

Opis

Zadaniem jest stworzenie systemu ankietowego. System ten ma umożliwić definiowanie ankiety z następującymi typami pytań:

1. Pytanie tekstowe otwarte – ankietowany wprowadza tekst o maksymalnej długości 250 znaków.
2. Pytanie liczbowe – ankietowany wprowadza wartość liczbową, np. osiągnięty dochód za ubiegły rok.
3. Pytanie logiczne – TAK/NIE – ankietowany wybiera określoną wartość

Pytania mogą być grupowane w strony. Wszystkie pytania mają być traktowane jako obowiązkowe.

Na początek warto się zastanowić nad schematem ideowym bazy danych – tutaj mam na myśli rozplanowanie encji oraz ich właściwości. Przykładowo: encja pytanie ma 3 rodzaje – tekstowe, liczbowe, oraz logiczne. Jakie mają one właściwości wspólne? Po czym rozróżnimy ich rodzaje?

Na podstawie schematu ideowego warto opracować schemat bazy danych – tabele, ich kolumny, oraz relacje między nimi. Tutaj należy się zastanowić nad takimi kwestiami jak klucze, indeksy, oraz ich implikacje dla działania bazy oraz jej wydajności.

Frontend powinien umożliwić użytkownikowi interakcję z systemem. Do przygotowania 3 widoki: tworzenie ankiety, wypełnianie ankiety, podgląd wyników (może być prosty, np. w formie pogrupowanej listy odpowiedzi na każde pytanie). Warto zacząć od zaprojektowania prostych makiet interfejsu w dowolnym narzędziu (dowolnie, od MS Paint do Figmy).

Backend powinien umożliwiać realizację funkcjonalności frontendu poprzez API REST. Warto rozplanować styki API i modele danych do tego API. Będą potrzebne zarówno we frontendzie jak i backendzie.

Przy planowaniu implementacji backendu warto wybrać i stosować wzorzec projektowy (np. w przypadku ASP.NET – MVC i Dependency Injection, ale nie tylko – są również inne przydatne przy tego rodzaju problemu), a co za tym idzie, zaplanować w jaki sposób zorganizowana zostanie implementacja komponentów. Podobny proces jest wskazany przy planowaniu frontendu.

Na razie kwestie związane z uwierzytelnianiem i autoryzacją są do pominięcia.

Sugerowane technologie:

- Frontend – Angular
- Backend - .NET Core (ASP.NET Core)
- baza danych – relacyjna (np. SQLite)

Sugerowane zastosowanie ORMa, np. Entity Framework Core.

Dowolność jeśli chodzi o wykorzystanie wzorców projektowych, użytych bibliotek oraz asysty AI na każdym kroku.

Możecie opisać swój proces projektowo-implementacyjny lub sporządzić inne notatki. Może być przydatne przy konsultacji.