Code Challenge Notice, Instructions & Rules

Government Contact: Jason.p.taylor@gov.bc.ca

This notice is dated Wednesday, August 21st, 2023 at 9:00 am Pacific Time (the "Notice Date").

Congratulations - you are a Shortlisted Candidate eligible to participate in the Code Challenge.

Rules and Instructions

Please be advised of the following rules and instructions:

1. These code challenge rules and instructions apply only to Shortlisted Proponents and are part of the RFP.
2. Shortlisted Proponents will have two days from the Notice Date to complete the Code Challenge. The deadline to complete the Code Challenge in accordance with these rules is 9:00 am Pacific Time on Wednesday, August 23rd, 2023 (the "Deadline").
3. The Shortlisted Proponent's Code Challenge submission Deliverables (defined below) must be received by the Province (as provided for by these instructions) and be deposited and located in the applicable Repository before the Deadline, failing which such submission will not be eligible for evaluation and the associated Shortlisted Proponent Proposal will receive no further consideration and such Shortlisted Proponent will be eliminated from the RFP competition.
4. Only the Proposed Resource that was put forward in a Shortlisted Proponent's RFP Proposal is eligible to participate in the Code Challenge.
5. The Shortlist Proponent Proposed Resources will be sent invitations via GitHub to join this private repository.
6. As of the Notice Date, the Code Challenge issue has been created in this private repository under the "BC Dev Exchange-Code Challenge" organization.
7. Shortlisted Proponents may direct clarifying questions by creating an issue in the applicable GitHub Repository. Any such questions must be received by the Government Contact before 12:00 pm Pacific Time on Thursday, August 22, 2023 (the "Code Challenge Questions Deadline").
8. The Province reserves the right to amend this notice before or after the Closing Time, including changing the Deadline or the Code Challenge Questions Deadline upon notice to all Shortlisted Candidates.
9. The Shortlisted Proponent must complete all of the following tasks and the Deliverables (as that term is defined in the "Deliverables" section below) and as such they must be deposited and received in the applicable Repository by the Province in the form specified by this notice before the Deadline.
10. The rules and instructions set forth in this notice are in addition to any rules, terms and conditions set forth elsewhere in the RFP.

Introduction

This code challenge requires you to develop and deploy a web application that performs data validation and visualization on employee details from various industries. The web application should run in a containerized environment and adhere to modern best practices, such as the 12 factor app and the ability to build portable and scalable web applications.

Deliverables

As part of this Code Challenge, your team must deliver an application that includes the following features:

- A front end developed with Vue.js served by Caddy webserver (or equivalent);
- A back end developed using Express/Node.js; and
- A database that stores the input files as a normalized data set in tables in Postgres database (or equivalent SQL database).

(collectively, the "Deliverables")

The application must be developed using containerization (e.g., Docker, Podman) of the front end, back end and database.

You must provide the evaluation panel with "single command" build instructions using a compose specification (or equivalent). For instance, if you are using Docker as your containerization technology, you would provide the evaluation panel with a single-line "docker compose" command to spin up your containers.

Business case – Scenario

For this Code Challenge, you will have to assist an anonymous user in inputting data, validating data, and displaying graphical reports based on the data.

Your application does NOT have to provide user authentication or authorization.

User Stories

User Story #1

As an anonymous user, I want to be able to upload a CSV file into the application.

Given that I am an anonymous user

When I activate the application's "Enter Data" user interface element

Then I am directed to upload a CSV file from my local machine

The CSV files must include the following columns (and only these columns):

- Employee ID
- Company
- Industry
- Work Hours
- Vacation Hours

A sample CSV file will be provided.

Note: You do NOT have to support upload of any file format other than CSV files. Your application will be expected to handle any file format errors gracefully.

User Story #2

As an anonymous user, I want my uploaded CSV file to be validated to ensure proper formatting of input data.

Given that I am an anonymous user

When I upload a CSV file to the application

Then the application validates each element of the CSV file for proper formatting

And the application displays a list of any elements that could NOT be properly validated.

The required columns must be validated as follows:

- Employee ID – positive integer
- Company – String, max 64 characters
- Industry – String, max 64 characters
- Work Hours – positive integer
- Vacation Hours – positive integer

Note: You will NOT have to validate against missing column elements – that is, you can assume that each CSV row contains an entry for each of the required columns.

User Story #3

As an anonymous user who has entered properly formatted data, I want to be presented with three graphical reports with respect to the data.

Given that I am an anonymous user

And that I have inputted a properly formatted CSV file

When the application has validated my CSV file

Then the application generates and displays three graphical reports based on the input data:

- A bar chart that shows the average work hours per industry;
- A pie chart that shows the average vacation hours per industry; and
- A line chart that shows the average overtime hours per industry.

Note: For generating the graphical reports, please use an open-source library of your choice.

Technical Requirements

Your solution should demonstrate the following:

- Ability to work with open-source code.
- Ability to assess customer requirements, develop and implement technical and business solutions to address.
- Ability to coordinate resolution of technical and business problems while managing multiple tasks and priorities.
- Ability to develop source code with version control using Git.
- Ability to develop custom, modern web applications
- Ability to use scripting languages such as Java/JavaScript/PHP (but not limited to these)
- Ability to use Ansible/Terraform/docker-compose/podman-compose (but not limited to these) to automate regular tasks and build technical infrastructure.
- Ability to use and administer continuous integration and deployment systems.
- Ability to develop applications using test-driven development.

You must build a solution that can be spun up as a local development environment on your computer, and then on someone else's, as is described in https://12factor.net/dev-prod-parity. Your solution must be deployable locally, regardless of Linux/Unix host OS (Operating System), either as a Docker container platform or some portable, virtual software development environment. You must provide simple build/deploy instructions **to include a single-line CLI command** (e.g., "docker-compose up...", "podman-compose up..."). Evaluators will be testing the ability to build and deploy your solution locally with one or two well-documented commands. Again, make sure that you document how to build and deploy your solution.

Submission

1. Include all artifacts that are required to build and deploy the solution including code, Dockerfiles, other configuration files, etc.
2. Include all artifacts that would typically be included in the "Definition of Done", including demonstration of testing (full test suites are not required, but only a demonstration of how your team would test its solution).

3. Include a README file with instructions for building and deploying your solution. If the instructions are omitted or unclear, points will be deducted.

4. Submit artifacts by way of a pull request to the private repository. All artifacts must be committed before the Deadline.

5. Attach an Apache License 2.0 to your pull request.

Evaluation Criteria

Criteria	**Max Points**
Architecture and Technical Design	10
Code Quality and Maintainability	20
Build and Deploy Instructions	10
User Stories – Requirements Satisfied	25
Total	65