Co-op - Application Developer, Requisition # 101439

Please answer ALL the following questions and return as directed in the covering email.

1) This position is a co-op opportunity based out of BC Registries' Office in downtown Victoria, BC, please confirm that you can work from the office (or in a location in British Columbia) for the duration of this co-op.

2) When answering the following seven questions, the following should give you context.

BC Registries and Online Services provides trusted registry services including the Corporate Registry, Personal Property Registry, Manufactured Home Registry, OneStop Business Registry and BC Online. We also have an important mandate to deliver secure and privacy-enhancing identity services for citizens and businesses, through the Provincial Identity and Information Management Program, to support access to digital government services and information.

To ensure fairness, no clarifying questions will be answered about this test. If you need to, you can add one or two sentences called "ASSUMPTIONS" at the top of any answer that will help the panel with the marking. Anything labeled under the assumptions heading will not be marked or counted in your score, either positively or negatively.

EXAM

1. You are responsible for implementing a system to communicate changes in the British Columbia corporate registry to external agencies such as the Federal government. When an update is made by staff or a member of the public to the Corporate Registry, the system must automatically push that information to those external agencies. Ideally the system support would be handled by the support group, who are comprised of help desk personnel (not developers or database administrators.) In no more than 250 words, describe the key considerations for this system and what functionality would be necessary to ensure partners receive the updates they need, that the system is fault tolerant and that support personnel could maintain the day-to-day operations of the system.

2. You are responsible for designing a table structure for the following data:

Corporation Name	Mailing Address	Director	Assumed Name
ABC Company	123 Main St	John Doe	ABC Boxes
ABC Company	123 Main St	John Doe	ABC Packing Tape
ABC Company	123 Main St	Jane Smith	ABC Boxes
ABC Company	123 Main St	Jane Smith	ABC Packing Tape
DEF Company	456 First Ave	Fred Smith	
GHI Company	789 Small St	Ellen Roberts	GHI Data Analytics
JKL Company	987 Big Ave	Trish Murray	

- Each Corporation must have
    - 1 Name
    - 1 Mailing Address
    - 1 or more directors
        - Each director can belong to exactly one corporation
- Each Corporation may have 0 or more assumed names

Your table definitions should follow the format below:

<u>Table1</u>
Field1 Integer Primary Key
Field2 Text
Field3 Text
<u>Table2</u>
Field1 Integer
Field2 Text
Field3 Integer Foreign Key from Table 1

NOTE: You will be marked on the table structures and organization, not the SQL required to generate the tables.

3. You have been given a task to write a GET request endpoint for a public RESTful API that needs to return some protected user data. The new endpoint in the API is already setup to accept the request, however, so far it only returns a 501 status code. In 250 words or less, describe the steps you would take and questions you would ask to troubleshoot the issue.

4. An application has the following micro services: one frontend UI, one backend API, one database, one queue, one queue listener and one cronjob. The API reads / updates the database and adds messages to the queue in response to requests from the UI. The queue listener reads new messages put on the queue, reads / updates the database, and then sends out notification via email / text message. The cronjob activates every hour, reads data in the database, and adds messages onto the queue. The follow of the application is as follows:
    - The user updates their items via the UI
    - The UI submits the changes to the API

- The API saves the changes with a pending status to a todo table in the database and puts a message on the queue containing the id of the todo row
- The queue listener reads the message off the queue and gets the todo item from the database and attempts to apply the changes of the todo item
- If successful, it updates the transaction table in the database, sets the todo item status to ‘complete’ and sends an email / text to the user
- If unsuccessful, it sets the todo item status to ‘failed’
- The cronjob pins up, looks in the todo table for any items with a ‘failed’ status and puts a message back on the queue for each one

Draw and label a diagram of this architecture.

5. You are monitoring the application from question 4. A user has called in complaining that they updated their items today but haven't received an email or test notification that the system completed the changes, and when they go back to make further changes, they see that their items are still in the same state as yesterday (no changes to their items are getting applied.)

   You can assume that the back API logs the status code and the user id for every request, the queue has logs for every added message, the queue listener has no logs, the cronjob has logs for every item it reads from the database, but not what it publishes. The database has a transaction table that is updated after every update from the API and queue listener that contains the user id, transaction id, toto item id and time.

   You have been given the userid of the troubled user and have been asked to figure out why they are having issues. In 250 words or less, in order of priority, list and describe each step you would take to figure out the issue. Assume that there could be multiple errors.

   Example:
   1. Check X to see if…
   - If X is true then I know that…
   2. Assuming I didn't find anything with step 1…

6. This is a code reading exercise:

   Your co-worker has been assigned to implement a feature that checks if a document is valid and is able to be submitted. If the document is not valid then it scrolls to a displayed invalid section in the document.

   Your task is to complete a code review on the following code that your co-worker has submitted. Please indicate any issues that you find to your co-worker's code and make suggestions on how they could fix the code.

Preconditions:

There are 9 components, all components are on all 4 document types. The components parameter will always be the following.

components = { 0: "effective-date-time", 1: "document-delivery-section",

2: "completing-party-section", 3: "transactional-folio-number-section", 4: "detail-section",

5: "special-resolution-confirm", 6: "certify-section", 7: "court-order", 8: "payment" }

The flags will be in the order given below. However, they may have different Boolean values. Below is an example of the flags parameter.

flags = { isValidEffectiveDate: true, isValidDocumentOptionalEmail: true, isValidCompletingParty: true,

isValidTransactionalFolioNumber: true, isValidDetailComment: true,

isValidSpecialResolutionConfirm: true, isValidCertify: false, isValidCourtOrder: true,

isValidStaffPayment: true }

There are four document types. They each have all 9 components and require all components to be valid for the document to be valid. Only the visible components are show below with each document type and are listed in order

Alteration: "effective-date-time", "document-delivery-section", "completing-party-section",

"certify-section", "court-order", "payment"

Change: "document-delivery-section", "completing-party-section",

"transactional-folio-number-section", "certify-section", "court-order", "payment"

Conversion: "certify-section"

Special Resolution: "document-delivery-section", "transactional-folio-number-section",

"special-resolution-confirm", "certify-section", "payment"

Other Assumptions that you can make.

- Flags will always be true or false.

- Every flag is used for all documents.

- All flags are required to be true for the document to be valid.

- documents.getElementById() takes an id string will return the id if it is a visible component in the document otherwise it will return null.

- document.scrollTo() takes an id string and scrolls to the component on the document if it is visible otherwise it does nothing.

Limit 250 words.

```
/**
 * @param flags list of current component validity flags
 * @param components list of current component IDs
 * @return whether all components are valid
 */
function validateAndScroll(flags, components) {
    // Create an array of the _ordered_ validity flags
    var valid_flag_array = Object.keys(flags).map(function (key) { return
flags[key]; });

    var invalidComponent =
document.getElementById(components[valid_flag_array.indexOf(false)]);

    if (invalidComponent) {
        console.log('What is happening!!!!', invalidComponent);
        document.scrollTo(invalidComponent);
        return false;
    }
    return true;
}
```

7. Assume you are responsible for creating automated tests on the function that was used in the previous question.

   What type of testing would be appropriate for testing the previous code? Please us pseudo code to explain how you would test the function.

   YOU DO NOT HAVE TO WRITE CODE!