CURSED SOCIALS · HACK CLUB YSWS · GUIDE PART 3 OF 3 (THE FINALE)

JavaScript & Shipping (make buttons actually work)

You actually made it to the final part. Your social site already looks legit — now let's make the buttons react when people click them and put your site live on the real internet.

People act like JavaScript is rocket science. For your first site, it's really not — you only need about 10 lines to make a button feel alive. Here's the finish line:

- **Part 1 (done!):** Files, basic HTML tags, colors, fonts, and the Box Model.
- **Part 2 (done!):** Navbars, avatars, photo grids, and making it work on phones.
- **Part 3 (you are here):** Working like counters, dark mode, live comments, and dropping your folder online for the grant.

1. Plug in your third file: `script.js`

Create one last file inside your `my-social` folder called `script.js`:

```
my-social/
├── index.html   <-- the page
├── style.css    <-- the looks
└── script.js    <-- the button magic
```

Now connect it to your HTML by adding this single line **at the very bottom of your `<body>`**, right above `</body>`:

```
  <!-- all your posts and buttons go above here -->

  <script src="script.js"></script>
</body>
</html>
```

> **Why the bottom of `<body>`?** Browsers read your file from top to bottom. Putting your script at the bottom guarantees your buttons already exist on the page before JavaScript tries to grab them.

Quick 3-second test

Type `console.log("js is alive!!");` inside `script.js`, save, refresh your browser, right-click the page, hit **Inspect**, and click the **Console** tab. If you see `js is alive!!`, you're connected.

2. The only JS pattern you need today

Almost every interactive feature on the web follows the exact same 3-step loop. Once this clicks, you can build anything:

```
[ 1. GRAB IT ]      --->   [ 2. LISTEN ]        --->   [ 3. CHANGE IT ]
Find the button            Wait for someone            Add 1 to likes,
by its id in HTML          to click on it              swap colors, etc.
```

JS Piece	What it means in plain English
`let likes = 0;`	Creates a box in memory (a variable) that holds a number you can change later.
`document.getElementById("like-btn")`	Searches your HTML and grabs the element with `id="like-btn"`.
`.addEventListener("click", ...)`	Tells the browser: "run this chunk of code every time someone clicks here."
`.textContent = "..."`	Replaces the words written inside a button, heading, or paragraph.
`.classList.toggle("dark")`	Switches a CSS class ON if it's off, or OFF if it's already on (perfect for dark mode).

3. Recipe 1: A Real Like Counter

First, make sure your button in `index.html` has an `id` so JavaScript knows which button to watch:

```
<button id="like-btn" class="like-btn">Like (0)</button>
```

Now paste this into `script.js`, save, and start clicking:

```
let likes = 0;
const likeBtn = document.getElementById("like-btn");

likeBtn.addEventListener("click", function () {
  likes = likes + 1;
  likeBtn.textContent = "Like (" + likes + ")";
});
```

Want to make it slightly funnier? Drop an `if` statement inside that click function so it reacts when someone spams the button:

```
  if (likes === 10) {
    alert("bro stop liking your own post");
  }
```

4. Recipe 2: One-Click Dark Mode

Dark mode looks super impressive and literally takes 4 lines of JS. Add a toggle button in your navbar in `index.html`:

```
<button id="theme-btn">Dark Mode</button>
```

Add a `body.dark` rule at the bottom of `style.css` to define your dark colors:

```
body.dark {
  background: #18161a;
  color: #f4ecdc;
}

body.dark .post {
  background: #242129;
  border-color: #f4ecdc;
}
```

Then tell `script.js` to flip that `dark` class on and off whenever the button is clicked:

```
const themeBtn = document.getElementById("theme-btn");

themeBtn.addEventListener("click", function () {
  document.body.classList.toggle("dark");
});
```

5. Recipe 3: Live Comment Box (Bonus Flex)

Want people to actually type a comment and see it appear on your feed? Add an input box, a button, and an empty container in `index.html`:

```
<input id="comment-input" placeholder="say something...">
<button id="post-btn">Post</button>
<div id="comments-list"></div>
```

And wire it up in `script.js`:

```
const input = document.getElementById("comment-input");
const postBtn = document.getElementById("post-btn");
const list = document.getElementById("comments-list");

postBtn.addEventListener("click", function () {
  if (input.value === "") return; // ignore empty clicks

  const newComment = document.createElement("p");
  newComment.textContent = "@you: " + input.value;
  list.appendChild(newComment);

  input.value = ""; // clear the box after posting
});
```

> **Note:** When you refresh the page, those comments disappear because there's no database yet — that's 100% normal and totally fine for this project!

6. Put it online (literally takes 10 seconds)

Right now your site only exists on your own laptop. Let's give it a real URL anyone can open:

1. Double-check that your main page is named `index.html` (all lowercase!) directly inside your `my-social` folder, alongside `style.css`, `script.js`, and any images you used.
2. Open **netlify.com/drop** in your browser.
3. Drag your entire `my-social` folder from your Desktop straight into the dashed box on the screen.
4. Wait about 5 seconds — boom, Netlify gives you a live `.netlify.app` link!
5. Open that link on your phone or send it to a friend to make sure your images and buttons work for other people too.

> **Heads up:** Free drag-and-drop links on Netlify expire after an hour unless you make a free Netlify account (it takes 20 seconds and keeps your site online forever, which you'll want so reviewers can open your link!).

7. Wait, why isn't my button doing anything?

Every developer hits JS bugs constantly. Open **Inspect** → **Console** first — any red text tells you the exact line number that broke.

What went wrong	How to fix it in 5 seconds
Console says `Cannot read properties of null`	Either your `<script>` tag is inside `<head>` (move it right above `</body>`) OR the `id` in HTML doesn't match the exact spelling in JS.
Console shows a red `SyntaxError`	You missed a closing quote `"`, parenthesis `)`, or curly brace `}`. Check the line number printed on the right of the error.
Site works on laptop, broken on Netlify	Check that your file is named `index.html` (not `Index.html` or `home.html`) and all photo files were inside the folder you dragged.
Nothing happens at all, zero errors	Save both `index.html` and `script.js` (Ctrl+S), then do a hard refresh (Ctrl+Shift+R).

Pre-submit checklist

- You wrote your own `index.html`, `style.css`, and `script.js`.
- Your page looks like a social feed (clone of an app or your own original idea).
- At least one JavaScript thing works when clicked (like counter, dark mode, comment box, popup, etc.).
- Your site is live on a real URL and opens cleanly without broken images.

You finished the whole guide — go ship it!

You started with an empty folder and built a real, live website from scratch. Submit your live link for your **$5-$10 Hack Club grant** and drop it in **#cursed-socials** on Slack so everyone can try it out. Once you've shipped Part 1, head over to **Part 2: Cursed Socials** — where you take everything you just learned and intentionally make the worst, funniest UI possible.