    </section>
```

CSS Code:

```
.categories-btn {
  display: flex;
  flex-direction: column;
  align-items: center;
  justify-content: center;
  gap: 20px;
  padding: 30px;
  background-color: #f8f8f8;
  border-radius: 15px;
  /* box-shadow: 0px 8px 16px rgba(0, 0, 0, 0.1); */
  width: 80%;
  margin: 50px auto;
}

/* Button Group Styling */
.button-group {
  display: flex;
  gap: 15px;
  flex-wrap: wrap;
  justify-content: center;
}

/* Button Styling */
.button-group button {
  padding: 12px 30px;
  font-size: 18px;
  border-radius: 30px;
  border: none;
  cursor: pointer;
  background-color: #e0e0e0;
  color: #333;
  transition: all 0.3s ease-in-out;
  box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
  font-weight: 500;
}

/* Hover and Active State */
.button-group button:hover {
  background-color: #000000;
  color: white;
  /* transform: scale(1.1); */
  box-shadow: 0px 6px 12px rgba(0, 0, 0, 0.15);
}
```

```css
.button-group button.active {
  background-color: #000000;
  color: white;
  font-weight: bold;
  box-shadow: 0px 6px 12px rgba(0, 0, 0, 0.2);
}

/* Search Input Styling */
.search-input {
  width: 60%;
  padding: 15px 20px;
  font-size: 18px;
  border: 2px solid #e0e0e0;
  border-radius: 30px;
  outline: none;
  transition: all 0.3s ease-in-out;
  box-shadow: 0px 4px 8px rgba(0, 0, 0, 0.1);
}

/* Focus and Hover Effects for Input */
.search-input:focus {
  border-color: #f0a500;
  box-shadow: 0px 6px 12px rgba(240, 165, 0, 0.2);
}

.search-input::placeholder {
  color: #aaa;
  font-weight: 400;
}
```

OUTPUT :



Explanation:

.categories-btn: Styles the main container for the category section, aligning content vertically and centering it. It includes padding, background color, and a rounded border for a clean look.
.button-group: Arranges the buttons in a horizontal layout with space between them. It wraps the buttons if the screen is too narrow to fit them all in one row.
.button-group button: Provides button styling with padding, rounded corners, and a subtle shadow. The buttons have a neutral background color and change color when hovered.

.button-group button
: Adds hover effects to buttons, changing the background to black, text to white, and enhancing the shadow for a raised effect.
.button-group button.active: Highlights the currently active button with a bold style, black background, white text, and a stronger shadow.
.search-input: Styles the search bar with padding, a rounded border, and shadow. It changes border color and shadow when focused to indicate interaction.
.search-input::placeholder: Customizes the placeholder text style, making it lighter for better readability.

Our menu categories-wise Section:

Styling and functionality of our menu categories-wise section using CSS.

HTML Code:

```
        <div class="ourmenu-category" id="best-seller">
            <div class="ourmenu-category-header">
              <h3>Best Seller</h3>
            </div>
            <div class="ourmenu-items">
              <div class="ourmenu-card">
                <img src="https://placehold.co/200x100?text=BestSeller" alt="Best
Seller Image">
                <div class="menu-card-content">
                  <h4>Best Seller Item 1</h4>
                  <p>Description of the best seller item.</p>
                  <span>Price: <strike class="strike-price">$10.99</strike> $8.99
</span>
                </div>
                <div class="add-to-cart-btn">
                  <button class="cta-button">Add to Cart</button>
                </div>
              </div>
              <div class="ourmenu-card">
              <img src="https://placehold.co/200x100?text=BestSeller" alt="Best
Seller Image">
              <div class="menu-card-content">
                <h4>Best Seller Item 2</h4>
                <p>Description of the best seller item.</p>
                <span>Price: <strike class="strike-price">$10.99</strike>
$8.99</span>
              </div>
              <div class="add-to-cart-btn">
                <button class="cta-button">Add to Cart</button>
              </div>
            </div>
            <div class="ourmenu-card">
```

```
            <img src="https://placehold.co/200x100?text=BestSeller" alt="Best
Seller Image">
            <div class="menu-card-content">
              <h4>Best Seller Item 3</h4>
              <p>Description of the best seller item.</p>
              <span>Price: <strike class="strike-price">$10.99</strike>
$8.99</span>
            </div>
            <div class="add-to-cart-btn">
              <button class="cta-button">Add to Cart</button>
            </div>
          </div>
          <div class="ourmenu-card">
              <img src="https://placehold.co/200x100?text=BestSeller" alt="Best
Seller Image">
              <div class="menu-card-content">
                <h4>Best Seller Item 3</h4>
                <p>Description of the best seller item.</p>
                <span>Price: <strike class="strike-price">$10.99</strike>
$8.99</span>
              </div>
              <div class="add-to-cart-btn">
                <button class="cta-button">Add to Cart</button>
              </div>
            </div>
            <div class="ourmenu-card">
              <img src="https://placehold.co/200x100?text=BestSeller" alt="Best
Seller Image">
              <div class="menu-card-content">
                <h4>Best Seller Item 3</h4>
                <p>Description of the best seller item.</p>
                <span>Price: <strike class="strike-price">$10.99</strike>
$8.99</span>
              </div>
              <div class="add-to-cart-btn">
                <button class="cta-button">Add to Cart</button>
              </div>
            </div>
            <div class="ourmenu-card">
              <img src="https://placehold.co/200x100?text=BestSeller" alt="Best
Seller Image">
              <div class="menu-card-content">
                <h4>Best Seller Item 3</h4>
                <p>Description of the best seller item.</p>
                <span>Price: <strike class="strike-price">$10.99</strike>
$8.99</span>
              </div>
              <div class="add-to-cart-btn">
                <button class="cta-button">Add to Cart</button>
              </div>
            </div>
          </div>
```

```
        </div>
```

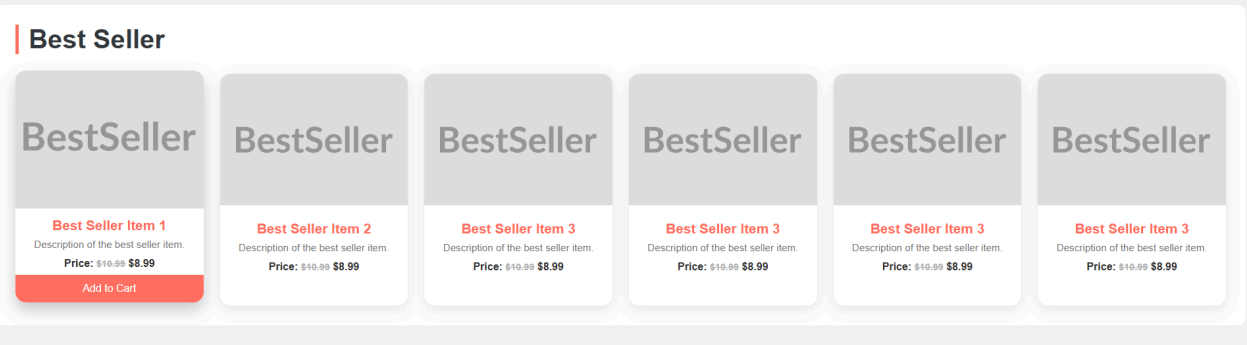
CSS Code:

```
/* Menu Category Styles */

.ourmenu-category {
/* max-width: 1200px; */
margin: 40px auto;
padding: 30px;
background-color: #ffffff;
border-radius: 12px;
/* box-shadow: 0 10px 20px rgba(0, 0, 0, 0.1); */
text-align: center;
}

.ourmenu-category-header {
display: flex;
/* justify-content: center; */
align-items: center;
margin-bottom: 30px;
}

.ourmenu-category h3 {
font-size: 2.5rem;
color: #343a40;
margin: 0;
border-left: 5px solid #ff6f61;
padding-left: 15px;
}

.ourmenu-items {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(240px, 1fr));
gap: 25px;
}

.ourmenu-card {
background-color: #fff;
border-radius: 16px;
box-shadow: 0 6px 20px rgba(0, 0, 0, 0.1);
transition: transform 0.3s ease, box-shadow 0.3s ease;
position: relative;
overflow: hidden;
padding-bottom: 10px;
}

.ourmenu-card:hover {
```

```
transform: translateY(-5px);
box-shadow: 0 12px 24px rgba(0, 0, 0, 0.2);
}

.card-image-container {
overflow: hidden;
border-radius: 16px 16px 0 0;
max-height: 200px;
}

.ourmenu-card img {
width: 100%;
height: 200px;
object-fit: cover;
transition: transform 0.4s ease;
}

.ourmenu-card:hover img {
transform: scale(1.1);
}

.menu-card-content {
padding: 20px;
}

.menu-card-content h4 {
font-size: 1.3rem;
color: #ff6f61;
margin: 0 0 8px;
}

.menu-card-content p {
font-size: 0.95rem;
color: #777;
margin: 0 0 12px;
}

.price {
font-size: 1.1rem;
font-weight: bold;
color: #333;
}

.strike-price {
color: #b0b0b0;
font-size: 0.9rem;
}

/* Updated CTA Button Hover Effect */
.add-to-cart-btn {
z-index: 1; /* Ensure it stays on top */
```

```
margin-top: 20px;/* Ensure it stays on top */
}
.cta-button {
width: 100%;
padding: 12px;
background-color: #ff6f61;
color: #fff;
font-size: 1rem;
border: none;
border-radius: 0 0 16px 16px;
cursor: pointer;
transform: translateY(20px);
opacity: 0;
transition: transform 0.4s ease, opacity 0.4s ease;
position: absolute;
bottom: 0;
left: 0;
}
.ourmenu-card:hover .cta-button {
transform: translateY(0);
opacity: 1;
}

.cta-button:hover {
background-color: #e55a4f;
}
```

OUTPUT:



Explanation:

General Structure

- **`.ourmenu-category` and `header`**: Provides layout and visual structure for the menu section, including a background color, padding, and border radius.

- **`.ourmenu-items`**: Uses a grid layout to arrange menu items responsively with gaps between them.
- **`.ourmenu-card`**: Styles each card with a background, border-radius, and shadow. Transitions create a subtle hover effect (`transform`, `box-shadow`).
- **`.ourmenu-card img`**: Contains the image in the card, adjusting its size and applying a zoom effect on hover.

Button and Hover Effects

- **`.cta-button`**: Initially hidden with `opacity: 0` and positioned off-screen with `transform: translateY(20px)`. Transitions make it smoothly appear on hover.
- **Hover Styles**: `.ourmenu-card:hover` raises the card and shows the button by resetting its `transform` and `opacity`.
- **`.cta-button:hover`** changes its background color for visual feedback.

Key CSS Properties

- **`z-index: 1`**: Ensures `.add-to-cart-btn` appears above other elements.
- **`transition`**: Smoothly animates `transform` and `opacity` over `0.4s` with an `ease` effect for the `.cta-button`.

1. `z-index` Example Explanation

In your code, the `.add-to-cart-btn` class has:

```
.add-to-cart-btn {
  z-index: 1; /* Ensures it stays on top */
}
```

- **Purpose**: This ensures that the "Add to Cart" button appears above any other elements that might overlap with it. If there were an element with a lower `z-index` or none specified, it would be rendered behind this button.
- **Practical Example**: If an image inside `.ourmenu-card` had a `z-index: 0`, the `.add-to-cart-btn` would stay on top, ensuring the button is clickable and not hidden behind the image.

2. `transition` Example Explanation

Your code snippet for the `.cta-button`:

```css
.cta-button {
  transition: transform 0.4s ease, opacity 0.4s ease;
}
```

- **Purpose**: This creates a smooth animation when the `transform` and `opacity` properties change, such as when the card is hovered over.

Effect in Action:

```css
.ourmenu-card:hover .cta-button {
  transform: translateY(0);
  opacity: 1;
}
```

- **Before Hover**: The button is shifted down by `translateY(20px)` and has `opacity: 0` (invisible).
- **On Hover**: It animates to move up (`translateY(0)`) and becomes fully visible (`opacity: 1`). The transition duration of `0.4s` ensures this change happens smoothly

Note:

Remember that above provided html structure is for single category i.e for best seller category but you can replicate this structure to create more categories like (trending, starters, beverages, main course) or make your own html structure and add other more categories as needed.

Combo-Table

Styling and functionality of our Combo table section using CSS.

HTML CODE :

```html
<section class="combo-comparison">
    <div class="container">
        <h2>Combo </h2>
        <table class="comparison-table">
            <thead>
                <tr>
```

```
                    <th rowspan="2">Combo Size</th>
                    <th rowspan="2">Items</th>
                    <th colspan="2">Price</th>
                    <th rowspan="2">Add to Cart</th>
                </tr>
                <tr>
                    <th>Original</th>
                    <th>Discounted</th>
                </tr>
            </thead>
            <tbody>
                <!-- Small Combo -->

                <tr>
                    <td>Small</td>
                    <td>
                        <div class="item">
                            <!-- <img
src="https://placehold.co/200x100?text=item+1" alt="Item 1" /> -->
                            <p>Item Name 1</p>
                        </div>
                        <div class="item">
                            <!-- <img
src="https://placehold.co/200x100?text=item+2" alt="Item 2" /> -->
                            <p>Item Name 2</p>
                        </div>
                        <div class="item">
                            <!-- <img
src="https://placehold.co/200x100?text=item+3" alt="Item 3" /> -->
                            <p>Item Name 3</p>
                        </div>
                    </td>
                    <td><strike>$30</strike></td>
                    <td>$25</td>
                    <td><button class="cta-btn">Add to Cart</button></td>
                </tr>

                <!-- Medium Combo -->
                <tr>
                    <td>Medium</td>
                    <td>
                        <div class="item">
                            <!-- <img
src="https://placehold.co/200x100?text=item+1" alt="Item 1" /> -->
                            <p>Item Name 1</p>
                        </div>
                        <div class="item">
                            <!-- <img
src="https://placehold.co/200x100?text=item+2" alt="Item 2" /> -->
                            <p>Item Name 2</p>
                        </div>
                        <div class="item">
```

```
                            <!-- <img
src="https://placehold.co/200x100?text=item+3" alt="Item 3" /> -->
                            <p>Item Name 3</p>
                        </div>
                    </td>
                    <td><strike>$40</strike></td>
                    <td>$35</td>
                    <td><button class="cta-btn">Add to Cart</button></td>
                </tr>

                <!-- Large Combo -->
                <tr>
                    <td>Large</td>
                    <td>
                        <div class="item">
                            <!-- <img
src="https://placehold.co/200x100?text=item+1" alt="Item 1" /> -->
                            <p>Item Name 1</p>
                        </div>
                        <div class="item">
                            <!-- <img
src="https://placehold.co/200x100?text=item+2" alt="Item 2" /> -->
                            <p>Item Name 2</p>
                        </div>
                        <div class="item">
                            <!-- <img
src="https://placehold.co/200x100?text=item+3" alt="Item 3" /> -->
                            <p>Item Name 3</p>
                        </div>
                    </td>
                    <td><strike>$50</strike></td>
                    <td>$45</td>
                    <td><button class="cta-btn">Add to Cart</button></td>
                </tr>
            </tbody>
        </table>
    </div>
</section>
```

CSS STYLE:

```
/* combos */


/* Basic styling for the table */
.combo-comparison {
  font-family: Arial, sans-serif;
  color: #333;
  margin-bottom: 30px;
  padding: 30px;
```

```
}

.combo-comparison h2 {
  /* text-align: center; */
  margin-bottom: 20px;
  font-size: 2.5rem;
color: #343a40;
/* margin: 0; */
border-left: 5px solid #ff6f61;
padding-left: 15px;
}

.comparison-table {
  width: 100%;
  border-collapse: collapse;
  margin: 0 auto;


}

.comparison-table th, .comparison-table td {
  padding: 15px;
  text-align: center;
  border: 1px solid #474646 ;


}

.comparison-table th {
  background-color: #ff6f61;
  font-weight: bold;
  border: 1px solid #474646 ;
  color: white;
}

.comparison-table tbody tr:nth-child(even) {
  background-color: #f9f9f9;
}

/* Style for the item details within cells */
.comparison-table .item {
  display: flex;
  align-items: center;
  justify-content: center;
  flex-direction: column;
  gap: 5px;
}
.item{
  margin-bottom: 10px;
}

.comparison-table .item img {
```

```
  width: 60px;
  height: auto;
  border-radius: 4px;
}

.comparison-table .item p {
  margin: 0;
  font-size: 14px;
  color: #666;
}

/* Styling for strike-through original price and discounted price */
.comparison-table td strike {
  color: #999;
}

.comparison-table td {
  font-size: 20px;
}

.comparison-table td button.cta-btn {
  background-color: #ff6f61;
  color: #fff;
  padding: 8px 12px;
  border: none;
  border-radius: 4px;
  cursor: pointer;
  font-size: 14px;
}

.comparison-table td button.cta-btn:hover {
  background-color: #e55a4f;
}
```

Output

Combo

Combo Size	Items	Price		Add to Cart
		Original	Discounted	
Small	Item Name 1 Item Name 2 Item Name 3	~~$30~~	$25	Add to Cart
Medium	Item Name 1 Item Name 2 Item Name 3	~~$40~~	$35	Add to Cart
Large	Item Name 1 Item Name 2 Item Name 3	~~$50~~	$45	Add to Cart

Explanation:

Container Styling (`.combo-comparison`):

- Sets the basic layout properties like font type (`Arial`), color, margin, and padding to ensure spacing around the table and content clarity.

Table Structure (`.comparison-table`):

- Full-width, centered table with collapsed borders for clean row and column divisions.
- `th` (Table Headers): Styled with a bold, white font and a bright background color (`#ff6f61`), creating a strong visual distinction for header cells.
- `td` (Table Cells): Centered text with consistent padding, providing a clear, readable layout.

Row Alternating Colors:

- `.comparison-table tbody tr:nth-child(even)`: Adds a light gray background color for even rows, improving readability.

Item Details (`.item`):

- A flex container aligning its children vertically for uniform item presentation (e.g., an image with a caption).
- Adds a gap between child elements and a bottom margin for spacing consistency.

Images (`.comparison-table .item img`):

- Thumbnail images have a `60px` width with rounded corners (`border-radius: 4px`), giving a polished look.

Pricing:

- The original price is styled with a `strike` element, using a light gray color to indicate it is discounted.

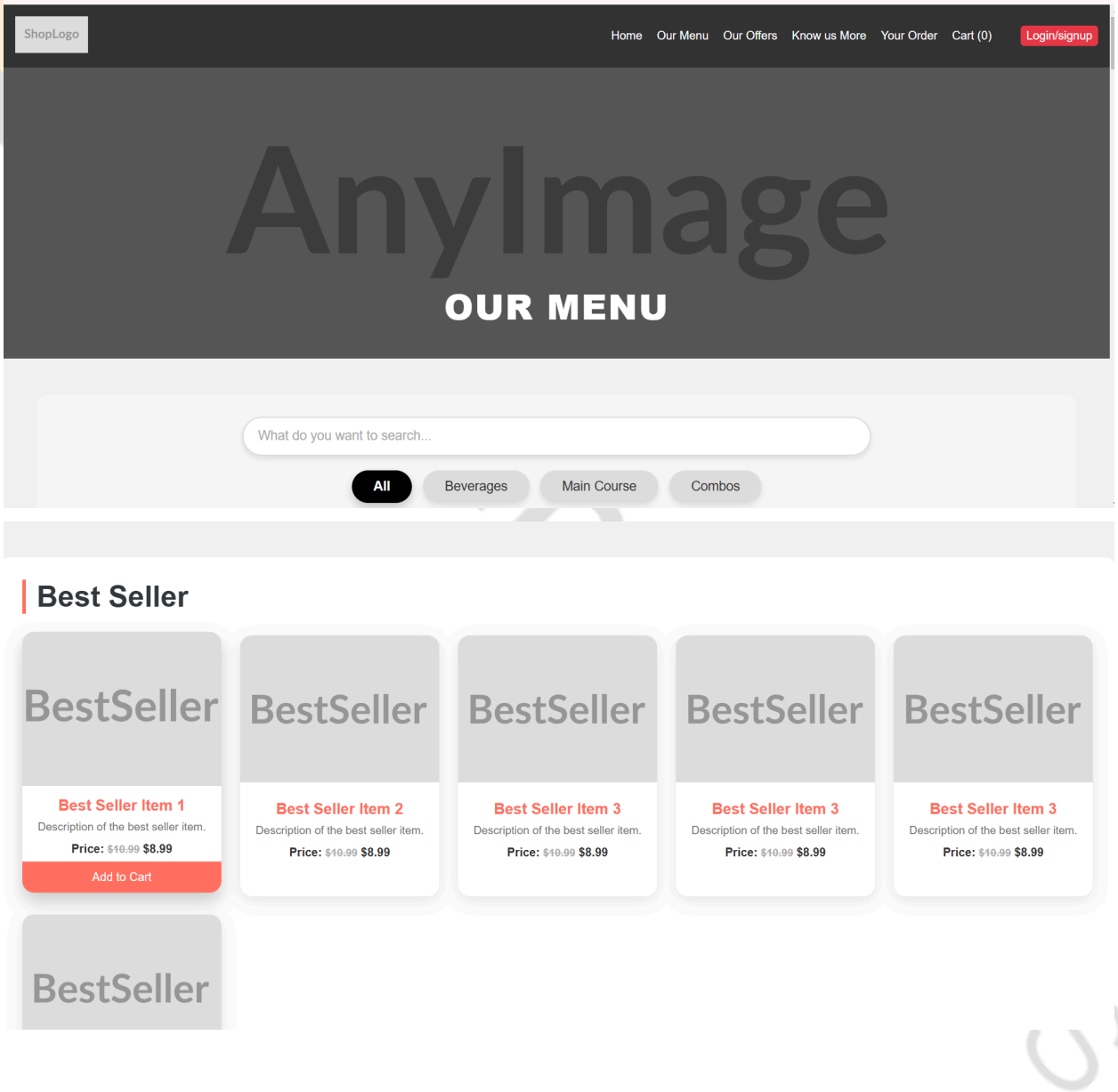
Buttons (`.cta-btn`):

- Styled with a matching background color (`#ff6f61`), white text, and padding for an inviting, clickable appearance.
- Hover Effect: Changes the background to a darker shade (`#e55a4f`), providing feedback when hovered over.

Note:
Now Add the Header section and footer section code to complete your Menu page. Also, include media queries to ensure your page is responsive and adapts well to different screen sizes. This way, you can maintain a consistent design and functionality across various devices.

Final Output:

[Ourmenupage.html]
[Ourmenupagestyle.css]



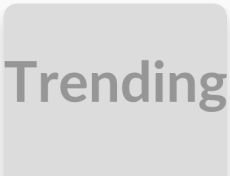
codingninjas

Trending

Trending Item 1

Description of the trending item.

Price: ~~$10.99~~ **$8.99**



Trending Item 2

Description of the trending item.

Price: ~~$10.99~~ **$8.99**

Add to Cart

Trending Item 3

Description of the trending item.

Price: ~~$10.99~~ **$8.99**

Trending Item 4

Description of the trending item.

Price: ~~$10.99~~ **$8.99**

Trending Item 5

Description of the trending item.

Price: ~~$10.99~~ **$8.99**

Starter

Starter Item 1

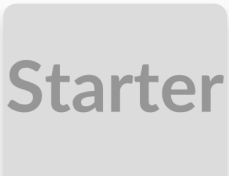
Description of the starter item.

Price: ~~$10.99~~ **$8.99**

Starter Item 2

Description of the starter item.

Price: ~~$10.99~~ **$8.99**



Starter Item 3

Description of the starter item.

Price: ~~$10.99~~ **$8.99**

Add to Cart

Starter Item 4

Description of the starter item.

Price: ~~$10.99~~ **$8.99**

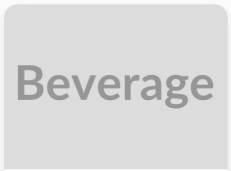
Starter Item 5

Description of the starter item.

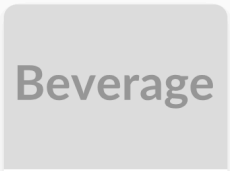
Price: ~~$10.99~~ **$8.99**

codingninjas

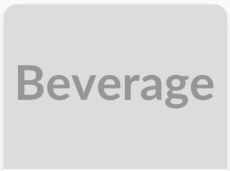
Beverages



Beverage Item 1
Description of the beverage item.
Price: ~~$5.99~~ **$4.99**

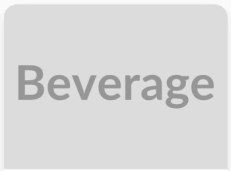


Beverage Item 2
Description of the beverage item.
Price: ~~$5.99~~ **$4.99**

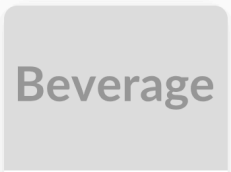


Beverage Item 3
Description of the beverage item.
Price: ~~$5.99~~ **$4.99**

Beverage Item 4
Description of the beverage item.
Price: ~~$5.99~~ **$4.99**
Add to Cart



Beverage Item 5
Description of the beverage item.
Price: ~~$5.99~~ **$4.99**



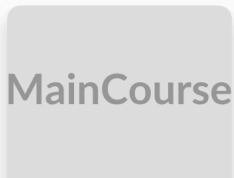
Main Course

Main Course Item 1
Description of the main course item.
Price: ~~$10.99~~ **$8.99**

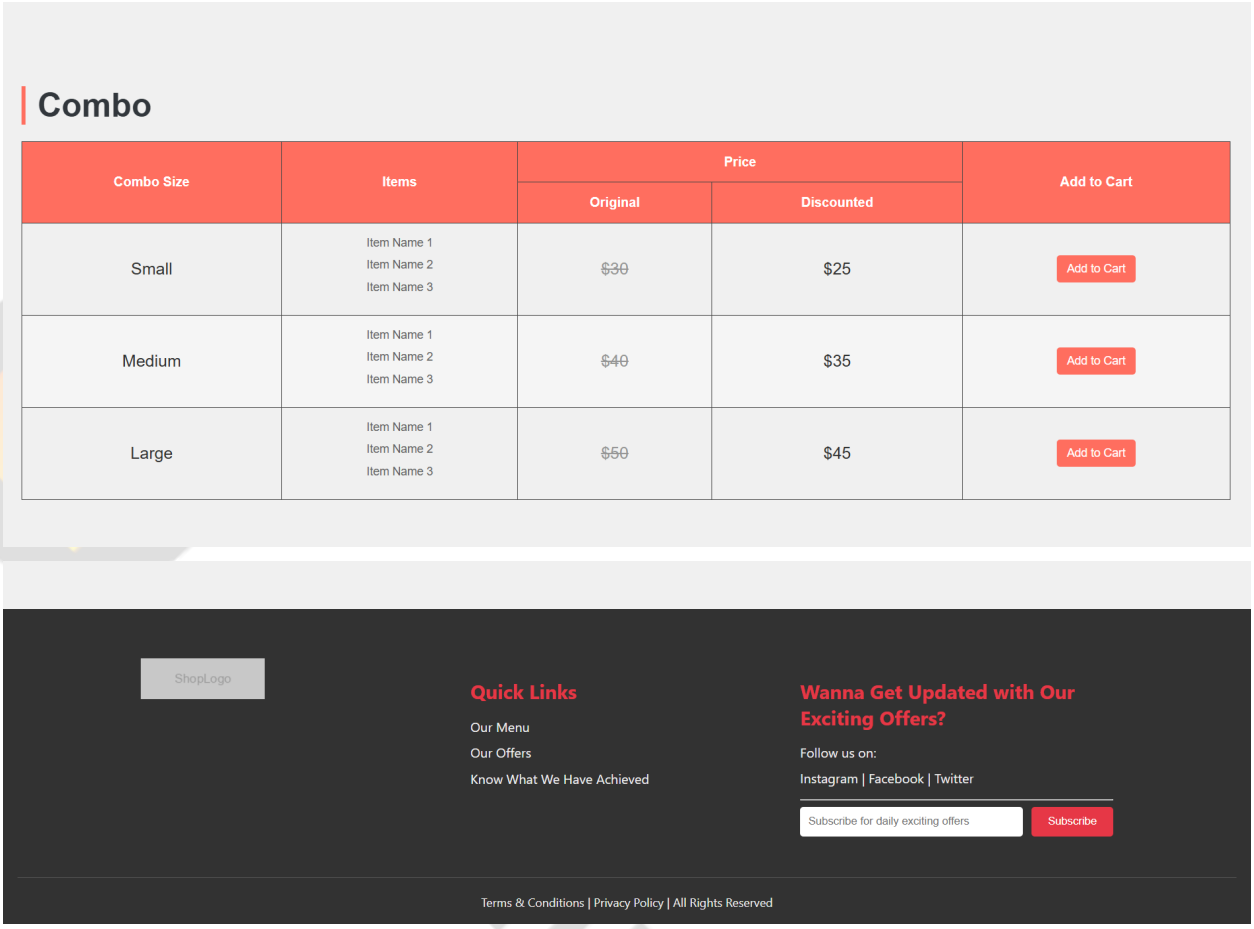
Main Course Item 2
Description of the main course item.
Price: ~~$10.99~~ **$8.99**

Main Course Item 3
Description of the main course item.
Price: ~~$10.99~~ **$8.99**

Main Course Item 4
Description of the main course item.
Price: ~~$10.99~~ **$8.99**



Main Course Item 5
Description of the main course item.
Price: ~~$10.99~~ **$8.99**
Add to Cart



Combo

Combo Size	Items	Price		Add to Cart
		Original	Discounted	
Small	Item Name 1 Item Name 2 Item Name 3	~~$30~~	$25	Add to Cart
Medium	Item Name 1 Item Name 2 Item Name 3	~~$40~~	$35	Add to Cart
Large	Item Name 1 Item Name 2 Item Name 3	~~$50~~	$45	Add to Cart

---

Concept in depths:

CSS Z-Index Property and Transition Effect:

---

1. CSS Z-Index Property

What is Z-Index?

- The **z-index** property in CSS controls the **stacking order** of elements on a webpage. In simpler terms, it determines which elements appear on top of others when they overlap.

How It Works:

- The **z-index** only works on elements that have a **positioning context**, such as `relative`, `absolute`, `fixed`, or `sticky`. By default, elements with the `static` position do not respond to z-index.

Key Points to Remember:

- **Higher z-index values** will push elements **on top** of others with lower values.
- **Negative z-index values** can push elements **behind** others.
- By default, all elements without a z-index are stacked in the order they appear in the HTML, with later elements appearing on top.

Example 1: Basic Z-Index

Let's say we have two div elements that are overlapping. By adjusting their z-index, we control which one appears on top.

```
<div class="box1">Box 1</div>
<div class="box2">Box 2</div>

<style>
  .box1 {
    width: 100px;
    height: 100px;
    background-color: red;
    position: absolute;
    top: 50px;
    left: 50px;
    z-index: 1;
  }

  .box2 {
    width: 100px;
    height: 100px;
    background-color: blue;
    position: absolute;
    top: 100px;
    left: 100px;
    z-index: 2;  /* Box 2 appears on top */
  }
</style>
```

Key Takeaway:

- **Box 2** will be on top of **Box 1** because its z-index is higher.

Example 2: Z-Index with Negative Values

You can use **negative z-index** to push an element behind others:

```
<div class="behind">I am behind!</div>
<div class="front">I am in front!</div>

<style>
  .behind {
    width: 100px;
    height: 100px;
    background-color: yellow;
    position: absolute;
    z-index: -1;  /* Behind the other element */
  }

  .front {
    width: 100px;
    height: 100px;
    background-color: green;
    position: absolute;
    z-index: 1;  /* In front of the other element */
  }
</style>
```

Key Takeaway:

- **Behind** has a negative z-index, so it stays behind the **front** element.

---

2. CSS Transition Effect

codingninjas

What is a CSS Transition?

- A **transition** is a way to make smooth animations when CSS properties change. For example, you can make an element **fade in**, **grow in size**, or **move** when the user interacts with it (e.g., hovering).

How It Works:

- To create a transition, you specify:
  - The **property** you want to animate (e.g., `background-color`, `height`).
  - The **duration** of the animation (e.g., `2s` for 2 seconds).
  - Optionally, you can also specify **timing functions** (how the animation accelerates or decelerates) and **delays**.

Example 1: Simple Hover Transition

In this example, the background color of a button changes smoothly when you hover over it.

```
<button class="hoverButton">Hover me!</button>
<style>
  .hoverButton {
    padding: 10px 20px;
    background-color: blue;
    color: white;
    border: none;
    cursor: pointer;
    transition: background-color 0.3s ease;  /* Transition on background color */
  }

  .hoverButton:hover {
    background-color: red;  /* Change color on hover */
  }
</style>
```

Key Takeaway:

- The **background color** of the button will transition smoothly from blue to red over 0.3 seconds when the user hovers over it.

Example 2: Changing Size on Hover

You can also change an element's size when the user interacts with it.

```
<div class="box">Hover to resize!</div>

<style>
  .box {
    width: 100px;
    height: 100px;
    background-color: orange;
    transition: transform 0.5s ease;  /* Transition on transformation */
  }

  .box:hover {
    transform: scale(1.5);  /* Grow the box to 1.5 times its size */
  }
</style>
```

Key Takeaway:

- The **box** will scale up smoothly when the user hovers over it. The transition takes 0.5 seconds.

Example 3: Opacity Fade In

Another common effect is fading elements in and out using opacity.

```
<div class="fadeBox">Fade In and Out!</div>

<style>
  .fadeBox {
    width: 150px;
    height: 150px;
    background-color: purple;
    opacity: 0;  /* Initially hidden */
    transition: opacity 2s ease-in-out;  /* Transition on opacity */
  }

  .fadeBox:hover {
    opacity: 1;  /* Fully visible on hover */
  }
```

```
</style>
```

Key Takeaway:

- The **fadeBox** element starts off invisible (`opacity: 0`) and becomes fully visible (`opacity: 1`) when hovered.

Important Notes on Transitions and Z-Index:

1. **Using Z-Index with Transitions:**
    - You can combine **z-index** and **transitions** to animate an element's stacking order.

Example: Animate a box that moves forward in the stack when clicked.

```
<div class="stackBox">Click me!</div>

<style>
  .stackBox {
    width: 100px;
    height: 100px;
    background-color: pink;
    position: relative;
    z-index: 1;
    transition: z-index 0.3s ease;
  }

  .stackBox:active {
    z-index: 10;  /* Move to top on click */
  }
</style>
```

2. **Multiple Transitions:**

You can animate multiple properties at once using **comma separation**.

```
.box {
  transition: transform 1s, background-color 1s;  /* Animate both properties */
}
```

Follow tech blogs like Smashing Magazine, CSS-Tricks, and Dev.to to stay current with industry trends