

Zadanie rekrutacyjne

Przygotuj moduł kontraktów do systemu HR. Moduł ma umożliwiać:

1. dodawanie kontraktu,
2. przeglądanie listy stworzonych kontraktów.

Wymagania dotyczące dodawania kontraktu:

1. Kontrakt składa się z następujących elementów (wszystkie są wymagane): nazwa, typ (Programista, Tester), doświadczenie w latach, stawka (zł netto).
2. Stawka jest wyliczana na podstawie polityki płac.
3. Stawka jest widoczna dla użytkownika podczas dodawania kontraktu.
4. Wyliczanie stawki ma być asynchroniczne (Ajax).
5. Walidacja pól ma być asynchroniczna (Ajax).

Reguły dotyczące polityki płac:

Typ kontraktu	Doświadczenie w latach	Minimalna stawka
Programista	<1;3)	2500
Programista	<3;5>	5000
Programista	>5	5500
Tester	<1;2)	2000
Tester	<2;4>	2700
Tester	>4	3200

Sposób wyliczenia stawki:

- Programista: minimalna stawka + (lata doświadczenia * 125)
- Tester: minimalna stawka + (lata doświadczenia * 100 + (minimalna stawka/4))

Wymagania dotyczące listy stworzonych kontraktów:

1. Lista jest stronicowana (10 elementów na każdej stronie).
2. Lista zawiera kolumny: Nazwa, Typ, Doświadczenie, Stawka.
3. Po każdej kolumnie można filtrować (można filtrować po kilku kolumnach jednocześnie).
4. Nad listą znajdują się 2 dodatkowe filtry:
 - a. Programiści z 5 letnim stażem
 - b. Kontrakty ze stawką >5000
5. Przeładowanie grida ma się odbywać asynchronicznie.

W rozwiązaniu wykorzystaj m. in. następujące technologie:

1. Entity Framework w podejściu Code First
2. Dependency Injection (preferowany Autofac)
3. ASP.NET MVC 5
4. AngularJs
5. MSSQL Server

Dodatkowe wytyczne:

- **zadbaj o architekturę rozwiązania,**
- **przestrzaj zasad OOP i SOLID,**
- **odseparuj logikę domenową (polityka płac) od części aplikacyjnej,**
- stwórz repozytorium Git na githubie lub bitbucket i prześlij link do rozwiązania,
- **skup się przede wszystkim na stronie serwerowej (przygotuj API),**
- akceptowane formy prezentacji rozwiązania: testy jednostkowe/integracyjne lub UI