Technical Screening Test

This exercise is intended to allow you to demonstrate your skills in building a small JavaScript-based web API. Your submission will be assessed on the following criteria:

1. Grasp and understanding of the JavaScript language
2. Writing clearing, concise, maintainable code
3. Ability to verify the correctness of the solution

The requirements are split into

MUSTs: things we expect your solution to include,

SHOULDs: things we'd like to see you include, and

COULDs: are nice-to-haves and you can choose to do any, all, or none of them.

You are encouraged to treat this as a microcosm of a real project, so approach it as you would any other project. JavaScript is required, but other technologies and libraries are at your discretion and welcomed.

Please include a README with your submission which describes how to run the project and explains the approach you took to producing the solution.

Submit your solution by either sending a link to a GitHub or Bitbucket repository containing your code, or a link to a zipped copy of your code in Dropbox or similar.

Exercise description

Harrison.ai is designing an image labelling system which will help their medical doctors to assign disease categories to areas of the image. So that another system can use these labelled images to train machine learning models. You are creating an API to manage raw medical images and labelled images, allowing persisting and modifying data, as well as (optionally) an interface for users to manage the progress.

Your solution

1. **MUST** have an API server written in JavaScript

2. **MUST** have routes for raw medical images
3. **MUST** have routes for labelled images
4. **MUST** persist data and/or metadata to a database
5. **MUST** be secured against anonymous access
6. **MUST** contain tests using a testing framework
7. **SHOULD** be able to detect sensitive data (Personal Identifiable Information) which is part of the image
8. **SHOULD** track which user makes changes to the data
9. **COULD** have a UI (but don't worry about UX)
10. **COULD** support searching for labelled medical images by Date, Disease type, or Status