Kargo Coding Challenge - Image Comparison Tool

This task is meant to be submitted as code and a working web tool. Please do this on a personal AWS account - and let us know if you need additional AWS resources. We will review the code and the functionality of the tool. We will not be reviewing the design. If you have any questions send an email to jonathan@mykargo.com

Overview: Create a web application that queries sets of images stored in AWS and presents them to users for a selection task. As users complete selection tasks - more tasks will appear.

Images: Images will be stored in an S3 bucket, no folders. Images will be ~300KB each and stored as pngs. The bucket could contain up to 10,000 images.

Sets: Each image will belong to a single set uniquely identified by an integer. One image in the set will be marked as the base image. Each set will have 2 - 50 images.

Table: An SQL table with four columns will exist. Each row will correspond to an image in the S3 bucket. The columns are:

- image:VARCHAR(255) with the filepath of the image
- set: INTEGER with the set number that the image corresponds to
- base: INTEGER with 1 if the image is a base image, and 0 if not
- result: VARCHAR(255) with the result of the selection task

You may add any columns you want to the table.

Selection Task: A selection task consists of comparing all of the images in a set to the base image. The user should be able to see the base image alongside the other images in the set. The user should be able to mark each non-base image in the set as 'MATCH', 'DIFFER', or 'UNKNOWN'. Once the user has marked all of the images in the set they should be able to review their selection and confirm it. When the user confirms the selection task - it should immediately update the *result* column of the table.

Task Queue: When a user completes a selection task, they should automatically see the next selection task. Once a selection task is complete, it should **never** be shown again to the user - even after reloading the page. This should continue until there are no selection tasks left.

Web Support: Desktop only. Chrome 87 only. No HTTPS needed.

Backend: Should be able to run on a relatively small AWS instance and load quickly. Any and all AWS tools can be used.

Sessions: If the user closes the browser in the middle of a selection task, the results do not need to be stored. Only needs to support a single person marking selection tasks at a time. No logins, security, or accounts are required.

Design: Very minimal design constraints - **it does not need to look good at all.** At any point in time the user should be able to see how many tasks are left, see how many images are left total, and see how many images are left in their current task.