**ფიზიკური პირების ცნობარი**

ფიზიკური პირის მოდელი:

* იდენტიფიკატორი (ავტომატურად გენერირებული, უნიკალური, დადებითი მთელი რიცხვი)
* სახელი (ტექსტური, სავალდებულო, მინიმუმ 2 და მაქსიმუმ 50 სიმბოლო, უნდა შეიცავდეს მხოლოდ ქართული ან ლათინური ანბანის ასოებს, არ უნდა შეიცავდეს ერთდროულად ლათინურ და ქართულ ასოებს)
* გვარი (ტექსტური, სავალდებულო, მინიმუმ 2 და მაქსიმუმ 50 სიმბოლო, უნდა შეიცავდეს მხოლოდ ქართული ან ლათინური ანბანის ასოებს, არ უნდა შეიცავდეს ერთდროულად ლათინურ და ქართულ ასოებს)
* სქესი (დასაშვები მნიშვნელობები: ქალი, კაცი)
* პირადი ნომერი (ტექსტური, სავალდებულო, 11 ციფრი)
* დაბადების თარიღი (თარიღი, სავალდებულო, მინიმუმ 18 წლის)
* ქალაქი (ქალაქის იდენტიფიკატორი, ქალაქების ცნობარიდან)
* ტელეფონის ნომრები
  + ნომრის ტიპი (დასაშვები მნიშვნელობები: მობილური, ოფისის, სახლის)
  + ნომერი (ტექსტური, მინიმუმ 4 და მაქსიმუმ 50 სიმბოლო)
* სურათი (ტექსტური, ფაილის relative მისამართი ფაილურ სისტემაში)
* დაკავშირებული ფიზიკური პირები
  + კავშირის ტიპი (დასაშვები მნიშვნელობები: კოლეგა, ნაცნობი, ნათესავი, სხვა)
  + დაკავშირებული ფიზიკური პირის იდენტიფიკატორი

ამოცანა:

* უნდა შევქმნათ ფიზიკური პირების ცნობარის API შემდეგი ფუნქციებით:
  + ფიზიკური პირის დამატება
  + ფიზიკური პირის ძირითადი ინფორმაციის ცვლილება, რომელიც მოიცავს შემდეგ მონაცემებს: სახელი, გვარი, სქესი, პირადი ნომერი, დაბადების თარიღი, ქალაქი, ტელეფონის ნომრები
  + ფიზიკური პირის სურათის ატვირთვა/ცვლილება (სურათი შევინახოთ ფაილურ სისტემაში)
  + ფიზიკური პირის დაკავშირებული ფიზიკური პირის დამატება/წაშლა
  + ფიზიკური პირის წაშლა
  + ფიზიკური პირის შესახებ სრული ინფორმაციის მიღება იდენტიფიკატორის მეშვეობით (დაკავშირებული ფიზიკური პირების და სურათის ჩათვლით)
  + ფიზიკური პირების სიის მიღება, სწრაფი ძებნის (სახელი, გვარი, პირადი ნომრის მიხედვით (დამთხვევა sql like პრინციპით)), დეტალური ძებნის (ყველა ველის მიხედვით) და paging-ის ფუნქციით
  + რეპორტი თუ რამდენი დაკავშირებული პირი ჰყავს თითოეულ ფიზიკურ პირს, კავშირის ტიპის მიხედვით
* მონაცემები უნდა შევინახოთ მონაცემთა ბაზაში
* API-ის ყველა ოპერაციის შესრულების დროს უნდა მოხდეს შესაბამისი მონაცემების სტრუქტურის და მთლიანობის ვალიდაცია. შეცდომის შემთხვევაში შესაბამისი მესიჯის დაბრუნება. მესიჯები უნდა იყოს მარტივად ლოკალიზებადი.
* შევქმნათ საერთო Action Filter რომელიც გადაამოწმებს მოთხოვნის მონაცემებს და თუ არ არის ვალიდური დააბრუნებს შესაბამის შეცდომას
* API middleware-ის შექმნა დაუმუშავებელი შეცდომების ლოგირებისთვის
* API middleware-ის შექმნა მოთხოვნის Accept-Language HTTP header პარამეტრის შესაბამისი ლოკალიზაციის/ენის დაყენებისთვის
* სასურველია Repository და Unit of work პატერნების გამოყენება

გამოყენებული ტექნოლოგიები:

* ASP.NET MVC (სასურველია ASP.NET MVC Core)
* Entity Framework (სასურველია Entity Framework Core)
* Microsoft SQL Server

მონაცემების მოწოდება:

* სორს კოდი (პროექტის სოლუშენი)
* მონაცემთა ბაზის შექმნის და საწყისი მონაცემების ინიციალიზაციის სკრიპტი

გთხოვთ ფაილ(ებ)ი ატვირთოთ Google Drive-ზე და ლინკი გამოგვიგზავნოთ, ან რომელიმე public repository-ში (Github, Bitbucket) ატვირთეთ კოდი და repository-ის ლინკი გამოგვიგზავნეთ.