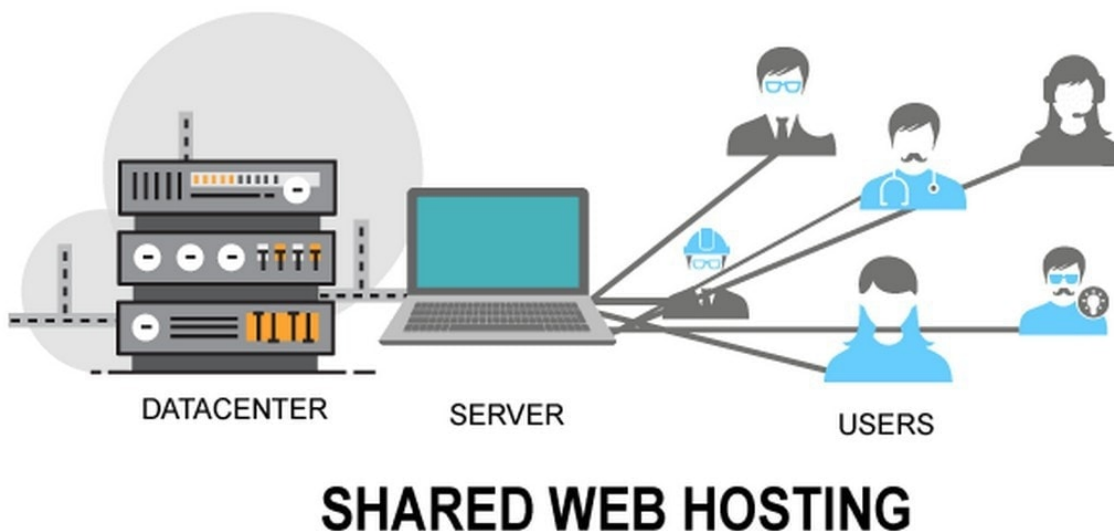


Why Single Page?

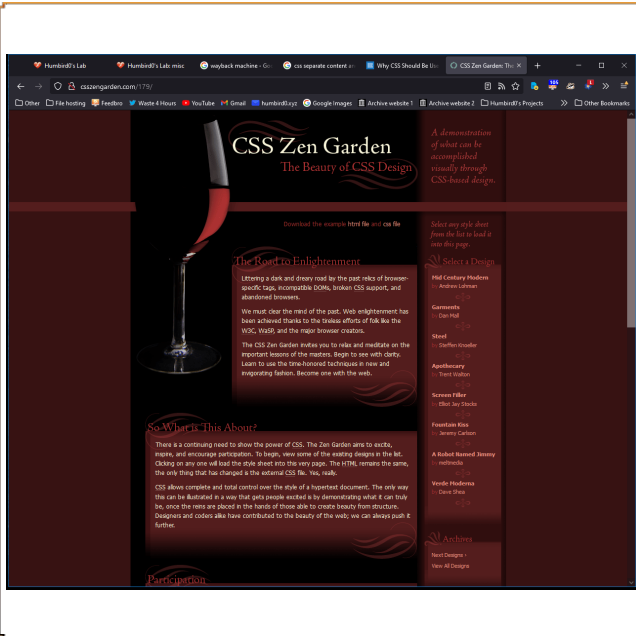
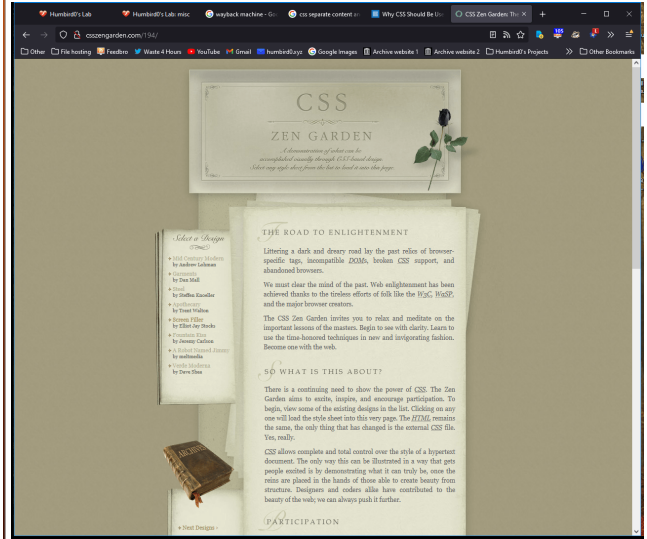
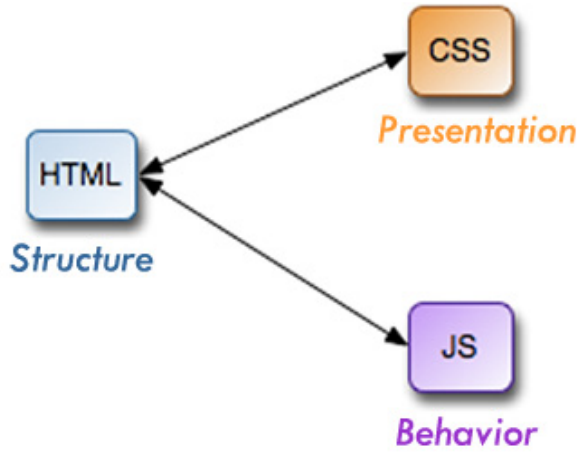
My website is primarily a single-page JavaScript site. I originally did this in pursuit of extreme efficiency. Only load the parts that change and keep everything else the same.



My original obsession over server efficiency came from when I was working on my Pokemon Hentai game and the free shared hosting I used at the time kept getting overloaded from all the downloads of the game. The host complained about “CPU Overuse”

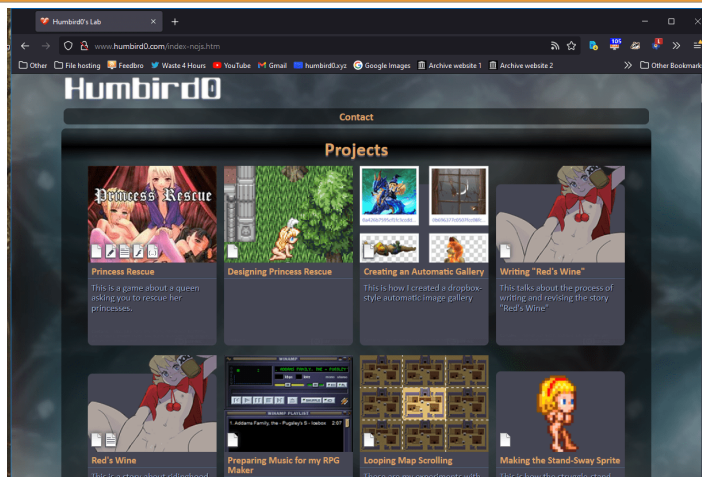


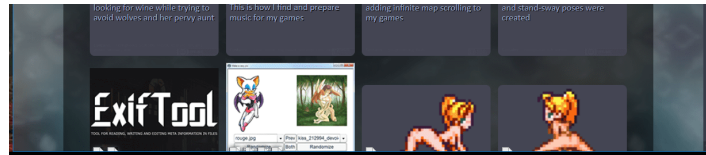
But over time my website changed priority toward being archive-friendly for the Internet Wayback Machine, so I started storing the data in the HTML itself instead of a tiny JSON file.



This also allowed the site to work without javascript with a nice-looking but functionally crude thumbnail gallery.

<http://www.humbird0.com/index-nojs.htm>

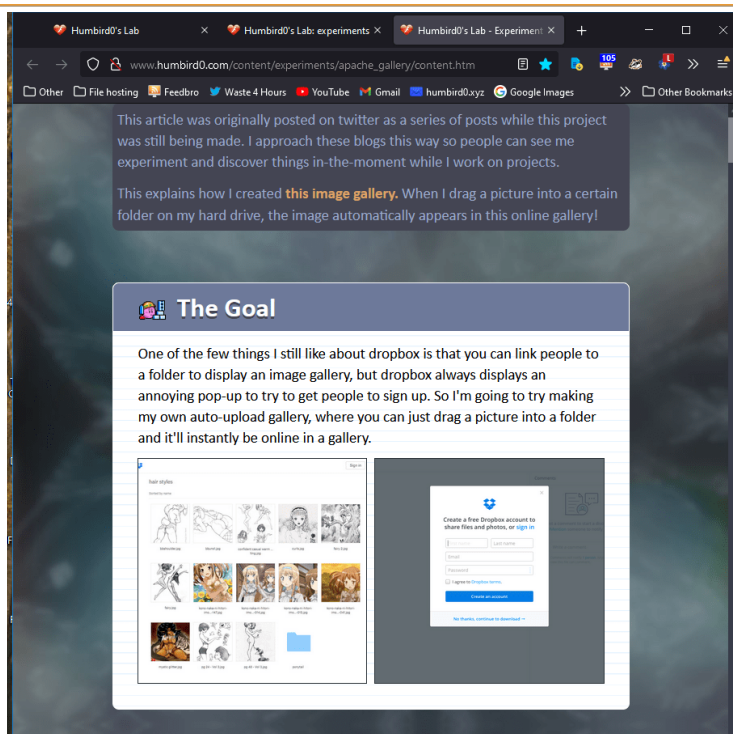




So lately I've been wondering what it might look like if it was primarily designed without JavaScript at all. Could the experience of navigating raw HTML be improved? This is more of a thought experiment. I actually do like my current website as it is.



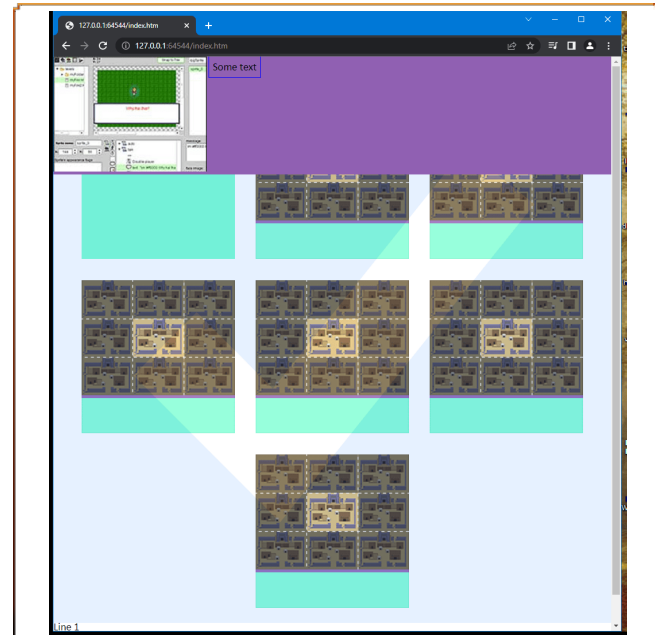
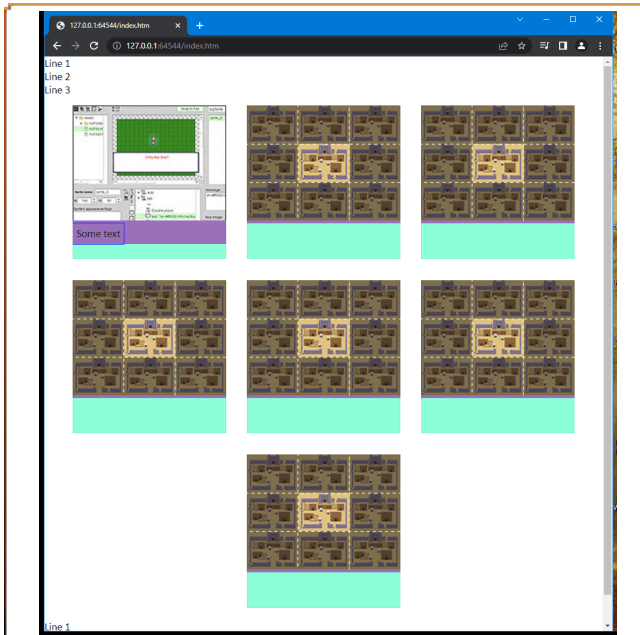
Even without JavaScript, I could have the thumbnails expand when I click on them, by attaching some CSS to invisible checkboxes. I did this with my RSS blog experiments.



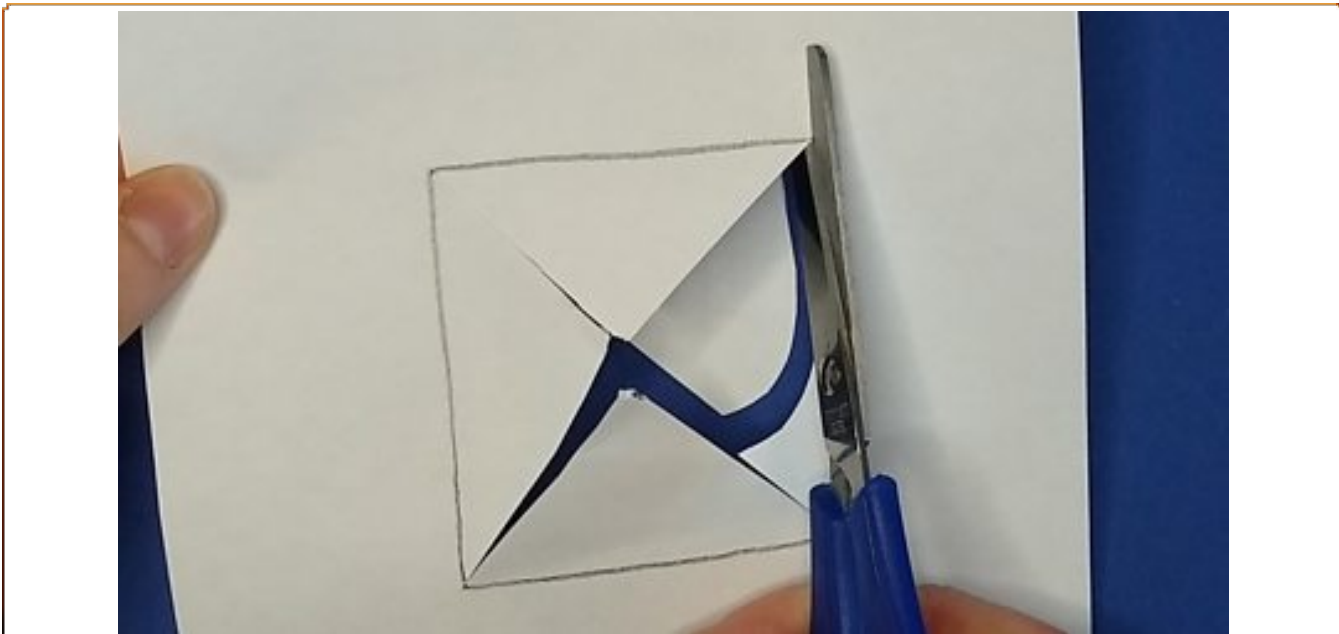
Attempt 1

0:01 / 0:04

What if I had the same thumbnail gallery, but they expand when you click on them? Just stretch an invisible checkbox over the entire thumbnail when it's unchecked, and make the checkbox full-screen when it's checked. Then sort it behind the content for click-away cancelling.



But how can I also SHOW people a way to close it? I need a standard close button on the top right. But without Javascript I need to re-use the existing checkbox. Maybe I can cut a hole in the content somehow to click the checkbox behind it?

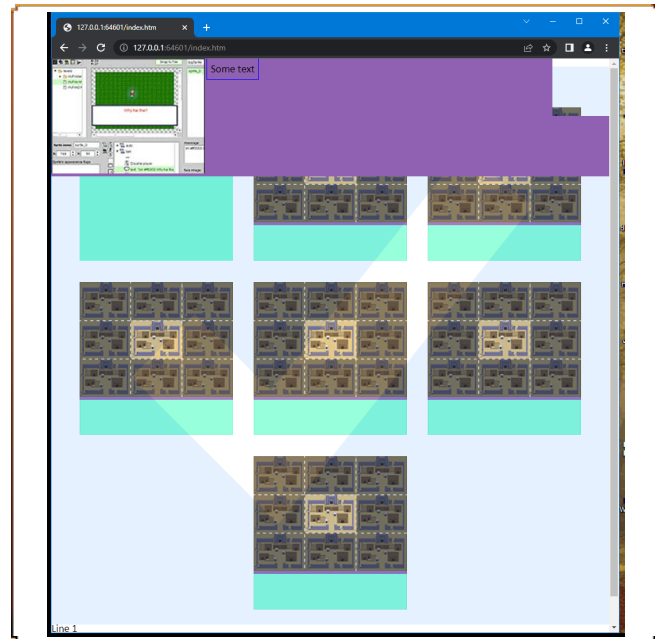
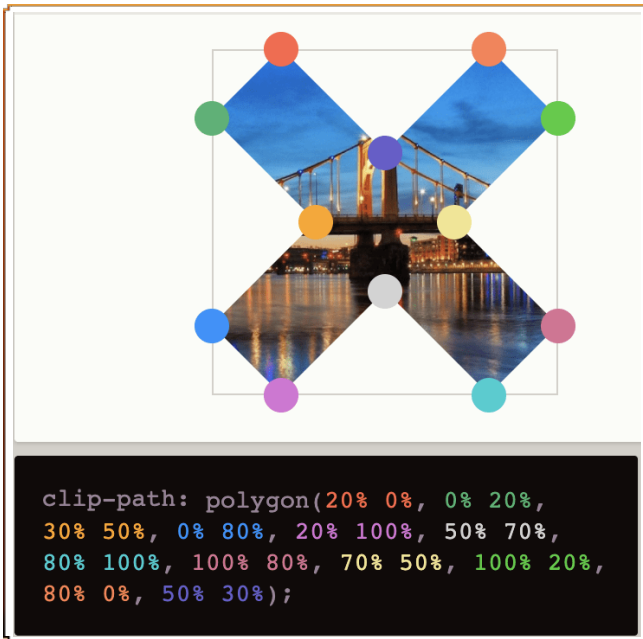


How would I even do that? Maybe I could use an SVG image with a hole in it as the

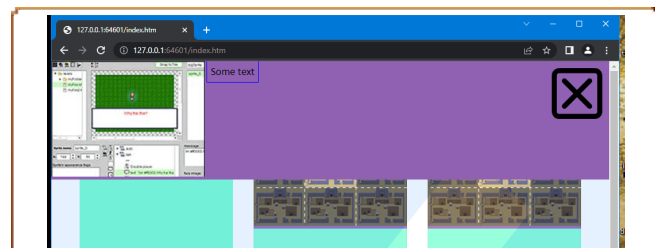
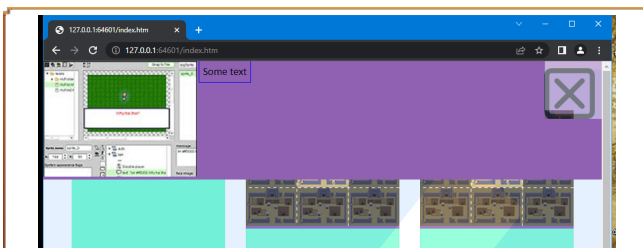
background? Nope that doesn't work. The browser treats the entire thing as a square image.

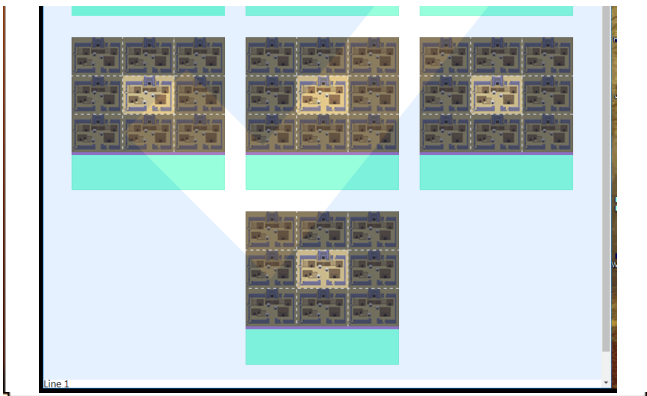
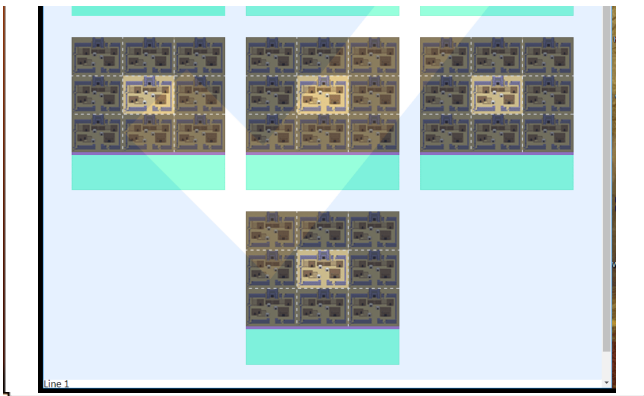


Apparently clip-path can do it.

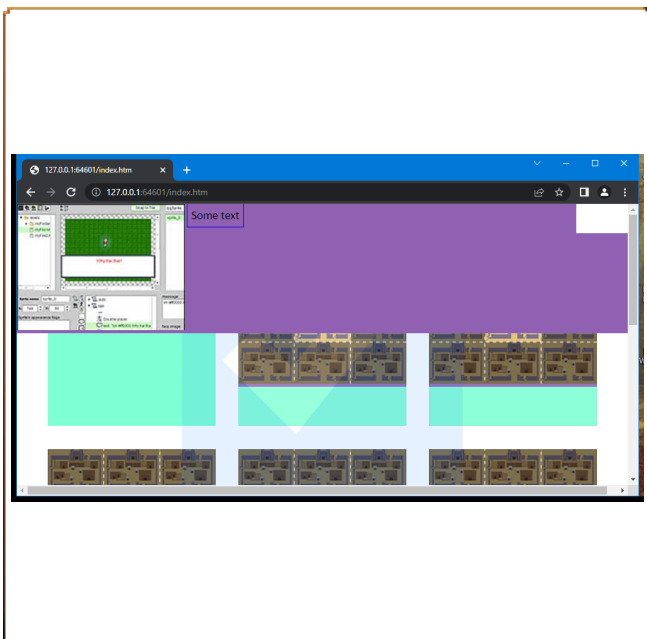
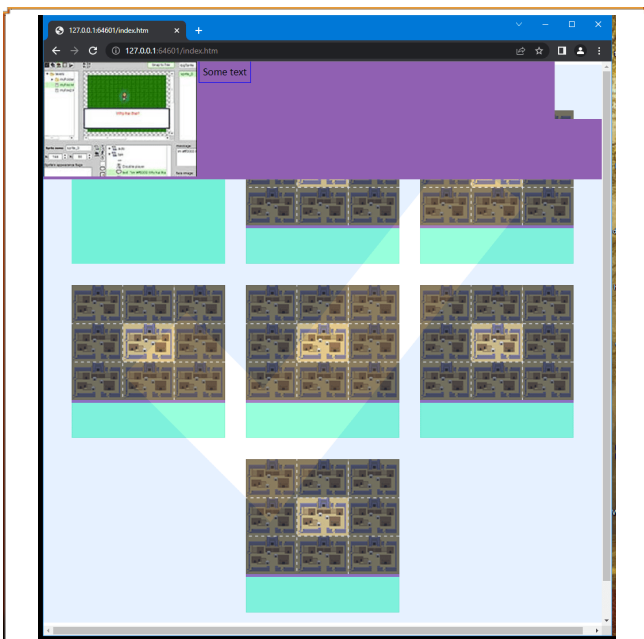


Now I just need to overlay a close icon over the hole and make that click-through using pointer-events none.

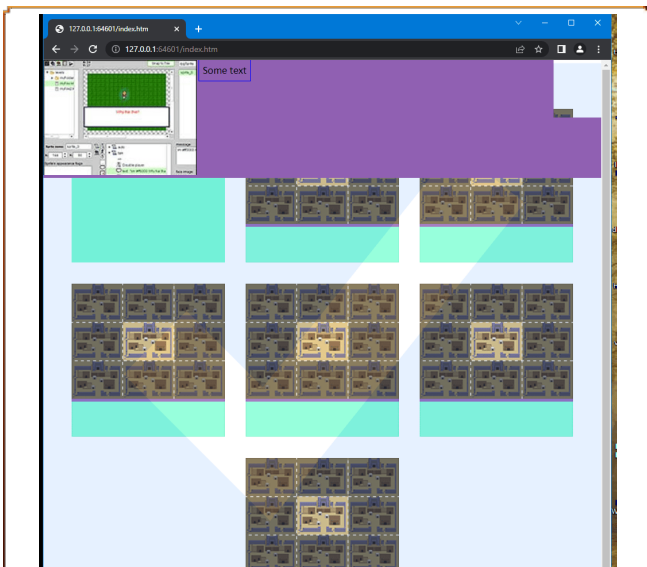


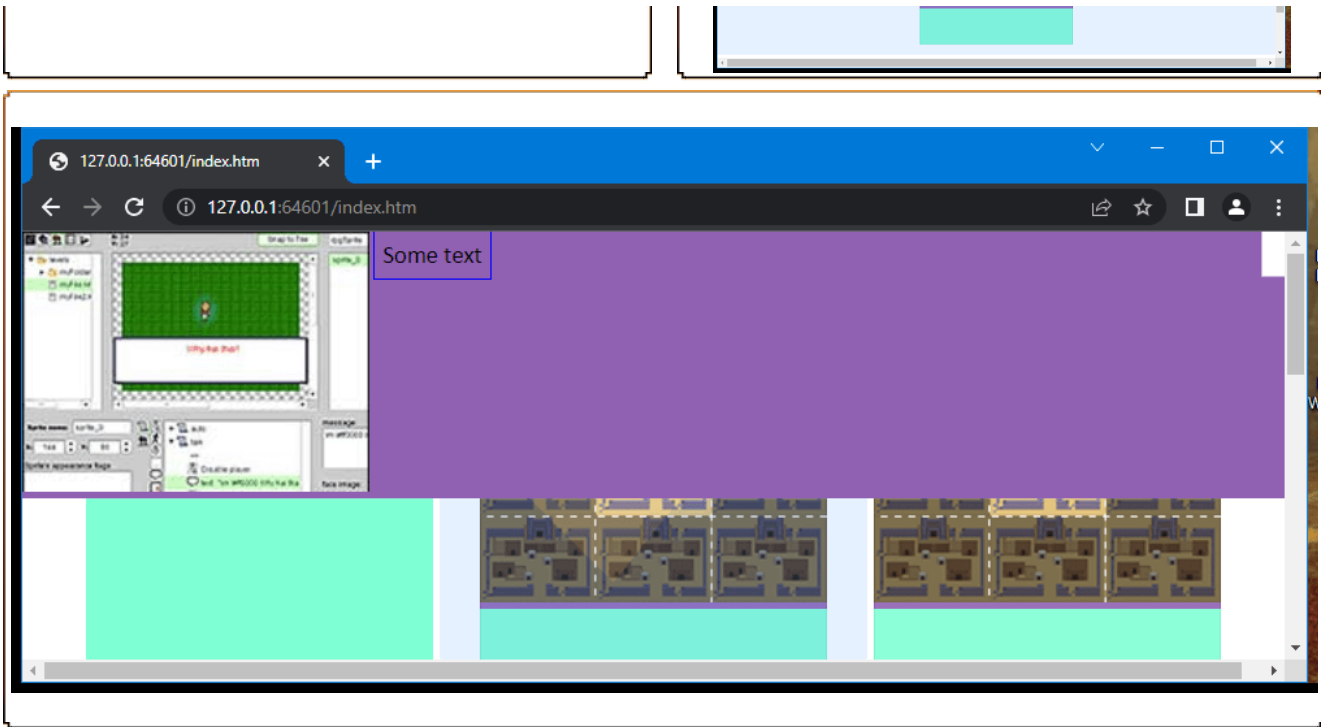


Controlling the size of this hole is tricky because it's based on `vw` `vh` units which are percentage of the viewport. I can't just say `100% - 12pt` because CSS doesn't let you mix units.

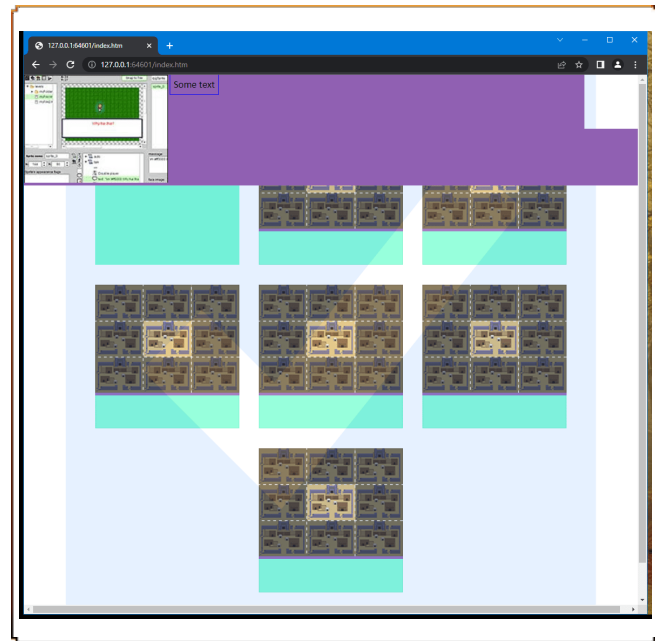
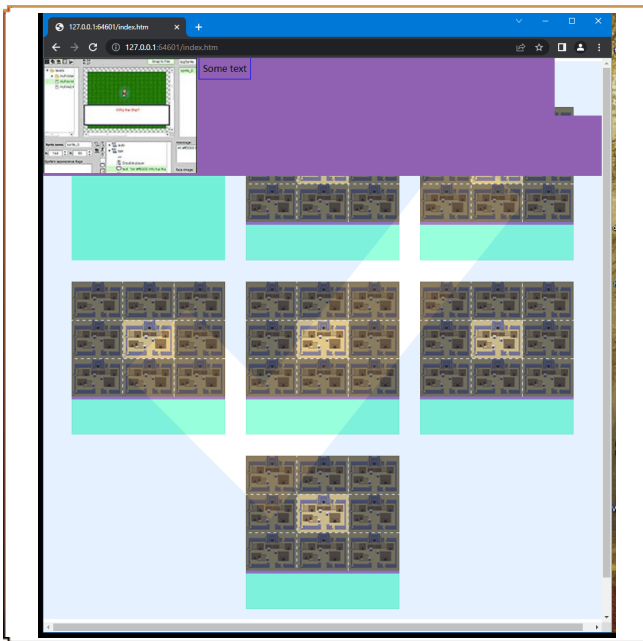


... Unless I use `calc()` then I CAN mix units!
But why does it sometimes look like a rectangle instead of a square?

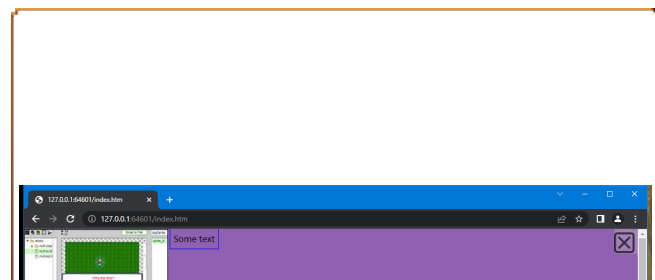


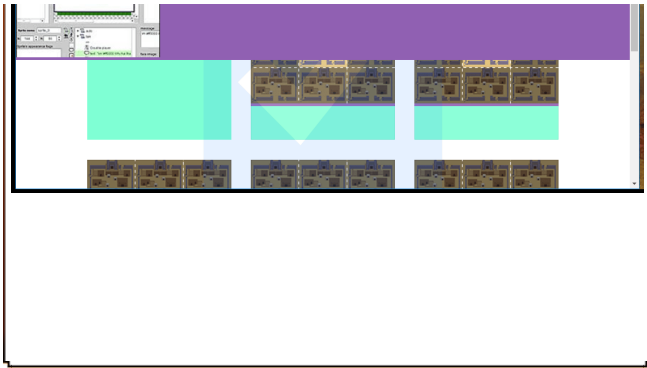
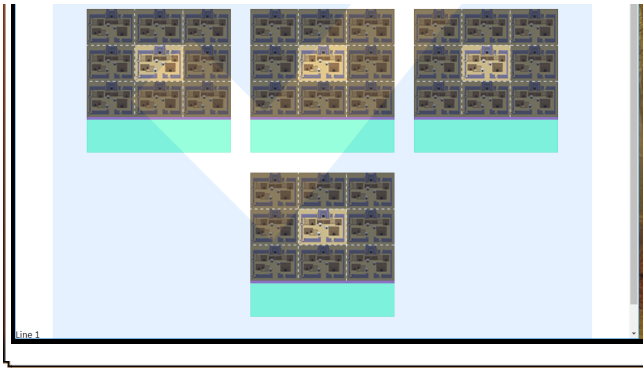


The scrollbar is covering it up! I'll just uh, subtract 16px and always have scrollbars visible.



The nice thing about using calc() is that this 'close button' is large on vertical displays (like phones) and small on wide displays (everything larger than phones)

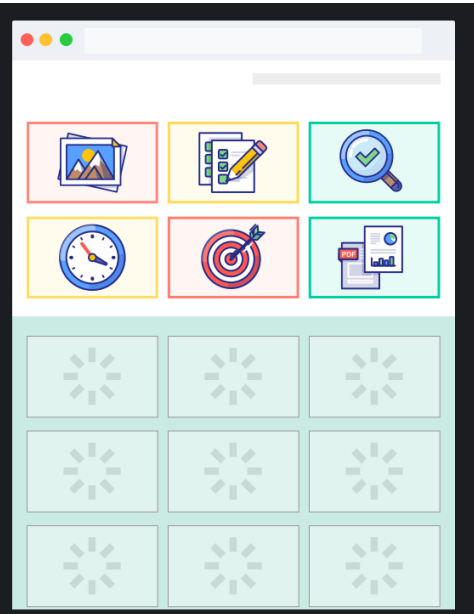




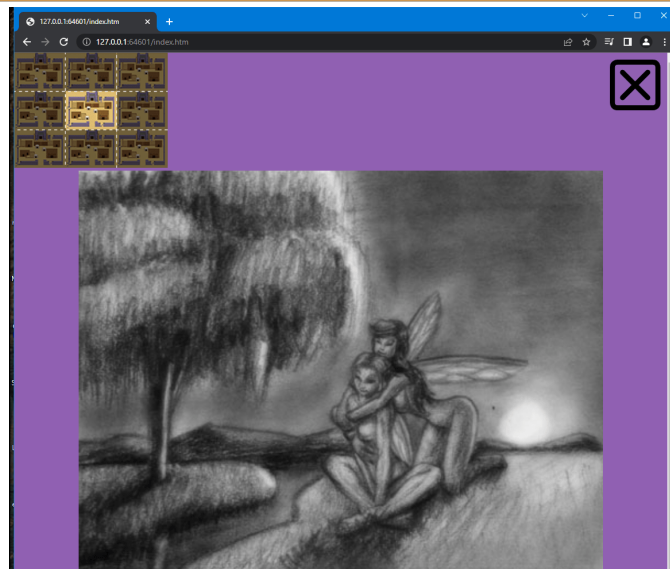
I can just make the content hidden while the checkbox is OFF and visible when it's ON and use loading='lazy' to prevent the browser from loading EVERY SINGLE video and full-size image until they're actually visible.

THE 'LOADING' ATTRIBUTE

```
<img loading=lazy>
<iframe loading=lazy>
```

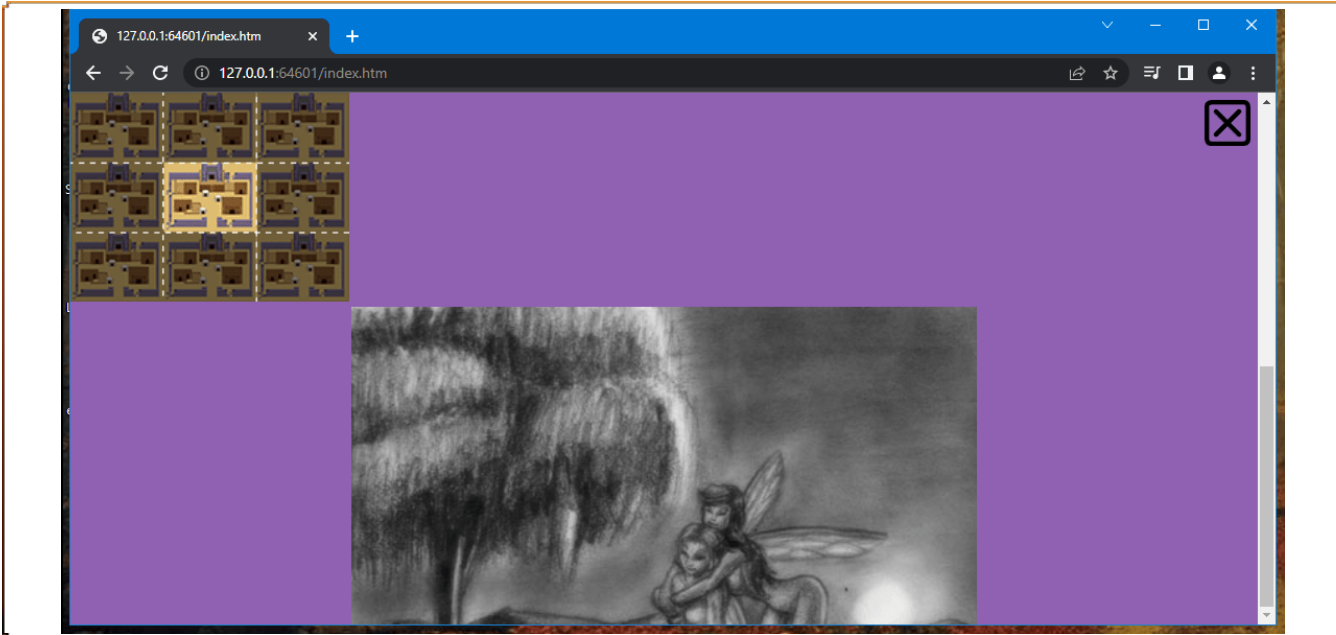


If I scale the content to ScreenHeight - ThumbHeight then you won't need to scroll to see it.





... Unless the screen is very small. Wait, why isn't scrolling working??



Apparently setting bottom to 0 fixes that. But you can't actually see the scrollbar?

```
css.less
// :root: vars;

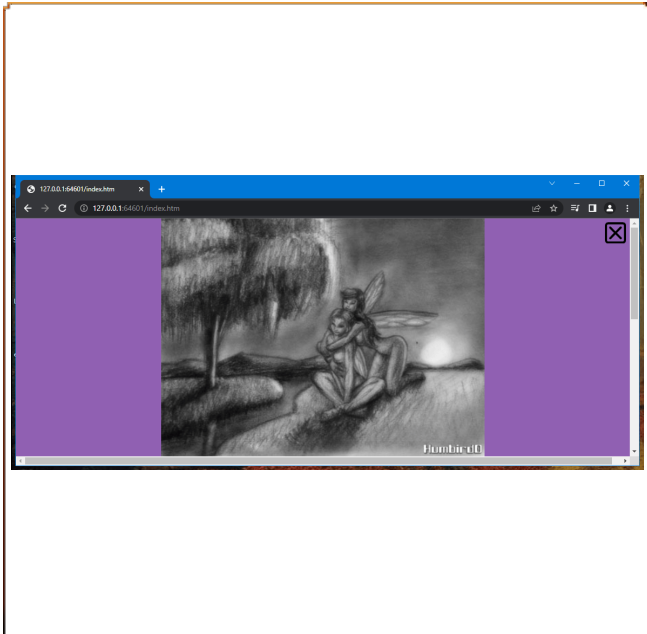
> .thumb-img{
  width: @thumb-width;
  position: relative;
} // img

> .media{
  display: none;
  width: @media-width;
  min-height: @media-min-height;
  max-height: calc(@media-max-height - @thumb-height);
  object-fit: contain;
} // .content (thumbnail content wrapper)

// .content (checkbox ON)
> input[type=checkbox]:checked ~ .content{
  width: @expanded-width;
  position: fixed; // relative to the screen, instead of the thumbnail
  top: 0;
  left: 0;
  bottom: 0; // enable scrolling
  overflow-y: scroll;
  // cut a hole in this element to allow the checkbox behind it to be clicked
  clip-path: polygon(0 0, calc(100vw - @scrollbar_size - @close-button-size) 0, calc(100vw - @scrollbar_size - @close-button-size) @close-button-size, 100vw @close-button-size, 100vw 100vh, 0 100vh);
  z-index: @expanded-layer;
}

> .media{
  display: inherit;
}
} // .content (checkbox ON)

// display a "close button" where the clip-path hole is
// close button (checkbox OFF)
.close-button
```

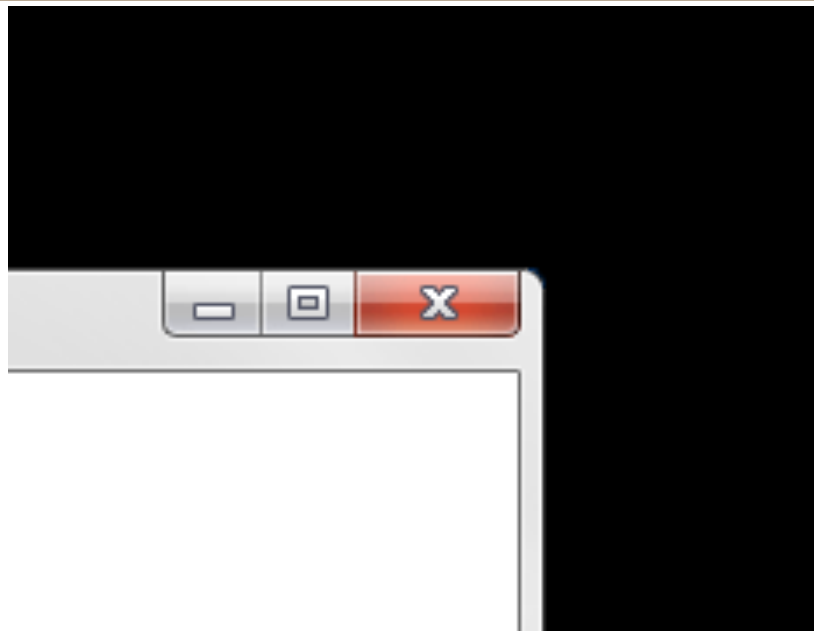


Also, the expanded view is now always fullscreen. So I guess I successfully made a 'single-page-app' without javascript.

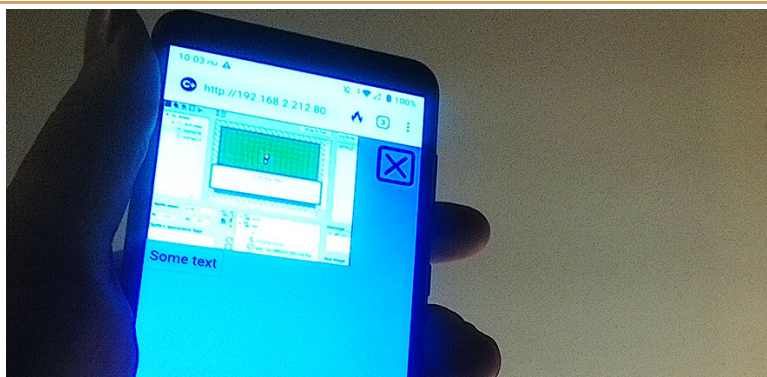


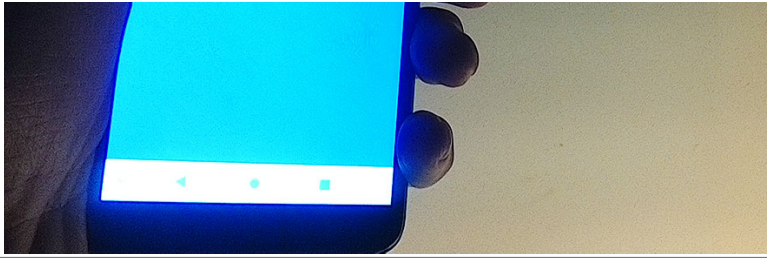


Wait a minute. If it's always fullscreen, then I no longer need to put the checkbox behind everything, so I could just move the checkbox itself to the top right and directly use it as a close button now. No holes needed.



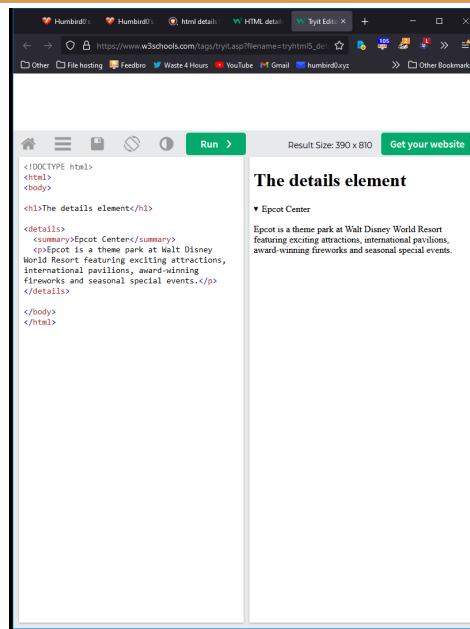
But there's a problem with things being full-screen. It'll look like a new page and on a phone you'll want to press the phone's 'back' button to exit the expanded view. But that will actually LEAVE the entire page!



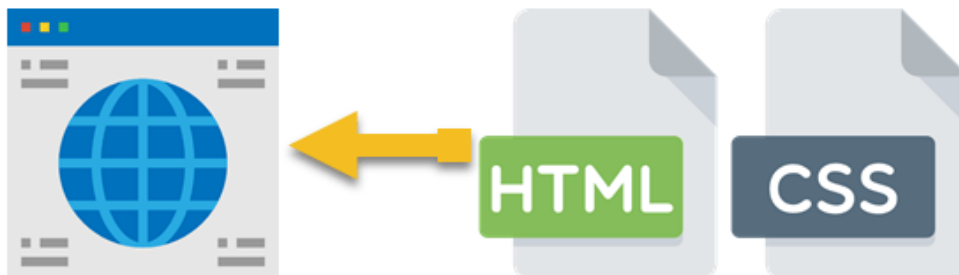


Maybe I could use a collapser instead of a checkbox? HTML has a native `<details>` element that does this without javascript. It's been around for years. For some weird reason NO website knows this tag even exists. It works fine.

https://www.w3schools.com/tags/tag_details.asp

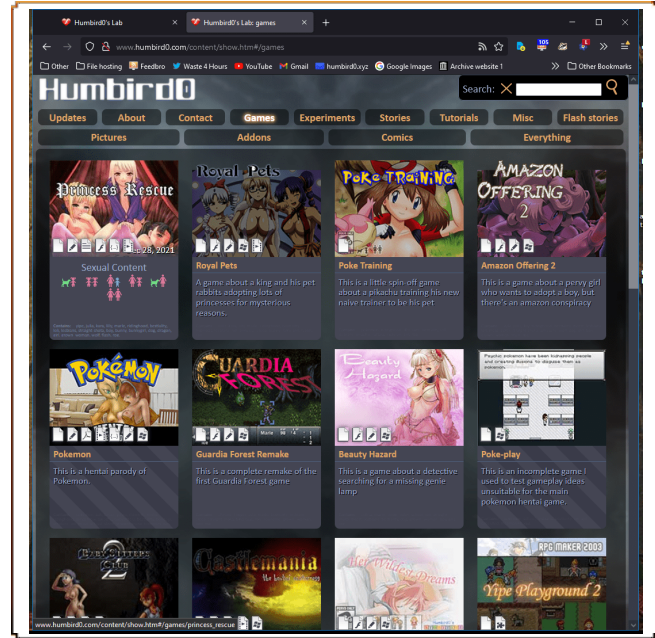
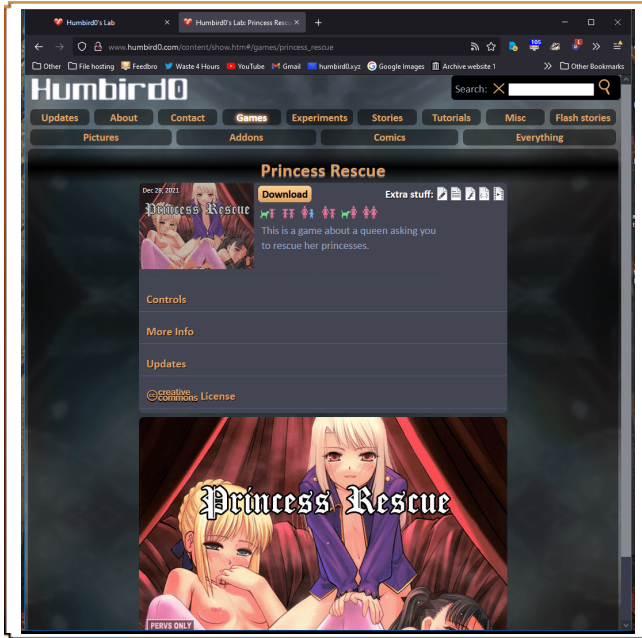


Maybe the least confusing solution is to just use separate web pages. Just like in 1998. I guess using fancy CSS for this is a bad idea after all.

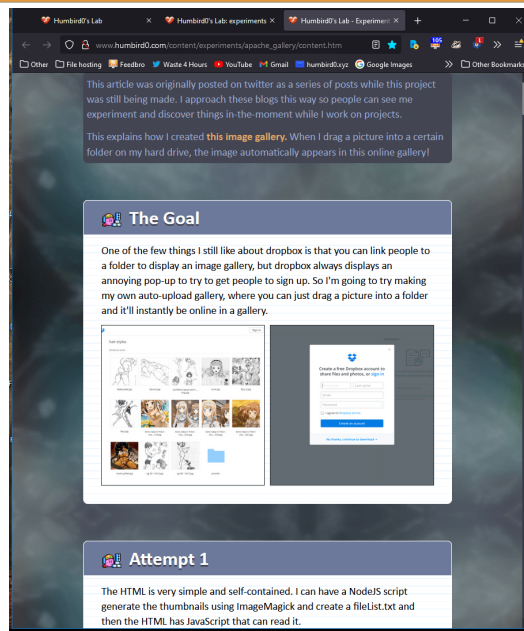


STATIC WEBSITE

I already use separate pages for the 'more info' part of my projects. The only problem with putting all the info in these pages is that you have duplicate info between the gallery and the individual pages.



I COULD leave out the title, thumbnail, and content labels in the separate pages and only display those in the thumbnails, but if someone finds one of the 'info' pages through google they won't know the title.



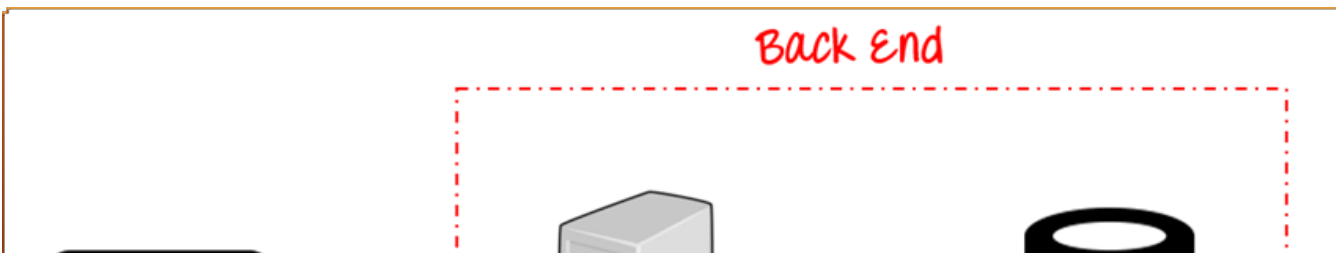
... or how to get back to the gallery, since the navigation would be missing, so I'd need to at least duplicate that part. This is a very old problem with a very old solution. Templates. That's why PHP was invented.

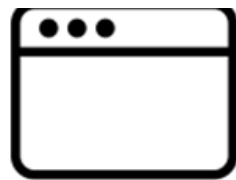


Before templates existed, framesets tried to fix this, but navigation still goes missing if you reach a page directly from Google. They also suck on mobile phones. Don't use them.



I've been trying to avoid relying on any server-side stuff since that whole side of things is hidden from Wayback Machine archiving, but duplicating info and navigation would probably be okay.





Browser



Web server



Database

© guru99.com

I've never used PHP. I heard it's terribly designed. But I also know it's been around for decades and was originally created for templating, so it's probably pretty good at its intended purpose. Just load 2 HTML files and display them both.



And no, I still won't use a database. My static website doesn't need SQL injection vulnerabilities, thank you.



