PHP Senior Backend Developer Project

In this project, we expect you to complete a matching system for recommending most relevant job positions to candidates. In this matching system, there will be a group of job positions to be used for matching. Your task is to implement a search api for getting search filters and sort items like age, education, ... and return matched positions as result.

Matching (Searching) Rules:

we have 4 kind of inputs for searching api and all of them can be nullable:

1. Terms level rules:
   1.1 these fields must be exactly match with positions:
      Category,
      Location,
      Education,
      Gender

2. Match level rules:
   2.1 This field should be like the inputs (it is not exact word search):
      Title

3. Sort level rules:
   Salary,
   Lived_at

4. Range filters:
   Expired_at,
   Created_at,
   Age

Technical Rules:

1. You must implement this project in php Laravel framework.
2. You can choose a database for storing data from mysql or mongodb by your own.
3. Implement crud apis for position model.
4. You must use elastic search to index positions after inserting or updating records.
5. you must implement seeders for adding at least 500 records to position model and also index inserted records with elastic search.
6. you must design, implement and document an api for searching as mentioned previously in matching rules section.
7. Matching algorithm must be implemented in elastic search calling from Laravel project.

There will be a positions table with following columns:

Id: e.g. 1,2,3

Title: e.g. php developer, translator, ui/ux designer, sales manager, marketing manager and …

Category: string e.g. It, Designer, accounting, health care, Laboratory, lawyer, social service, security service

Min_Age: (nullable) int e.g. 18

Max_Age: (nullable) int e.g. 40

Education: string e.g. BSc. MD, PhD,diploma

Gender: (nullable) stirng male/female

Salary: (nullable) e.g. 2000000

Location: (nullable) string e.g. Tehran, Isfahan, Iran and …

Created_at: datetime

Expired_at: datetime

Lived_at: datetime

• Also you can design your fields or change field types if needed

==Extra==s (Including the following extras to your project will be regarded as a strong plus):

1. use docker and docker-compose to run project
2. Implementing api for getting job position reference and return most relevant candidate

This project has a time limitation of 72 hours so please send your implemented project before expiration in any way you desire like pushing project in github or gitlab.

The most important features should be as bellow:

1. Inserting and syncing data with your database and elastic search
2. Implement crud apis for position model
3. Implement searching api

Good luck