Matroid front-end challenge

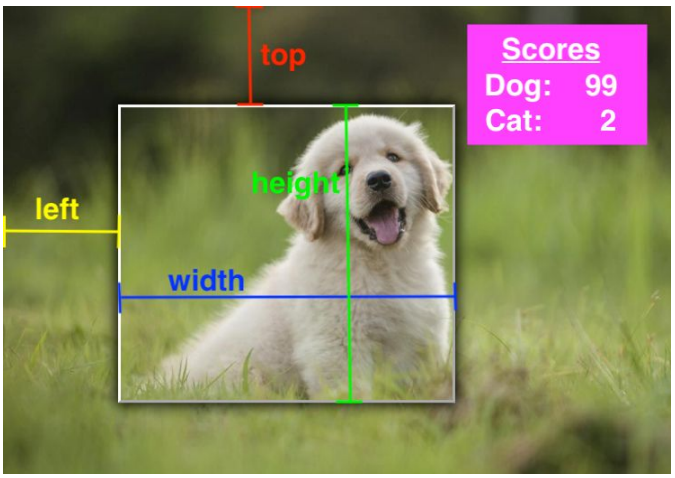
Part of Matroid's functionality is monitoring video streams for events. Your challenge is to create a single page web app to display a timeline of events that have been detected from multiple source streams.

Events data

When your app loads, it should fake an api request that receives the event data in event_data.txt as a response (note: you should change the file extension from .txt to .js; it's a text file so it doesn't get flagged by email clients). Events are structured as follows:

```
{
  videoStream: 'Pet Store',
  timestamp: 1506867328,
  imageSource: 'https://site.com/path/to/img',
  predictions: [
    {
      boundingBox: {
        top: 0.113,
        left: 0.254,
        height: 0.754,
        width: 0.498
      },
      scores: [
        {
          label: 'Cat',
          score: 99
        },
        {
          label: 'Dog',
          score: 10
        }
      ]
    },
  ]
},
```

Events have an associated video stream, a timestamp in seconds, and an image source that links to an image of the event, and one or more predictions. Each prediction has a bounding box indicating the location of an object of interest within the image, and one or more scores indicating potential labels for that object.



Feature requirements

Core requirements

- Users should be able to view and interact with a timeline of all events
- Users should be able to click on an event and see a detailed view that:
    - Displays the event image
    - Shows the location and scores for each prediction
    - Shows the event time in a human readable way
- Users should be able to filter events by prediction label and score (i.e. allow users to enter “cat” and 70, and then only show events with a prediction for “cat” with a score higher than 70).
- Your app should have basic styles that make interaction easy and intuitive. We don’t expect a work of art, but engineers at Matroid determine a big part of the look and feel of our web platform.

Bonus requirements

- Allow users to filter events by additional fields: e.g. video source and/or timestamp
- Give users the ability to indicate whether or not a prediction is correct, and update the UI accordingly
- Add more advanced styles
- Add basic tests

Submission details

- Submit your code as a zip file
- You can structure your project however you’d like, but there should be an index.html file in the root directory that when loaded into the browser loads your app. *If you’d like your code to be run another way, please include instructions in a readme file.*
- Feel free to use any frameworks and/or packages that you feel are right for the job. At Matroid we’re particularly partial to React. It’s also fine to just use vanilla Javascript.

Evaluation

Code quality

- Your code will be evaluated for clarity and good design choices. Feel free to add comments as necessary.
- Code style: your code should use standard Javascript conventions and indentation. Variables and functions should have appropriate and descriptive names. When in doubt, consult the Airbnb Javascript style guide.

Functionality & User Experience

- Your app should implement all basic functionality in a bug free way
- Your app should use good UX patterns and have basic styling