TRUENORTH | LoanPro

Code Challenge

Project Exercise

Arithmetic Calculator REST API

Implement a Web platform to provide a simple calculator functionality (addition, subtraction, multiplication, division, square root, and a random string generation) where each functionality will have a separate cost per request.

User's will have a starting credit/balance. Each request will be deducted from the user's balance. If the user's balance isn't enough to cover the request cost, the request shall be denied.

This Web application and its UI application should be live (on any platform of your choice). They should be ready to be configured and used locally for any other developer (having all instructions written for this purpose).

Entities:

- User
  - id
  - username (email)
  - password
  - status (active, inactive)
- Operation
  - id
  - type (addition, subtraction, multiplication, division, square_root, random_string) ○ cost
- Record
  - id
  - operation_id
  - user_id
  - amount
  - user_balance
  - operation_response
  - date

Tech Stack

- Node.js, Go, or Python
- Vue.js or React.js

Technical Requirements

- Use third-party operation for random string https://www.random.org/clients ●

All client-server interaction should be through RESTful API (versionated).

- Collection endpoints should be able to provide filters and pagination.
- Use a Bootstrap or Material Design library (CSS/Design Library) of your choice.
- Add automated tests such as Unit Test (whether for frontend or backend).
- Records should be soft-deleted only.

UI views:

- Login (and “sign out” button anywhere available for all session-required screens)
  - A simple username and password input form
- New Operation
  - An input form providing all fields to request a new operation on behalf of the current user
- User Records
  - Datatable of all operation records from the current user
    - Datatable should have pagination (page number and per-page option) and sorting
    - Datatable should have a filter/search input field for partial matches
    - Delete button to delete records

Even though all operations are pretty simple (addition, subtraction, multiplication…), they must be performed on the backend.

Key items to evaluate:

- Live version (providing credentials to access)
- Code quality (code conventions, design patterns, decoupling, and cohesion)
- Automated tests
- Web Security
- API design

Plus features:

- Use of Serverless Framework
- Use of AWS technologies
- Use of Vue.js
- Use of Go or Python

Provide basic setup instructions and all required API documentation in the repo README file. Frontend and Backend should be on separate stacks and repos. Deliverables should be the repo of both the frontend and backend and the URLs of live UI and API.