Full stack engineer test

Objective

Implement a web application that allows users to store, at later retrieve, files at a specified URL. The application should be composed of two distinct components - frontend JavaScript application, and a backend API.

Your assessment should be based on the Propylon Document Management Assessment bootstrap project found here:

https://github.com/propylon/document-manager-assessment

You can fork this repository and implement the features requested below.

The bootstrap project is a starting point, and the submission will be assessed based on completion of the requirements below as well as any tests, documentation, improvements that would accompany a professional software implementation.

Basic instructions are included on how to run the project, please augment this documentation with any additional steps for running your implementation as well as any extra context information you think would be important for another engineer to work with your code.

This submission must be completed using the frameworks defined in the bootstrap project, implementations utilizing other component frameworks will be rejected.

Requirements

Functional

- Stores files of any type and name
- Stores files at any URL
- Does not allow interaction by non-authenticated users
- Does not allow a user to access files submitted by another user
- Allows users to store multiple revisions of the same file at the same URL
- Allows users to fetch any revision of any file
- Demonstrate functionality that allows a client to retrieve any given version of document using a endpoint that implements a Content Addressable Storage mechanism.

Non-functional

- Demonstrate knowledge of best-practices in relation to unit testing.
- Clear documentation detailing how to build and run the frontend and backend.

Stretch

- Demonstrate basic read/write permissions enforcement on individual versions of documents.
- Create a UI for viewing differences in content between file versions.

Example

A user may submit the file "review.pdf" to the application, specifying "/documents/reviews/review.pdf" as the desired URL. The user later submits a new version of the file at the same URL.

The user can now retrieve the latest version of the file by accessing the document URL ("/documents/reviews/review.pdf"). The original version of the file can be accessed at the URL ("/documents/reviews/review.pdf?revision=0").

Timeframe

You can define the timeframe for returning this assessment within reason. We have found this exercise to typically take three to five days. It is more important that you commit to a delivery date and submit the assessment on time.

To coordinate this delivery date please reach out marina.saric@rws.com.

Delivery

The solution should be returned to Propylon in such a way that the commits that have been made to fulfil the requirements are visible. This can be done through a website like GitHub, or via an email zip archive or using other methods. If shared as a link to a GitHub repository this repository should be marked as private.

Unless otherwise noted, private repositories containing the assessment submission should be shared with the GitHub account that is included in the initial assessment email.

Submissions that do not contain a git history will be rejected. Submissions via a zip archive that contain all virtual environment dependencies or node dependencies will be rejected. Public/unprotected GitHub repositories will be rejected.

In the case of an emailed ZIP file, compress the project folder and send it to marina.saric@rws.com. Should your email client reject the ZIP for containing source code, please encrypt the ZIP with the password "propylon". This should bypass any checks by the mail server.