

Verkada - Front-end Architect Coding Assignment

Motivation

Verkada has a web-based command center where users can see all of their cameras. To quickly view the last 30 days of footage from a camera, we've created an "infinite" scrolling list that displays a grid of thumbnails. We only have a finite number of thumbnails so it's not truly infinite, but we can easily scroll to any of the hundreds of thousands of images. A naive implementation would be to put each image in an `` tag, but the browser cannot download and render all of those images, so we need a better solution.

Your challenge is to create this infinite scrolling list using thumbnails from our office camera.

This is not a news-feed style list where more items load after you reach the bottom: the user must be able to use the browser scrollbar to jump to any image. We are only interested in visualizing the historical thumbnails (no real-time data).

We don't expect you to be a brilliant visual designer, but do pay attention to the details and send us something polished!

Thumbnails

- `https:// Verkada.com/office-camera/` ; the last timestamp is `1544111111`. For example, the first thumbnail is: `https:// Verkada.com/office-camera/1544111111.jpg`
- Thumbnails are taken every 20 seconds. The first timestamp is `1544111111`; the last timestamp is `1544111111`. For example, the first thumbnail is: `https:// Verkada.com/office-camera/1544111111.jpg`

Requirements

- Use Javascript, HTML, and CSS (or your favorite CSS derivative)
- You can use UI framework such as React
- Please avoid using jQuery or any sort of "infinite scrolling list" libraries
- Thumbnails should be shown in a grid with 3 thumbnails per row.
- Only the visible images should be downloaded and displayed: when the page loads, don't immediately create $n = 100,000$ `<div />`s or $n = 100,000$ network requests. Ideally, the number of DOM nodes on the page should be $O(\text{visible images})$, not $O(n)$!
- The browser's native scrollbar should be usable -- users must be able to scroll to anywhere in the history by clicking and dragging the scrollbar.