Code Challenge Notice, Instructions & Rules

Government Contact: Tahna.Neilson@gov.bc.ca

This notice is dated November 28, 2022 at 9 a.m. (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 Request for Proposals (RFP).
2. Shortlisted Proponents will have at least three (3) 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 a.m. Pacific Time on Thursday, December 1st, 2022** (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 Proponent Resources that were put forward in a Shortlisted Proponent's RFP Proposal are eligible to participate in the Code Challenge.
5. The Shortlisted Proponent Resources will be sent invites 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 "BCDevExchange-CodeChallenge" 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 4:00 p.m. Pacific Time on Tuesday, November 29th, 2022** (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 the following tasks and the Deliverable 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:
   a. Complete all code changes required to complete the code challenge (the "Deliverable"); and
   b. Attach an Apache License 2.0 to Deliverable.
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.

Deliverables

Introduction

This code challenge asks you to build an Open Geo-spatial Consortium (OGC) compliant web map application that displays road events, webcam images, a travel advisory message and allows for email notifications for the users.

The solution should demonstrate modern best practices, such as the 12 factor app, and the ability to build modern web applications. The solution does not need to be pretty (you will not be evaluated on subjective aesthetic concerns), but it does need to work, have a focus on security, test-driven development and building the necessary architectural components for deployment on another machine.

Business case:
DriveBC is the provincial traveller information system. It is used to inform travellers of events that may impact their travel along a provincial highway. Travellers will need to view current and geo-spatially accurate information on a map, create routes and receive notifications of any new events along that route.

Description:
The web map-based application should allow map navigation and routing (using listed third-party APIs) as well as allowing users to view, save, and delete routes they create and receive email notifications of events along that route.

API Documentation References:

- **Road Event API** – https://tst-api.open511.gov.bc.ca/help
- **Webcam Image API** – https://tst-images.drivebc.ca/webcam/api/v1/webcams
- **BC Route Planner REST API** – https://www2.gov.bc.ca/gov/content/data/geographic-data-services/location-services/route-planner (Use the following API key: GIPS68ey7YBSIs22RPYJo2lGZ1ASxfSf or apply for your own API key via https://api.gov.bc.ca/devportal/api-directory)

High Level Requirements

1. Build a new web application.
2. Use standard REST API requests.
3. Build a local development environment that can be deployed on another machine with a one or two-line command.
4. Include a backend, database and programmatically perform CRUD operations on the database.
5. Include an embedded map that allows the visualization of geo-spatial information.

Assumptions

1. A simple authentication process should be used and this could even be hard-coded. The authentication process will not be evaluated as part of the scoring.
2. The Spatial Reference Identifier (SRID) of the web-based map must be EPSG:3857.
3. The route is persisted in the database as a geometry object.
4. All data exchanges happen through REST APIs.
5. Events/Webcams related to road segments may not coincide directly with the road segment geometry
6. An email emulator (e.g., Mailtrap, MailHog, MailSlurper) should be used to provide notifications. The ability to interact with an SMTP server will NOT be evaluated.
7. The evaluation will test mobile functionality using an emulator.

Ministry Team Support

The evaluation team will be available to enter test Road Events from 9 a.m. to 4 p.m. Pacific Time on November 28, 29 and 30, 2022. The proponent should provide information on where they would like these Events to be added and on what frequency. To make a Road Event request, the proponent shall create an issue in the GitHub repository. The issue header shall begin with “Support Request – Road Event”.

The APIs will be available outside of the firewall for the duration of the Code Challenge. These APIs are not throttled and the Ministry expects appropriate etiquette when interacting with them. The evaluation team will be monitoring the servers throughout the Challenge and will be available for support from 9 a.m. to 4 p.m. Pacific Time on November 28, 29 and 30, 2022. If a proponent requires firewall or server support, the proponent shall create an issue in the GitHub repository. The issue header shall begin with “Support Request – Firewall/Server”.

User Stories

The following user stories must be completed.

#1

As a user, I want to navigate a web-based map interface on my mobile device (e.g., pan/zoom), so that I can click on map layer features (Events and webcams) to reveal information pop-ups, switch between base layers, and turn on/off these layers.

Given that I am an unauthenticated user

And I'm on the web page in a mobile emulator that presents me with a web map interface

When I complete the following actions:

• Zoom to current location

• Drag and pan the map

• Scroll the map window

• Click on a layer feature on the map

• Switch between base layers

• Turn on/off Road Event and Webcam images overlay layer(s)

Then the map responds appropriately (e.g., the map moves commensurate with the drag/pan action, increases/decreases magnification, displays information about the clicked-on layer feature (in a popup), and makes visible/hides map layers).

#2

As a commercial truck driver, I want to enter my starting location and destination (by dropping pins) on a map to plan my route so that I will be notified by email of any new road events that occur along that route.

Given that I am an unauthenticated user

And I create and save a route along with my email address

When A new road event is created along that route

Then I should receive an email notification whenever a new road event occurs along that route

#3

As a DriveBC administrator, I want to create and save a traffic advisory message that I can publish to the application so that the message is visible to the public.

Given that I am an administrator

And I have created and saved a draft travel advisory message

When I publish the travel advisory message

Then the message will be displayed on the map in a side-bar panel

[Please note that messages need to be created and published only – you do NOT need to create the functionality to either edit or delete messages.]

Technical Requirements

Expected pieces of the architecture are at minimum a 1.) front-end, 2.) a database and 3.) backend.
Solutions may be developed using (but not limited to) the following preferred tech stack:

1. Backend: .NET 6 or PHP
2. Database: Relational database (your choice of MS SQL Server, PostGreSQL/PostGIS, MariaDB)
3. Front-end: A modern JavaScript framework (e.g., React, Bootstrap, JQuery) including a Geospatial JavaScript framework (e.g., OpenLayers, Leaflet).

Please note that the listed stack technologies are preferred but NOT required. Proponents will NOT lose points in this evaluation if they use a technology other than the preferred technology.

Your solution should demonstrate the following:

- Use of Open Geospatial Consortium (OGC) compliant frameworks and services.
- Integrate with BC Route Planner REST API and manipulate and display results.
- Build a local development environment that can be deployed quickly/easily on another machine.
- Containerized development and release management.
- Illustrate dynamic updates without complete page refreshes.
- Ability to work with open-source code.
- Ability to assess customer requirements, develop and implement technical and business solutions to address them.
- Ability to develop source code with version control using Git.
- Ability to develop custom, modern web applications.
- Ability to use Ansible/Terraform/Docker (but not limited to these) to automate regular tasks and build technical infrastructure.
- Ability to develop applications using test-driven development.

- Ability to interpret client requirements and synthesize this with knowledge of underlying infrastructure applications, systems, and processes, and determine how these contribute to system design plans and development efforts over the lifecycle of a development project.
- Ability to coordinate resolution of technical and business problems while managing multiple tasks and priorities.

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 host OS (Operating System), either as a Docker container platform or some portable, virtual software development environment (such as Vagrant).

Provide simple build/deploy instructions - a single-line CLI command is preferred (`vagrant up...`, `terraform apply...`, `docker run...`, `docker-compose up...`, etc.). Evaluators will be testing the ability to build and deploy your solution locally with a few, well-documented, commands. Be sure to document how you built and deployed the app.

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	**Max Points**
Meets requirements	15
Architecture and technical design	6
Code quality and maintainability	6
Build & deployment instructions and documentation	3
Total	30