Final Styling Touches to the Autocomplete Enabled Engine

In this lesson, we continue to make our backend component look like Google's.

WE'LL COVER THE FOLLOWING

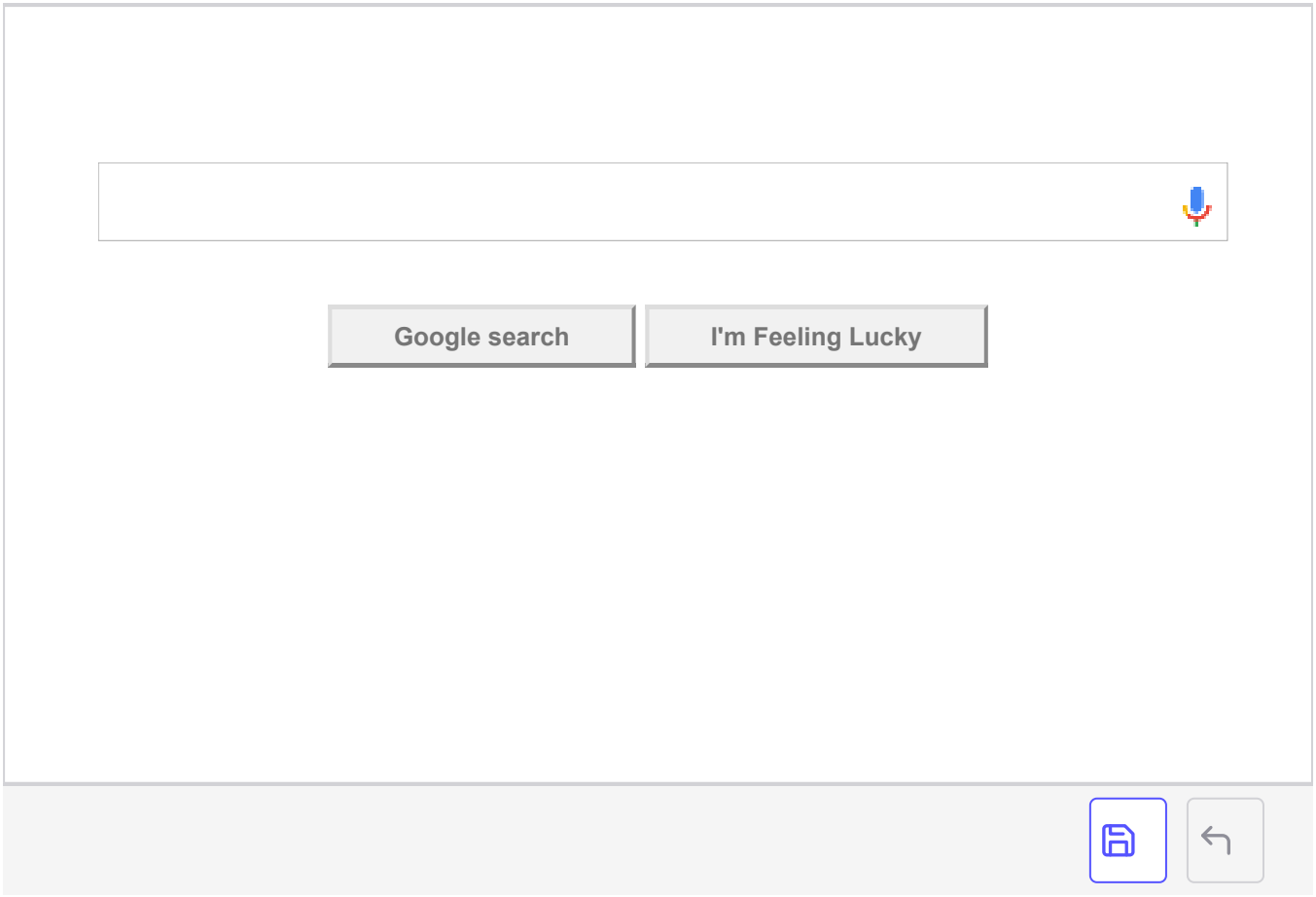
Adding a microphone icon

Alright, we're getting there! Let's add the little microphone icon.

Output

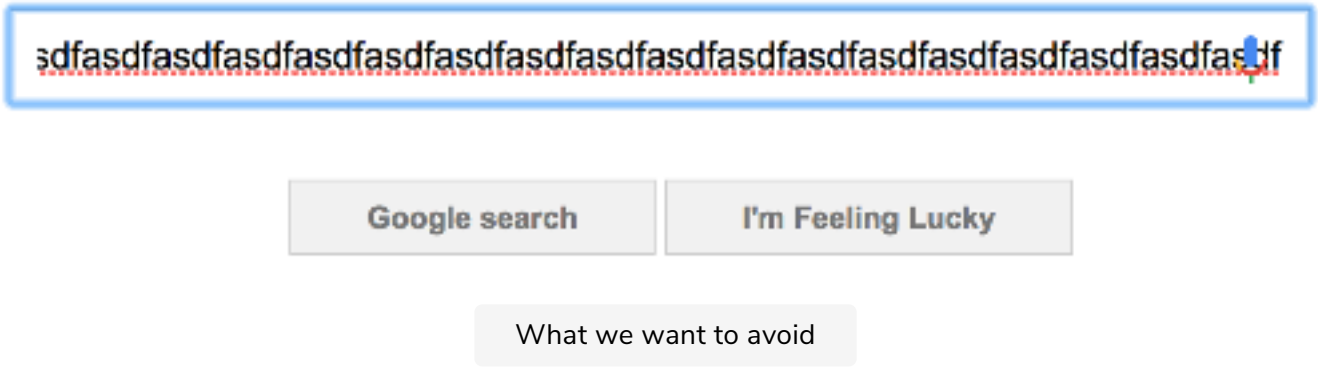
HTML

CSS (SCSS)



Changelist

- Microphone
  - Since the text shouldn't extend to the icon, we added extra padding to the text field
    - Since we've introduced a dependency between two elements, we should make that clear in the code. So I took the width of the microphone as a variable, and the padding is `calc`ed to that width along with the default width (0.6rem).
  - We're using absolute positioning so that it always sticks to the right side of the input bar, and since absolute positioning is relative to the first parent that has a non-static (default) position, we're setting the search bar position relative.
    - This div seemingly had no purpose until now. I hadn't considered this case before when I was laying out the divs, but because we liberally added divs, our HTML is flexible enough to handle all sorts of cases that come up.



What we want to avoid

- A bit of padding to the buttons

Tweaking the buttons

The final thing to do before just making things pixel perfect is to make the buttons merge with the results. It seems our options are as follows:

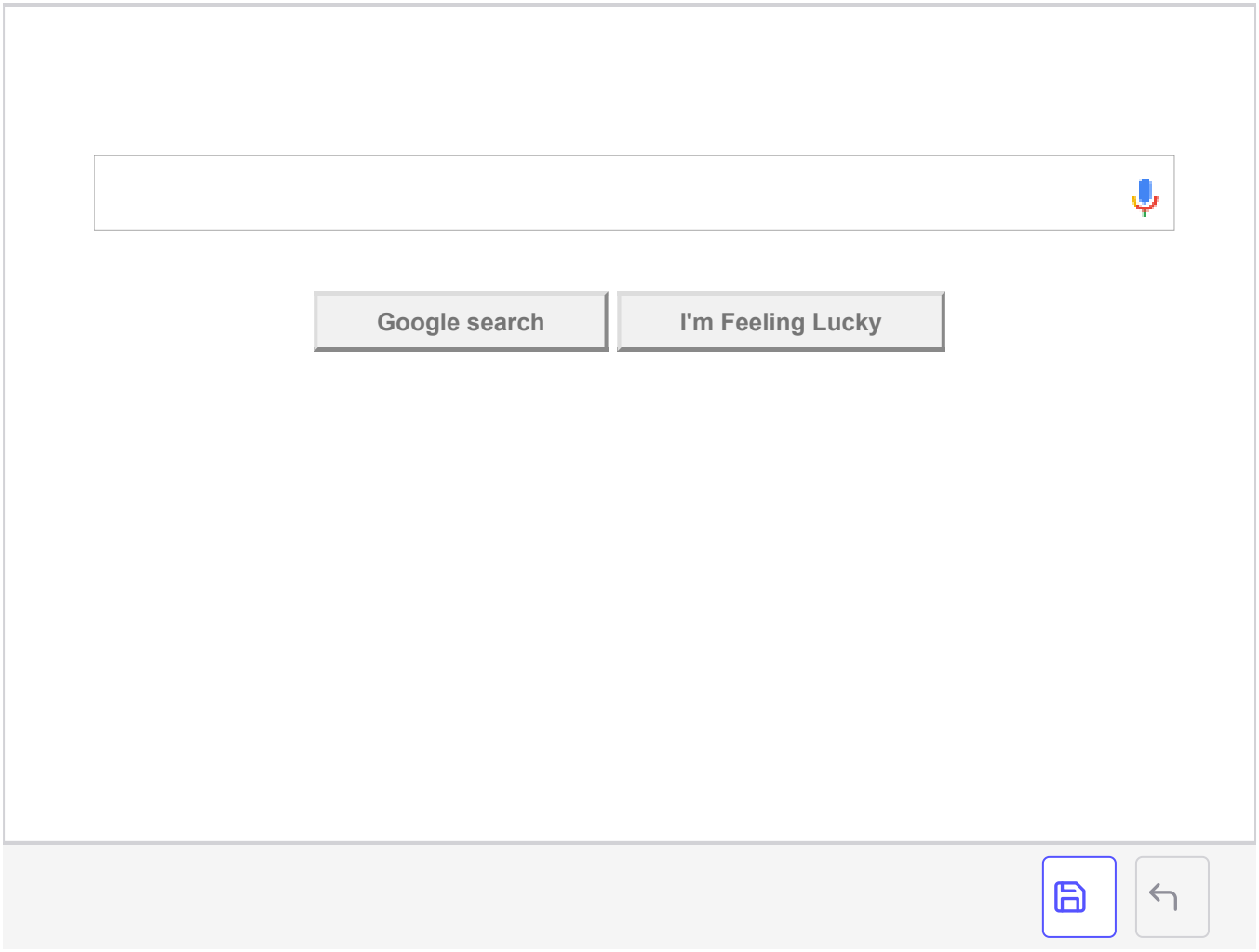
- Add the buttons as another list item in the results list and remove the current one
- Since the background colors are the same, adjust the borders so that it looks like it encompasses the buttons

We'll go with the second option since I had previously anticipated we'll need *some* extra styling for our buttons when autosuggest is active (remember, we're adding `search__actions--autosuggest` when the results show up).

Output

HTML

CSS (SCSS)



I removed the property for the `last-child` of the results since the buttons are now acting as the “last-child.”

By the way, this is a good time to mention another best practice. Keep your rules for related HTML elements close together.

Exhibit A:

```
.foo { ... }
.foo__bar { ... }
.foo__bar--baz { ... }
.foo__bar--waz { ... }

.lorem { ... }
.lorem__ispum { ... }
```

Exhibit B:

```
.foo { ... }
.lorem__ispum { ... }
```

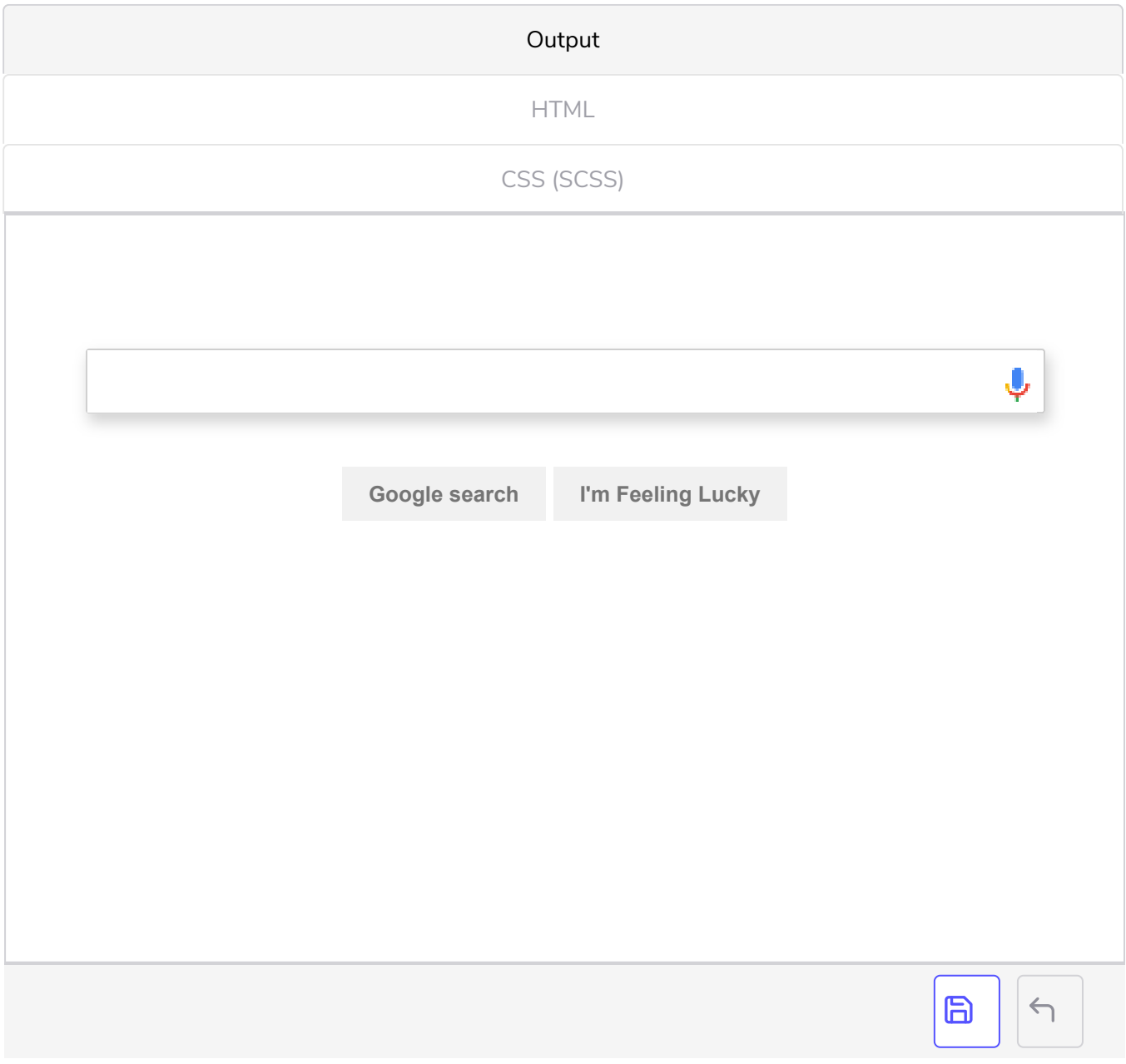
```
.foo__bar--baz { ... }


.lorem { ... }
...
```

The former is easier to grok, and finding what you're looking for is more maintainable since changes to components often involve changes to siblings/parents, which are colocated in exhibit A, and ultimately, it's easier to debug when things don't go as expected.

Final touches to the CSS

We've now arrived at what I believe is the most tedious part of web development: tweaking CSS to make things look pixel perfect. Alas, training your eye to be attentive to the nuances of CSS is a necessary skill in frontend dev.



Changelist

- Box shadow
  - The syntax used here is `box-shadow: $x-offset $y-offset $blur-radius`
  - Since we don't want any box-shadow to show for the bottom of the autosuggest list (to maintain the illusion it's merged with the buttons), I set y-offset to be the negative of blur-radius.
  - I had to dynamically adjust box-shadow as the autosuggest shows up. In the JavaScript client, I added another element to toggle the `--autosuggest` modifier in the same places the buttons have the classes toggled.
- Adjusted the width of the input bar to line up with the autosuggest results
- Removed outline
  - These are generally for accessibility. You might find them useful depending on the audience you're targeting with your web app.
- Added border-radius.
- Removed the border from the input and applied it to the surrounding div to match up the lines
- Changed the cursor hover over the list items to the default
- Buttons
  - Remove the border to give it that "flat design" look
  - Adjusted the padding – it was too much the first time around

That wraps up our component! Thanks for following along, and see you in the next chapter.