**The multiverses are converging, and we’re hiring!**

In an effort to hire the best and the brightest talent not only in this universe-ities but in others, the Public Service Commission is looking at implementing a new recruitment solution that will allow talented job-seekers to apply to government jobs no matter where they are. The challenge is substantial: we need to hire people across multiple Canadas, keep track of the special skills needed to provide government services in parallel universes, and make sure anyone can apply, no matter the quality of their internet connection in their own reality. Time is running out – we needed this yesterday, and we haven’t perfected time-travel yet.

**You are tasked to create a web application that will implement the features in the table below**

**Technological Stack**: Use the technology of your choice as long as the application is available online and compatible with the latest version of Chrome or Firefox. The web application will be used to validate the implementation of each feature. Source code also needs to be available online (ex. Github)

You need to provide a diagram that outlines the technologies that you have decided to use to build your applications (Language, Database, Scripting, etc..)

In order to help you start the PSC has provided a JSON data structure

**Functionality list, with associated points:**

|  |  |  |
| --- | --- | --- |
| **Rated Functionalities expected to be included in your application** | **Points  (1,2,3,5,8,13)** | **Implemented  (Yes/No)** |
| Display, add, remove and update job-seeker profile (see JSON) | 13 |  |
| Display, add, remove and update job description (see JSON) | 13 |  |
| Display available jobs on a map according to their addresses | 8 |  |
| Display job-seekers on a map according to their addresses | 8 |  |
| Identify the job closest to a given job-seeker | 5 |  |
| Job-seeker account management (Sign up, add and modify):   * job-seeker information * job-seeker skills | 8 |  |
| Allow job-seekers to upload a résumé to their profile | 5 |  |
| Search feature:   * Job-seeker * Profiles * Job description | 8 |  |
| Browse  Jobs, résumé, job-seekers | 8 |  |
| Job listing should allow sorting and filtering (Date posted, Salary, etc..) | 5 |  |
| Accessibility: Application need to be compatible with WCAG 2.0 level A | 8 |  |
| Job Matching :   * Identify potential matching job-seekers for each job offering * Identify Job opportunity for each job-seeker | 13 |  |
| Ability to share Job description on Social Media | 5 |  |
| Authentication for system users (job-seekers and hiring manager): Username and Password | 8 |  |
| Log-in to include Captcha like feature | 5 |  |
| Self-serve password recovery feature | 5 |  |
| UI must be available in both official languages (French / English) | 8 |  |
| Development stack diagram | 8 |  |
| Code quality: up to 8 point based on inline comments, documentation, variable names, ease of reading | 8 |  |
| 5 min. application recap video available online | 5 |  |

Data structure:

Candidate

[

{

**"id"**:Interger, //Unique ID

**"first\_name"**:"String", //First name of candidate

**"last\_name"**:"String", //Last name of candidate

"**universe\_permits**":Array of Integer, //Array listing all Universe’s this candidate can apply

**"willing\_to\_do\_overtime"**:Bool, // Willing to do overtime

**"willing\_to\_relocate\_to\_other\_universe"**:Bool, //Willing to relocate to another univers

**"home\_address"**:"String", // Address in String unformatted

**"education\_level"**:Integer, // ID education level

**"skills"**:"Array of Strings" // All skills tied to this candidate

}

]

Job Posting

[

{

**"id"**:Integer, // Unique ID

**"description\_en"**:"String", // English description

**"description\_fr"**:"String", // French description

**"salary"**:Interger, // Job postings salary

**"work\_location"**:"String", // Address in String unformatted

**"work\_universe"**: Integer, //Universe ID

**"required\_skill"**: "Array of String", // All skills required for this Job Posting

**"date\_posted"**:"Date", // Date

**"education\_level"**:Integer} // ID minimum education level required

}

]

Universe

[

{

**"id"**:Integer, // Unique ID

**"Universe"**:"String" // Name of Universe

}

]

Education

[

{

**"id"**:Integer, // Unique ID

**"Education\_Level "**:"String" // Name of Education level

}

]

Expected deliverables:

* Application URL accessible online
* Application source code
* Dev Stack Diagram
* List of completed feature from the functionality list
* 5 minutes application recap video