

Evidencia domácich zvierat

Je potrebné vytvoriť jednoduché softvérové riešenie pre evidenciu domácich zvierat. Budeme evidovať jednotlivých majiteľov a k nim priradovať ich domáce zvieratá – psy a mačky. Jeden majiteľ môže vlastniť viac mačiek a psov. Jeden pes alebo mačka môže byť vlastnená viacerými majiteľmi. Majiteľom a zvieratám pridajte vlastnosti podľa vlastného uváženia, avšak zvieratá musia mať spoločnú vlastnosť „Počet kŕmení“ a „Dátum narodenia“. Pri psoch budeme evidovať aj parameter „Úroveň výcviku“ (číslo 1 – 10) a „Predpokladaný vzrast“ v centimetroch, pri mačkách budeme zasa evidovať parameter „Chytá myši“ (boolean).

Zadanie:

1. Graficky navrhnuť databázový ER diagram pre túto situáciu (MSSQL) + vytvorenie DB a tabuliek. Naplniť do DB zoznam 10 psov, 10 mačiek, 5 majiteľov a vzájomne prepojenia.
2. Na serveri vytvoriť C# model tried pre túto situáciu. Teda trieda majiteľa so zoznamom jeho zvierat. Tento model tried bude použitý po načítaní dát z databázy – do tohto modelu sa načítané dáta sparsujú a naplnené objekty sa odošlú do frontendu kvôli zobrazeniu.
3. Navrhnuť GUI v REACT.JS – na úvodnej obrazovke bude zoznam majiteľov – načítaných z DB. Po kliku na majiteľa zobraziť podrobné informácie o majiteľovi (meno, vek,... + priemery vek jeho zvierat) a vykreslí sa aj zoznam jeho psov a mačiek (nech je osobitná komponenta na psa a na mačku). Pri každom zvierati zobraziť aj jeho vek vo formáte napr. „6 rokov 2 mesiace 8 dní“ (nie len dátum narodenia z DB) a počet kŕmení + ďalšie vlastnosti z DB.
4. Každý pes a mačka bude mať v sebe tlačidlo „Nakŕmiť“, týmto sa inkrementuje počet nakŕmení daného zvieratá. Toto sa odošle na server a uloží sa to aj v DB.
5. Každý majiteľ bude mať v sebe checkbox. Majiteľov teda bude možné označovať. Pod zoznamom majiteľov bude tlačidlo „Zobraziť informácie“, po kliku na toto tlačidlo sa zobrazí počet označených majiteľov, celkový počet zvierat, priemerný počet zvierat na majiteľa a priemerný vek zvierat od označených majiteľov.

Technológie: typescript, react (ideálne class components), .net core, react bootstrap (optional), pri komunikácii so serverom používať obyčajný fetch z javascriptu – buď GET alebo POST. Projekt vytvoriť vo VS2019 – založiť .NET Core projekt s reactom. Na prácu s DB použiť čisté SQL príkazy s parametrami (**nie entity framework**).

Dbať na

- optimalizáciu – počet požiadaviek na server, optimalizácia SQL, zbytočne renderovanie vo frontende
- kvalitu návrhu štruktúry aplikácie a databázy, kvalitu návrhu react componentov
- korektne a jednoduché zobrazenie aj na mobile
- nemať duplicity v kóde
- **NA GRAFIKE FRONTENDU AŽ TAK NEZÁLEŽÍ** – nie je potrebné sa nejak zdržiavať farbami a pod. Dôležité je pri grafike len to, aby to bolo optimalizované aj na mobily - nech sa zobrazenia neprekrývajú a podobne.

Ak kvôli časovej náročnosti zadanie nedokončíte, tak to až tak nevádi. Dôležitá je kvalita a nie kvantita.