

Pravimo deo HR platforme za dodavanje i praćenje kandidata za posao i njihove veštine kako bi olakšali HR procese u kompaniji.

Zadatak je implementacija C# .NET 5 projekta za dodavanje, ažuriranje, brisanje i traženje kandidata za posao i njihove veštine (osnovne CRUD operacije).

Evo liste zahteva koje treba uzeti u obzir:

- Svaki kandidat za posao ima sledeće attribute: ime (puno ime), datum rođenja, kontakt broj, e-mail i spisak veština koje poseduje; veštine poput Java programiranje, C# programiranje, dizajn baze podataka, engleski, ruski, nemački jezik

...

- Svaka veština ima svoje ime

Zadatak:

1) Kreirajte model baze podataka i odgovarajuće tabele za kandidate za posao i veštine (možete koristite bilo koju relacionu bazu podataka koju želite)

2) Kreirati odgovarajući model i klase usluga koje predstavljaju ove funkcionalnosti

3) Izložite ove funkcionalnosti preko REST veb servisa sa sledećim operacijama:

dodajte kandidata za posao, dodajte veštine, ažurirajte kandidata za posao veštinom, uklonite veštinu sa posla

kandidat, ukloniti kandidata, pretražiti kandidata po imenu i/ili datim veštinama

4) Dodatni poeni za jedinične testove za servisni sloj

5) Dodatni poeni za jedinične testove za API-je za odmor

6) Dodatni poeni za implementaciju razmetanja

7) Dodatni poeni za implementaciju korisničkog interfejsa preko React 16+"

We are creating a part of an HR platform for adding and monitoring job candidates and their skills in order to facilitate HR processes within a company. The task is to implement C# .NET 5 project for adding, updating, deleting, and searching job candidates and their skills (basic CRUD operations). Here is list of requirements that should be taken into consideration: • Each job candidate has the following attributes: name (full name), date of birth, contact number, E-mail and the list of skills which he/she possesses; skills like Java programming, C# programming, Database design, English, Russian, German language ... • Each skill has its name Task: 1) Create database model and appropriate tables for job candidates and skills (you can use any relational database you prefer) 2) Create appropriate model and service classes which represent these functionalities 3) Expose these functionalities through REST web services with following operations: add job candidate, add skills, update job candidate with skill, remove skill from job candidate, remove candidate,

search candidate by name and/or given skill(s) 4) Extra points for unit tests for service layer 5) Extra points for unit tests for rest APIs 6) Extra points for implementing swagger 7) Extra points for implementing UI via React 16+"