



Interactive Recipe Pages



By: Madison Gibbons, Tijuan Johnson, and
Edward Luong



Delicious Delicacies

We created a small recipe book with six recipe pages. There's a homepage with six recipe cards that lead to personally designed pages, complete with ingredients, step-by-step instructions, and relevant pictures. We did all of this with only HTML and CSS.

Homepage Overview

On the home page, there's a header with a logo (courtesy of Google) and the page title. Below it, there's a rectangle with the six recipe cards, all links to the respective recipes. Each card has an image and it flips when the user hovers over it, with a title and description on the back.

The main issue on the homepage was the CSS based filtering system. I explain it in the next few slides.

Homepage Filters

The problem I was repeatedly having was the use of the combinators. Combinators are used to connect different sections of code. Since the div that holds the checkmarks isn't a direct sibling of the recipe cards, I was having trouble finding a method that works. After a while, I finally found the `:has` pseudo class.

There was another issue of the pictures wrapping when the screen is smaller, but that issue was resolved with flex-wrap.

Homepage Filters

The `:has` pseudo class basically checks if the given id or class has the specified element, shown on the right. I connected this to the recipe pages on the sibling div, main.

```
/* The :has class checks if the inner has the relevant information
I used it to check if the checkboxes with the id had been checked and then used
the sibling combinator to affect the recipe cards in the main div */

.checks:has(#dessert:checked) ~ .main .recipe:not(.dessert){
  visibility: hidden;
  opacity: 0;
  transition: 0.35s ease-out; /* Makes the div disappear smoothly */
}

.checks:has(#no-dairy:checked) ~ .main .recipe:not(.no-dairy){
  visibility: hidden;
  opacity: 0;
  transition: 0.35s ease-out;
}
```


Homepage Recipe Cards

Each card consists of an overall recipe div that defines it as a recipe and which filter applies to it. Inside of that, there's an inner div to deal with location of the card and front and back divs. In the code, the most important parts are the :hover pseudo class, "transform-style: preserve-3d", and "backface-visibility: hidden." To my understanding, the transform-style makes it so the div can exist on a 3D plane. This would make it able to have two sides. Backface-visibility makes it so when we flip the card, what we flipped originally isn't visible.

TLDR: I made the cards able to flip by hiding the back of it and animating the flip.

Homepage Recipe Cards

```
.recipe:hover .inner{  
  transform: rotateY(180deg);  
}
```

You, 30 minutes ago • Uncommitted changes

```
.inner{  
  position: relative;  
  transition: transform 0.8s;  
  transform-style: preserve-3d; /*necessary to have two sides*/  
  /*To my understanding, transform style makes it so that the two sides of the  
  card can exist on one page, instead of just the front*/  
  width: 379px;  
  height: 100%;  
  margin:auto;  
}
```

```
.frontcard, .backcard{  
  position: fixed;  
  backface-visibility: hidden;  
  /*So we can have two sides*/  
}
```

Madison's Recipe Pages Overview

On my recipe pages I did Teriyaki Chicken Fried Rice and Loaded Tacos / Taco Bowls. They look the same besides the colors and font sizes. They have a header with the logo and the title of the recipes. Beneath that, they have an image and an interactable ingredient checklist on the left and the directions on the right. T

here's a small button with a nutrition content dropdown. Always at the bottom of the page, there's a related images section that is able to scroll if needed with a couple of pictures relevant to the recipe.

Madison's Recipe Pages Layout

A majority of the page layout has to do with display: flex and creating many divs to move around. The main issue I had with these pages was making the bottom div stay at the bottom of the page, but also impact the rest of the code (so nothing overlaps).

After much googling, I found “flex-direction” which allowed me to change the layout of the entire page to a column. That way, I could make the bottom div stay at the bottom of the page, without making it absolute or fixed.

Madison's Recipe Pages Code

```
html, body{
  margin: 0;
  height: 100%;
  overflow: auto; /*Automatic scrolling when content is too long for page*/
}
body{
  display: flex;
  flex-direction: column; /*bottom section can stay at the bottom*/
  /* Flex direction specifies the direction*/
  background: url(teripics/terivariation5.png) no-repeat center;
  background-size: cover;
}
```

```
.all{
  margin-top: 20px;
  display: flex;
  flex-direction: column;
  /*my inner divs can be on top of each other*/
  flex-grow: 1; /*takes up the entire page horizontally*/
}

.maincontent{
  /* border: 5px solid black; */
  display: flex;
  flex-grow: 1;
}

.right{
  flex-basis: 70%;
  /* border: 2px solid purple; */
}

.left{
  flex-basis: 30%;
  /* border: 2px solid red; */
  position: relative;
}

.bottom{
  /* border: 2px solid pink; */
  padding-top: 10px;
  text-align: center;
}
```

Edward Recipe Overview and Layout

- To keep the overall uniform format, my pages were similar to the others in terms of formatting.
- I went with the simple design and kept a straight cut format with each element in their respective positions. The background was set with solid colors matching the recipe in the page.
- For the relevant images at the bottom, I used padding and margining to make sure that the images fit in the page and can enlarge without clipping into the sides of the window.
- As for the steps, I was able to modify the appearance of the listed points by using list-style-type.

Edward Recipe Functionality

- For the ingredients list, I used a checkbox label to be able to visually alter the text of each ingredient by crossing them out and change the background to a darker color.
- The ingredients list also has the :hover function so that the user can highlight each ingredient when hovering the cursor over each one.
- The relevant images at the bottom also has the :hover function so that each one can be enlarged to see each one with more detail.
- I also included a dropdown where the user can hover over a button and reveal the nutrition facts regarding the recipe shown.

```

<div class="ingredientlist">
  <h2>Main</h2>
  <label class="ingredient">
    <input type="checkbox">
    <span class="checkmark"></span>
    <span class="selected">1 lb fillet salmon</span>
  </label>
  <label class="ingredient">
    <input type="checkbox">
    <span class="checkmark"></span>
    <span class="selected">1 oz ginger</span>
  </label>
  <label class="ingredient">
    <input type="checkbox">
    <span class="checkmark"></span>
    <span class="selected">2 oz green onion</span>
  </label>

```

Pieces of Edward's Code

```

/*Steps*/

.steptop {
  text-align: center;
  text-decoration: underline;
  border-top: solid;
  border-left: solid;
  border-right: solid;
  border-color: #909090;
  padding-bottom: 1px;
  border-width: 2px;
}

.steplist {
  border: 2px solid #909090;
  padding: 2em;
  line-height: 2;
}

.stephead {
  list-style-type: upper-roman;
}

.stephead > li {
  text-decoration: underline;
  font-weight: bold;
  font-size: 21px;
}

```

TJ Recipe Overview

- ❑ For my recipes, I decided to with brownies and cinnamon rolls.
- ❑ The themes for my webpages were intentional to try and capture the essence of the dessert I was doing.
- ❑ Each page has a background image that is a pattern to fit the theme of the recipe
- ❑ Both pages have the same layout but they have different elements that tie them apart from each other (color, images, patterns)
- ❑ Each page has an interactable ingredients list with checkboxes to indicate that you have used/taken note of a specific ingredient.
- ❑ Additionally, there are detailed instructions and a nutrition button that you can use to view the nutrition facts

TJ Recipe Layout

- ❑ At the top of my page you are greeted by the banner which is in a header that takes on a class for styling purposes. There is also a site logo that is displayed on both pages to enforce uniformity across recipe sites
- ❑ To style my webpages I mainly used a flex container to hold the main content.
- ❑ In the main content we have ingredients, which includes a label and has an input type of “checkbox”
- ❑ The instructions content utilizes an ordered list to clearly lay out the steps.
- ❑ Additionally, each element of the main content is stored in div element to allow for further customization and styling options.
- ❑ At the bottom of my page, I utilized the flex container again to hold the gallery for my images that are each set to a certain size by a class.

Snippet of TJ Code

```
.brownieimg{
  display: block;
  height:200px;
  width:auto;
  margin-left:auto;
  margin-right:auto;
  border-style: solid;
  border-radius: 5px;
  border-color: ■white;
}

.flexC {
  margin:30px;
  display: flex;
  background-color: ■chocolate;
  border-style: solid;
  border-radius: 10px;
  border-color: ■white;
  border-width: 5px;
}

.flexF {
  display: flex;
  background-color: ■lightblue;
}

.flexC > div{
  background-color: ■whitesmoke;
  margin: 100px;
  padding: 5px;
  font-size: 30px;
}
```

```
<body>
<header class = "main-header">
  <div class = "logo">
    <img src = "cookie.png" alt = "cookie logo">
  </div>
  <div class = "welcome">
    Brownie Recipe
  </div>
</header>


```